

علوم و مهندسی کامپیوتر

نشریه علمی پژوهشی انجمن کامپیوتر ایران

صاحب امتیاز: انجمن کامپیوتر ایران
مدیر مسئول: دکتر جعفر حبیبی
سر دبیر: دکتر حمید سربازی آزاد

شورای علمی

- گ. آقا، استاد دانشگاه ایلینویز، امریکا
ف. ارباب، استاد سی.دبلیو.آی و دانشگاه لایدن، هلند
ن. باقرزاده، استاد دانشگاه کالیفرنیا-ایروین، امریکا
ک. بدیع، دانشیار مرکز تحقیقات مخابرات، ایران
ب. بوزه، استاد دانشگاه ایالتی اورگان، امریکا
ب. پرهامی، استاد دانشگاه کالیفرنیا-سنتا باربارا، امریکا
ف. جهانیان، استاد دانشگاه میشیگان، امریکا
آ. زومایا، استاد دانشگاه سیدنی، استرالیا
ح. سربازی آزاد، استاد دانشگاه صنعتی شریف، ایران
ب. شیرازی، استاد دانشگاه ایالتی واشنگتن، امریکا
ا. کبیر، استاد دانشگاه تربیت مدرس، ایران
ر. صفابخش، استاد دانشگاه صنعتی امیرکبیر، ایران
ح. ر. عربنیا، استاد دانشگاه جورجیا، امریکا
ع. عدالت، استاد کالج سلطنتی لندن، انگلستان
م. فتحی، دانشیار دانشگاه علم و صنعت، ایران
م. ح. قاسمیان، استاد دانشگاه تربیت مدرس، ایران
م. قدسی، استاد دانشگاه صنعتی شریف، ایران
ع. هورسان، استاد دانشگاه ایالتی پنسیلوانیا، امریکا
ف. لو، استاد دانشگاه هنگ کنگ، چین
ع. موقر، استاد دانشگاه صنعتی شریف، ایران
ن. مهدوی امیری، استاد دانشگاه صنعتی شریف، ایران
م. ر. میبیدی، استاد دانشگاه صنعتی امیرکبیر، ایران
ک. ناکانو، استاد دانشگاه هیروشیما، ژاپن
م. ولدخوا، استاد دانشگاه گلاسگو، انگلستان

همکاران دفتر مجله

محمد ارجمند
شهیندخت شهنازی
لیلا نورانی
آرش توکل

نشانی

تهران، خیابان آزادی، ضلع غربی دانشگاه صنعتی شریف، کوچه شهید ولی... صادقی، پلاک ۲۶، طبقه ۴، واحد ۱۶، دفتر انجمن کامپیوتر ایران، نشریه علوم و مهندسی کامپیوتر
تلفن: ۶۶۰۳۲۰۰۰-۶۶۰۸۷۲۲۴
دورنگار: ۶۶۰۲۱۱۴۹
پست الکترونیکی: csi-jcse@ipm.ir
تور جهان گستر: <http://www.jcse.ir>

مقالات درج شده در این نشریه صرفاً بیانگر نظرات مؤلفین آنها است و مسئولیت صحت و سقم داده‌ها و نتایج بر عهده آنها است.

لیتوگرافی، چاپ و صحافی: مرکز نشر دانشگاهی

فهرست مقالات

- گسترش روش حل قيود در تحليل پروتکل‌های امنیتی با قابلیت رمزگذاری جابجاپذیر ۱
سعید جلیلی و سیدمهدی سجادی
- بهینه‌سازی سبد سهام با محدودیت به کمک شبکه عصبی بر روی گرید ۱۱
علی غفاری‌نژاد، محمد کاظم اکبری و فرهنگ نصرت ماکویی
- ارائه مدل جدیدی از الگوریتم فازی حرکت دسته جمعی ذرات با افزایش دقت و صحت همگرایی در بهینه‌سازی توابع (FPSO) ۱۶
محمد فیوضی، جواد حدادنیا، سیدکمال‌الدین موسوی مشهدی و محمد علی‌پور
- موضوعات تحقیقاتی مطرح در مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل ۲۶
احمد عبداله‌زاده بارفروش و شیوا وفادار
- ارائه چارچوب آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و تحلیل نتایج آن با استفاده از معیارهای کمی ۴۰
بهشید بهکمال، محسن کاهانی و سهیلا دهقان‌زاده
- ارائه الگوریتم مناسب برای بخش پردازش سیستم‌های شناسایی رادار ELINT و ESM ۴۶
علی ناصری
- ارزیابی معیارهای کارآیی در مدیریت پردازش شبکه‌ای بوسیله اشیای سیار ۵۹
سیدمحمد رضا میرزابابائی، محمد کاظم اکبری و محمود صادقی

گسترش روش حل قیود در تحلیل پروتکل‌های امنیتی با قابلیت رمزگذاری جابجایی

سیدمهدی سجادی

سعید جلیلی

دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران

چکیده

در این مقاله به منظور ارتقای مدل نفوذی، دو قابلیت جدید به روش حل قیود در توصیف و واری پروتکل‌های امنیتی افزوده شده است. اولین قابلیت، افزودن عناصر رمزنگاری با سطح انتزاع پایین‌تر از رمزکردن با کلید عمومی یا کلید مشترک است که توانایی ما را در توصیف و واری پروتکل‌های امنیتی به طور قابل ملاحظه‌ای افزایش می‌دهد. قابلیت دوم استفاده از متغیر ویژه ای در توصیف ویژگی‌های امنیتی است. صحت و پایان‌پذیری الگوریتم حل قیود بهبود یافته با افزودن این قابلیت‌ها، اثبات شده است. علاوه بر این، روش پیشنهادی، در ابزار تحلیل پروتکل TCO/CS پیاده‌سازی شده، و با استفاده از آن پروتکل انتقال کلید شامیر-ریوست-ادلن واری شده و چهار حمله بر ضد آن شناسایی شده است. بر اساس اطلاعات نویسندگان، دو حمله ناقص ویژگی تصدیق اصالت پیام برای اولین بار گزارش می‌شود.

کلمات کلیدی: حل قیود، عملیات جبری، مدل نفوذی، واری پروتکل‌های امنیتی.

۱- مقدمه

امنیتی استفاده می‌شود. کورین و همکارانش [۳، ۴، ۵] علاوه بر بهبودهایی که بر روش حل قیود ارائه کردند، نوعی منطق زمانی را نیز برای توصیف ویژگی‌های امنیتی پیشنهاد کردند.

در روش‌های معمول تحلیل رسمی پروتکل‌های امنیتی، از جمله روش حل قیود، عناصر پایه رمزنگاری به صورت جعبه سیاه در نظر گرفته می‌شود؛ زیرا فرض می‌شود که مهاجم به ویژگی‌های جبری توابع ریاضی به کار رفته دسترسی ندارد. این فرض حتی برای عملیات رمزنگاری با ویژگی جابجایی و یا شرکت‌پذیری که به طور گسترده در طراحی پروتکل‌های امنیتی استفاده می‌شود، هم صحیح نیست. برای مثال اثبات شده بود پروتکل تصدیق اصالت بازگشتی بول [۶]، در مدلی که XOR را به صورت یک عمل رمزنگاری انتزاعی (به عبارت دیگر جعبه سیاه) در نظر گرفته، صحیح است، اما با در نظر گرفتن ویژگی خودحذفی XOR^۵ آسیب‌پذیری این پروتکل کشف شد [۷].

به لحاظ نظری کارهای بسیاری در زمینه افزودن عناصر رمزنگاری با ویژگی‌های جبری به مدل‌های تحلیل پروتکل‌های امنیتی و ارائه رویه‌های تصمیم‌متناظر انجام شده است. کمون-لاند و شماتیکف [۸] رویه تصمیمی بر اساس تکنیک حل قیود برای مساله امنیتی پروتکل‌های رمزنگاری ارائه کردند که

پروتکل‌های امنیتی مکانیزم اصلی برای تبادل اطلاعات به صورت محرمانه و یا تصدیق اصالت اند. درستی عملکرد این پروتکل‌ها تضمین می‌کند نفوذی نمی‌تواند به اطلاعات محرمانه دست یابد و یا نمی‌تواند به ناحق خود را اصیل جلوه دهد. اهمیت این موضوع ما را به استفاده از روش‌های رسمی در واری پروتکل‌های امنیتی علاقمند می‌کند. روش‌های معمول تحلیل رسمی پروتکل‌های امنیتی بر پایه مدل مهاجم دولو-یانو^۱ بنا شده اند. در این مدل، پروتکل را در قبال مهاجمی تحلیل می‌کنیم که به طور کامل بر شبکه کنترل دارد و می‌تواند پیام‌ها را به دلخواه قطع و یا جعل کند.

در مدل‌سازی رفتار نفوذی یک مساله اساسی چگونگی محدود کردن فضای حالات و به عبارت دیگر، فضای جستجو است. برای غلبه بر این مشکل میلن و شماتیکف [۱] روشی بر اساس مدل فضای رشته‌ای^۲ [۲] ارائه کردند که در آن از رشته‌های پارامتری^۳ برای نقش‌های پروتکل، از عملگر بستر مجموعه ترم‌ها برای نمایش نفوذی و از روش حل قیود^۴ برای واری پروتکل در قبال ویژگی‌های

زمان گذشته، مرتبه اول و نقطه ای است که برای واری ویزگی های وابسته به زمان پروتکل های امنیتی طراحی شده است. این منطق دارای عملگرهای معمول زمانی زمان گذشته، سوره های وجودی و عمومی است که محمول های آن منحصر به محمولات نشانگر رویدادهای وضعیتی و محمول $learn(m)$ است، ویزگی های امنیتی پروتکل توصیف می شود. در نهایت یک جایگذاری^۷ را جستجو می کنیم که با توجه به قوانین دولو-یانو و دانش اولیه نفوذی، آن جایگذاری بتواند مجموعه قیود را حل کند و نقیض فرمول زمانی را نتیجه دهد. در صورت یافتن چنین جایگذاری در واقع موفق به نقض یک ویزگی امنیتی شده ایم و بر اساس آن جایگذاری، حمله ای را یافته ایم. در غیر این صورت با توجه به اثباتی که برای صحیح و کامل بودن این روش ارائه شده است عدم وجود چنین حمله ای نتیجه می شود [۳، ۴، ۵].

۲-۱- مدل سازی پروتکل امنیتی در فضای رشته ای

در رویکرد تحلیل میلن و شماتیکف [۱]، ترمها (اجزای تشکیل دهنده پیام) عبارتند از: مقادیر ثابت: $c \in C$ ، متغیرها: $x \in V$ ، کلید عمومی: $pk(t_1)$ ، زوج سازی: $[t_1, t_2]$ ، درهم سازی: $h(t_1)$ ، رمزگذاری نامتقارن: $[t_1]_{t_2}^{\leftarrow}$ ، رمزگذاری متقارن: $[t_1]_{t_2}^{\rightarrow}$ و امضا: $sig_{t_1}(t_2)$ که t_1 و t_2 دو ترم هستند.

تعریف ۱-۲ (رویداد): رویدادها بر دو نوعند:

رویداد ارتباطی: یا به صورت $\langle a : m \triangleright b \rangle$ است، که خواننده می شود «طرف a پیام m را برای مقصد مورد نظر b ارسال می کند» و یا به صورت $\langle a : m \triangleleft b \rangle$ که خواننده می شود «طرف a پیام m را از طرفی که می پندارد b است، دریافت می کند». در هر دو حالت a را طرف فعال و b را طرف مفعول می نامیم. رویداد وضعیتی: به صورت $p(d_1, \dots, d_n)$ تعریف می شود که $d_i, n \geq 0$ برای $i = 1, 2, \dots, n$ یک ترم، و P یک نماد تابع است. $start(\dots)$ ، $end(\dots)$ و $run(\dots)$ رویدادهای وضعیتی استفاده شده در توصیف یک پروتکل هستند. در مدل فضای رشته ای، نقش^۸ های پروتکل توسط رشته های پارامتری^۱ نمایش داده می شود.

تعریف ۲-۲ (رشته^{۱۱}): عبارت است از دنباله ای متناهی از رویدادها که همه رویدادهای آن دارای عامل فعال یکسان باشند.

تعریف ۲-۳ (شبه کلاف^{۱۲}): به یک مجموعه چندگانه^{۱۳} از رشته ها شبه کلاف اطلاق می شود. در واقع یک شبه کلاف تعیین می کند چه نشستی بین چه نقش هایی برقرار است.

۲-۲- مدل نفوذی

در روش حل قیود برای نمایش توانایی های نفوذی از قوانین دولو-یانو که در شکل ۱ نشان داده شده است، استفاده می شود. هر قانون به شکل $A \rightarrow t$ است، به این معنی که نفوذی با دانستن مجموعه ترم های A ، بر اساس قانون با برچسب^{۱۴} قادر است ترم t را بسازد.

$$\begin{aligned} r_1 : \{[x, y]\} &\rightarrow split \{x, y\} & r_6 : \{x, y\} &\rightarrow senc \{[x]_y^{\leftarrow}\} \\ r_2 : \{[x]_{pk(e)}^{\rightarrow}\} &\rightarrow pdec \{x\} & r_7 : \{x\} &\rightarrow hash \{h(x)\} \\ r_3 : \{[x]_y^{\leftarrow}, y\} &\rightarrow sdec \{x\} & r_8 : \{x\} &\rightarrow sig \{sig_{pk(e)}(x)\} \\ r_4 : \{x, y\} &\rightarrow pair \{[x, y]\} & r_9 : \{[x]_y\} &\rightarrow open \{[x]_y^{\leftarrow}\} \\ r_5 : \{x, y\} &\rightarrow penc \{[x]_y^{\rightarrow}\} & r_{10} : \{[x]_y^{\leftarrow}\} &\rightarrow hide \{[x]_y\} \end{aligned}$$

شکل ۱- مجموعه قوانین نفوذی

XOR را به کار می گیرند. در [۹]، چوالیر و همکارانش، با انتزاعی که برای قوانین نفوذی در نظر گرفتند، رویه تصمیم را بهبود دادند و نشان دادند که مساله امنیتی برای تعداد محدود نشست، برای کلاس بزرگی از پروتکل هایی که از خواص عملگر XOR پشتیبانی می کند، Co-NP است. [۷]، مرجع خوبی برای اطلاع از کارهای تئوری انجام شده برای سایر عملیات جبری می باشد.

تا آنجا که می دانیم تنها سه ابزار موفق به پشتیبانی عملیات جبری در واری پروتکل های امنیتی شده اند. در ابزار Maude-NRL [۱۰] تنها شرکت پذیری محدود شده عملگر پیوند پیام ها (عملگر زوج سازی در این مقاله) مورد توجه قرار گرفته، اما در مورد خواص جبری عمل رمزکردن بحثی به میان نیامده است. با این ابزار برای نمونه یک پروتکل مثال ساده، مورد تحلیل قرار گرفته است. در [۱۱] و [۱۲] به ترتیب مولفان به دو ابزار OFMC و CL-AtSE که از ابزارهای تشکیل دهنده سیستم تحلیل پروتکل AVISPA هستند، عمل رمزکردن XOR را افزوده اند. در این پیاده سازی، یکسان سازی با فرض محدودیت عمق تابع رمزنگاری پیاده سازی شده است. در [۱۱] اگر چه ادعا شده عملگر رمزکردن جابجایی پیاده سازی شده است، اما تنها یک حمله برضد پروتکل شامیر-ریوست-ادلمن^۶ گزارش شده است.

ما روش کورین و همکارانش [۳، ۴، ۵] در تحلیل پروتکل های امنیتی را در دو جهت گسترش داده ایم: (۱) مدل پروتکل و مدل مهاجم (شامل الگوریتم های کاهش و حل قیود) را به گونه ای گسترش داده ایم تا بتوانیم پروتکل هایی که عمل رمزنگاری در آنها قابلیت جابجایی دارد را نیز توصیف کنیم؛ تا بدین صورت قادر باشیم تحلیل پروتکل های امنیتی را در سطحی جزئی تر انجام دهیم (برای مثال برای پشتیبانی عملگر XOR، باید علاوه بر خاصیت جابجایی، خواص شرکت پذیری، عضو خنثی و خود حذفی را نیز اضافه کنیم. در این مقاله تنها روی خاصیت جابجایی متمرکز شده ایم و تلاش کرده ایم این خاصیت را پوشش دهیم). (۲) در تحلیل این پروتکل ها از متغیر ویژه ای در توصیف ویزگی های امنیتی پروتکل استفاده کرده ایم به گونه ای که در صحت، و پایان پذیری الگوریتم حل قیود خدشه ای وارد نشود. علت استفاده از چنین متغیری این است که بتوانیم حمله ای که در آن با جایگذاری مقدار فرستاده شده توسط فرستنده با مقداری به غیر از مقادیر ثابت، صورت می گیرد را شناسایی کنیم.

صحت و پایان پذیری الگوریتم حل قیود بهبود یافته، اثبات شده است؛ به علاوه، سیستم حل قیود پشتیبان عملیات جبری، TOY-CS، در زبان برنامه نویسی تابعی منطقی TOY، پیاده سازی شده است. برای نمونه با این سیستم پروتکل سه گذره شامیر-ریوست-ادلمن، واری شده است و چهار حمله بر ضد آن یافته شده است که تا آنجا که می دانیم دو حمله برای اولین بار شناسایی می شود.

در بخش ۲ مفاهیم پایه آورده شده است. در بخش ۳ چگونگی ارتقای مدل پروتکل و مدل نفوذی شرح داده شده است. بخش ۴ به چگونگی استفاده از متغیر ویژه در توصیف ویزگی های امنیتی می پردازد. بخش ۵ به اثبات صحت و پایان پذیری الگوریتم های کاهش و حل قیود بهبود یافته اختصاص یافته است. بخش ۶ به پیاده سازی اختصاص داده شده است. در بخش ۷ پروتکل شامیر-ریوست-ادلمن را تحلیل نموده ایم و نهایتاً بخش ۸ به نتیجه گیری اختصاص یافته است.

۲- مفاهیم پایه

در روش حل قیود، نقش های پروتکل امنیتی به صورت رشته های پارامتری مدل می شود. به ازای هر درهم سازی رویدادها، مجموعه قیودی را تعریف می کنیم. سپس با استفاده از منطق زمانی PS-LTL [۳، ۴، ۵] (نوعی منطق زمانی خطی،

که به طور متقارن رمزگذاری شده است، با در اختیار گرفتن کلید، رمزگشایی می کند. در این قانون، دلیل استفاده از τ_k پنهان کردن ترم رمزگذاری شده $k: T \cup \{\tau_k\}$ به منظور جلوگیری از کاربرد مجدد این قانون برای قید جدید است.

```

C := initial constraint sequence
σ := {}
repeat
  let c* = m : T be the first constraint in C
  s.t. m is not a variable
  if c* not found then output Solvable!
  apply rule R1 to c* until no longer applicable
  ∀ Ri, i > 1
  if Ri is applicable to C then
    (C'; σ') := Ri(C; σ)
    create node C'; add C → C' edge
    push(C'; σ')
  (C; σ) := pop
until emptystack
    
```

شکل ۲- الگوریتم کاهش P [۱]

پیش از ارائه الگوریتم حل قیود به چند تعریف دیگر می پردازیم:

تعریف ۲-۷ (رد^{۱۶}): به دنباله ای متناهی از رویدادها، رد گفته می شود. الحاق یک رویداد به یک رد را با نماد $\langle tr \text{ ev} \rangle$ نمایش می دهیم.

تعریف ۲-۸: دانش جمع آوری شده توسط نفوذی از یک رد tr را به صورت زیر تعریف می کنیم:

$$K(tr) = \{m \mid last(tr_i) = \langle a : m \triangleright b \rangle, 1 \leq i \leq length(tr)\}$$

که $length(\langle tr \text{ ev} \rangle) = length(\langle tr \rangle) + 1$ ، $length(\langle \rangle) = 0$ ، $last(\langle tr \text{ ev} \rangle) = ev$ شامل tr_i رویداد ابتدای رد tr است.

یک حالت را پنج گانه $\langle S, IK, CS, tr, \varphi \rangle$ در نظر بگیرید که S یک شبه کلاف، IK دانش اولیه نفوذی، CS مجموعه قیود ساده، tr رد و $\varphi = \text{learn}(m)$ نمایانگر ویژگی امنیتی محرمانگی استاندارد در منطق $PS\text{-}LTL$ است.

الگوریتم حل قیود $[\Delta]$ ، که حالت سیستم را از $\langle S, IK, CS, tr, \varphi \rangle$ به $\langle S', IK, CS', tr', \varphi \rangle$ تغییر می دهد در شکل ۴ آورده شده است. یک اجرای الگوریتم برای شبه کلاف اولیه S_0 با دانش اولیه نفوذی IK دنباله ای از قدم های اجرایی است که از حالت $\langle S_0, IK, \{\}, \varphi \rangle$ شروع می شود.

در گام های ۱ و ۲ رشته نمادینی از شبه کلاف S انتخاب می شود. اگر اولین عنصر رشته، رویداد $send$ و یا یک رویداد وضعیتی باشد، تنها این رویداد، به انتهای رد جاری اضافه می شود و تغییری در مجموعه قیود و حل جزئی حاصل نمی شود (لازم است ذکر شود اثر رویداد $send$ در قسمت b گام ۲، جایی که $K(tr)$ را محاسبه می کنیم ظاهر می شود)؛ در حالی که رویداد $receive$ باعث تغییر در هر سه مجموعه قیود، حل جزئی و رد جاری می شود. در گام های ۳ و ۴ فرمول نشان دهنده ویژگی محرمانگی به قید تبدیل شده، بررسی می شود آیا مجموعه قیود ناشی از افزودن این قید به مجموعه قیود جاری، حل پذیر است یا خیر (در گام ۳، عملگر T ، ویژگی محرمانگی را به یک قید نگاشت می کند و رویه تصمیم D با اعمال P ، حل پذیر بودن مجموعه قیود را بررسی می کند) در صورت حل پذیر بودن این مجموعه قیود، الگوریتم با یافتن رد tr' به عنوان سناریو نفوذ خاتمه می یابد؛ در غیر این صورت الگوریتم از گام ۱ ادامه می یابد.

سه قانون اول را قوانین تجزیه، پنج قانون بعد را قوانین ترکیب و دو قانون آخر را قوانین مخفی سازی رمزکردن می گوئیم. این قوانین قابلیت های زیر را برای نفوذی مدل سازی می کنند: زوج سازی و جداسازی ترم ها، درهم سازی، رمزکردن متقارن ترم ها با هر کلید، رمزگشایی متقارن در صورت در اختیار داشتن کلید، در رمزنگاری نامتقارن رمزکردن با هر کلید و رمزگشایی نامتقارن تنها برای ترم هایی که با کلید عمومی نفوذی رمز شده اند، و در نهایت امضا کردن ترم ها با کلید مخفی نفوذی. قوانین مخفی سازی رمزکردن در بخش ۲-۳ توضیح داده شده اند.

عملگر \mathcal{F} بیانگر کلیه ترم هایی است که نفوذی می تواند از یک مجموعه از ترم های بدون متغیر^{۱۴}، استخراج کند و به صورت زیر تعریف می شود:

تعریف ۲-۴ (عملگر بستار): فرض کنیم T مجموعه ای از ترم های بدون متغیر باشد، $\mathcal{F}(T)$ را به صورت $\bigcup_{n \geq 0} \mathcal{F}^n(T)$ تعریف می کنیم که $\mathcal{F}^0(T)$ مجموعه ای با تعریف بازگشتی زیر است:

$$\mathcal{F}^0(T) = T$$

$$\mathcal{F}^n(T) = \mathcal{F}^{n-1}(T) \cup \{t \mid A \rightarrow_{\ell} t \text{ is a Dolev-Yao rule and } A \subseteq \mathcal{F}^{n-1}(T)\}$$

۲-۳- حل قیود

اساسی ترین ایده در روش حل قیود دو مفهوم قید و مجموعه قیود است که به صورت زیر تعریف می شود:

تعریف ۲-۵ (قید): یک قید به صورت زوج $m:K$ تعریف می شود که m یک ترم و K مجموعه ای از ترم ها است. قید ساده قیدی است که m متغیر باشد. یک مجموعه قیود CS مجموعه ای متناهی از قیدهاست.

تعریف ۲-۶ (حل قیود): جایگذاری σ را یک حل برای قید $m:K$ می نامیم اگر $m\sigma \in K\sigma$ بدون متغیر باشند و $m\sigma \in \mathcal{F}(K\sigma)$. γ را یک حل جزئی برای قید $m:K$ می نامیم هرگاه $\gamma m: \gamma K$ قابل حل باشد. حل یک مجموعه قیود جایگذاری است که هر قید را حل پذیر می سازد.

مثال: جایگذاری $\sigma = \{N_A/na, K_{st}/k_{st}\}$ حلی برای مجموعه قیود زیر است: $\{[a, N_A]: \{[a, na]\}, [na, K_{st}]_{k_{st}}^{\rightarrow} : \{[a, na], [N_A, k_{st}]_{k_{st}}^{\rightarrow}\}\}$

الگوریتم کاهش P ^{۱۵} [۱] که در شکل ۲ نشان داده شده است، درخت کاهش با ریشه مجموعه قیود اولیه، می سازد که راس های آن، مجموعه قیود و یال های آن، قوانین شکل ۳ است. هر حالت این الگوریتم با زوج $C; \sigma$ نشان داده می شود که C مجموعه قیود جاری و σ ، جایگذاری جزئی برای متغیرهایی است که در مجموعه قیود اولیه ظاهر می شوند. این الگوریتم، مجموعه قیود اولیه CS را در نهایت به زوج $CS'; \gamma$ نگاشت می کند که CS' مجموعه ای از قیود ساده و γ یک حل جزئی CS است. قیدی که توسط P انتخاب می شود را فعال می گوئیم. در شکل ۳ همه قیود $T_i: v_i$ که قبل از قید فعال $m: T$ ظاهر می شوند را با $C_{<}$ و دنباله قیودی که بعد از $m: T$ ظاهر می شوند را با $C_{>}$ نشان می دهیم.

در شکل ۳، قانون R_1 متغیرهای ظاهر شده در سمت راست قید فعال را حذف می کند. قانون R_2 با یکسان سازی m با ترمی در مجموعه ترم های T ، قید فعال را حذف کرده، عمومی ترین یکسان سازی را به جایگذاری های گذشته اضافه می کند. قوانین R_3 تا R_7 حالتی را مدل می کنند که ترم m را بتوان از مولفه های قابل تجزیه ترم های T استخراج کرد. قوانین R_8 و R_9 بدون جایگذاری متغیرها، ترم های موجود در سمت راست قیود را تا آنجا که ممکن است می شکنند. قانون R_{10} نشان می دهد که نفوذی، ترمی را که با یک کلید عمومی رمز شده است، رمزگشایی می کند. قانون R_{11} نشان می دهد که نفوذی، ترمی را

را به راحتی به روش حل قیود اضافه کنیم.

برای این منظور متغیر ویژه v ، $[x]_{se(y)}$ و $se(y)$ به مجموعه اجزای پیام افزوده می‌شود. متغیر ویژه v در بخش ۴ معرفی می‌گردد. $[x]_{se(y)}$ به معنای عملیات رمز کردن پیام x با کلید مخفی متعلق به طرف v است. فرض می‌شود هر طرف تنها قادر است با کلید مخفی خود رمزگذاری یا رمزگشایی کند و هیچ طرفی به کلید مخفی طرف مقابل دسترسی ندارد. عملیات رمز کردن با کلید مخفی دارای خاصیت جابجایی است. به این معنی که اگر $se(a)$ و $se(b)$ به ترتیب کلیدهای مخفی طرف‌های a و b باشد، داریم: $[[x]_{se(a)}]_{se(b)} = [[x]_{se(b)}]_{se(a)}$. اصلاح مجموعه قوانین کاهش، دومین تغییر، در روش حل قیود است. ابتدا قانون R_1 به صورت زیر اصلاح می‌شود:

$$R'_1: \frac{C_{<}, m: x \cup T, C_{>}; \sigma}{C_{<}, m: T, C_{>}; \sigma} \quad x \text{ is a variable and } x \neq v$$

این قانون، متغیر ویژه v که در سمت راست یک قید ظاهر شده است را حذف نمی‌کند. قانون R_2 نیز به صورت زیر اصلاح می‌شود:

$$R'_2: \frac{C_{<}, m: T, C_{>}; \sigma}{\tau C_{<}, \tau C_{>}; \tau \cup \sigma} \quad \text{where } \tau \in mcsu(m, t), t \in T$$

که در آن $mcsu(m, t)$ مجموعه کامل و مینیمال از یکسان سازها به پیمانده جابجایی، حاصل از الگوریتم یکسان سازی جابجایی^{۱۷} [۱۳، ۱۴] میان دو ترم m و t است. (ثابت شده است که برای هر دو ترم s و t ، $mcsu(s, t)$ همواره وجود دارد و مجموعه‌ای متناهی است [۱۳، ۱۴]). سایر قوانین بدون تغییر باقی می‌ماند:

$$R'_i = R_i, \quad i = 3, \dots, 11$$

قوانین زیر را نیز به مجموعه قوانین کاهش اضافه می‌کنیم:

$$R'_{12}: \frac{C_{<}, [m]_k: T, C_{>}; \sigma}{C_{<}, k: T, m: T, C_{>}; \sigma}$$

$$R'_{13}: \frac{C_{<}, [m]_{k_1}]_{k_2}: T, C_{>}; \sigma}{C_{<}, k_1: T, [m]_{k_2}: T, C_{>}; \sigma}$$

$$R'_{14}: \frac{C_{<}, m: T \cup [t]_k, C_{>}; \sigma}{C_{<}, k: T \cup [t]_k, m: T \cup t \cup k, C_{>}; \sigma}$$

$$R'_{15}: \frac{C_{<}, m: T \cup [t]_{k_1}]_{k_2}, C_{>}; \sigma}{C_{<}, k_1: T \cup [t]_{k_2}]_{k_1}, m: T \cup [t]_{k_2} \cup k_1, C_{>}; \sigma}$$

قوانین R'_{12} و R'_{14} و عملیات رمز کردن و R'_{15} رمزگشایی را با استفاده از کلید مخفی با خاصیت جابجایی مدل می‌کنند. در واقع قوانین R'_{13} و R'_{15} خاصیت جابجایی رمز کردن را مدل می‌کنند. به علاوه دلیل استفاده از ترم $[t]_k$ (مشابه آنچه در مورد ترم $[t]_k$ گفته شد)، پنهان کردن ترم رمزگذاری شده $[t]_k$ به منظور جلوگیری از کاربرد مجدد این قانون برای قید جدید $k: T \cup [t]_k$ است. آخرین تغییر، بر الگوریتم کاهش \mathbf{P} اعمال می‌گردد. بدین صورت که خط پنجم الگوریتم را به صورت زیر اصلاح می‌کنیم:

s.t. m is a special variable v else is not a variable

در الگوریتم جدید که آن را \mathbf{P}_{Rev} می‌نامیم اجازه می‌دهیم قیدهایی به شکل $v: T$ به عنوان ورودی ظاهر شوند. به علاوه در الگوریتم \mathbf{P}_{Rev} از قوانین R'_i به جای قوانین R_i استفاده می‌کنیم.

۴- استفاده از متغیر ویژه در توصیف ویژگی‌ها

در روش حل قیود برای واریسی ویژگی محرمانگی استاندارد (که به صورت $\mathbf{learn}(m)$ بیان می‌شود) در پروتکلی که در آن مقدار محرمانه ثابت m توسط

$$R_1: \frac{C_{<}, m: x \cup T, C_{>}; \sigma}{C_{<}, m: T, C_{>}; \sigma} \quad x \text{ is a variable}$$

$$R_2: \frac{C_{<}, m: T, C_{>}; \sigma}{\tau C_{<}, \tau C_{>}; \tau \cup \sigma} \quad \text{where } \tau \in mgu(m, t), t \in T$$

$$R_3: \frac{C_{<}, [m_1, m_2]: T, C_{>}; \sigma}{C_{<}, m_1: T, m_2: T, C_{>}; \sigma}$$

$$R_4: \frac{C_{<}, h(m): T, C_{>}; \sigma}{C_{<}, m: T, C_{>}; \sigma}$$

$$R_5: \frac{C_{<}, [m]_k: T, C_{>}; \sigma}{C_{<}, k: T, m: T, C_{>}; \sigma}$$

$$R_6: \frac{C_{<}, [m]_k^{\leftrightarrow}: T, C_{>}; \sigma}{C_{<}, k: T, m: T, C_{>}; \sigma}$$

$$R_7: \frac{C_{<}, \text{sig}_{pk(e)}(m): T, C_{>}; \sigma}{C_{<}, m: T, C_{>}; \sigma}$$

$$R_8: \frac{C_{<}, m: [t_1, t_2] \cup T, C_{>}; \sigma}{C_{<}, m: t_1 \cup t_2 \cup T, C_{>}; \sigma}$$

$$R_9: \frac{C_{<}, m: [t]_{pk(e)}^{\rightarrow} \cup T, C_{>}; \sigma}{C_{<}, m: t \cup T, C_{>}; \sigma}$$

$$R_{10}: \frac{C_{<}, m: [t]_k^{\rightarrow} \cup T, C_{>}; \sigma}{\tau C_{<}, \tau m: \tau [t]_k^{\rightarrow} \cup \tau T, \tau C_{>}; \tau \cup \sigma} \quad \text{where } \tau \in mgu(k, pk(e)), k \neq pk(e)$$

$$R_{11}: \frac{C_{<}, m: T \cup [t]_k^{\leftrightarrow}, C_{>}; \sigma}{C_{<}, k: T \cup [t]_k, m: T \cup t \cup k, C_{>}; \sigma}$$

شکل ۳- مجموعه قوانین کاهش [۱]

1. Choose non-deterministically a non-empty strand $r \in S$. Let $r = \langle ev \ r' \rangle$. Consider the following cases for ev :
 2. If ev is a send communication event or a status event, let γ be the empty substitution and CS'' be CS .
 3. If ev is a receive communication event, i.e. $ev = \langle a: m \triangleleft b \rangle$, check that the intruder can generate m using the knowledge $K(tr) \cup IK$, by applying procedure \mathbf{P} to $CS \cup \{m: (K(tr) \cup IK)\}$ obtaining a new simple constraint set CS'' and a partial solution γ (Note that there may be many possible CS'' and γ).
 4. Let $S' := (S \setminus \{r\} \cup \{r'\})\gamma$, $CS' := CS''$ and $tr' := \langle tr\gamma \ ev\gamma \rangle$.
 5. Compute $\pi = T(\mathbf{learn}(m), tr', IK) = m: (K(tr') \cup IK)$.
- Evaluate $D(\pi, CS')$:
- Apply \mathbf{P} to $CS' \cup \{m: (K(tr') \cup IK)\}$.
- If $D(\pi, CS') = \text{true}$, then stop and output the current (offending) trace tr' . Otherwise, continue the run from step 1

شکل ۴- الگوریتم حل قیود [۵]

۳- ارتقای مدل پروتکل و مدل نفوذی

ما در این مقاله با توسعه مدل پروتکل، عملگر جدید رمز با کلید مخفی که دارای خاصیت جابجایی است را به مدل افزوده ایم، تا بدین صورت زیر بنایی فراهم کنیم که بتوانیم سایر خواص جبری که در توصیف پروتکل‌های امنیتی استفاده می‌شود

می نامیم اگر جایگذاری مانند σ وجود داشته باشد که $(tr' = tr\sigma) \ t' = t\sigma$.
در اینجا مفهوم اعتبار را برای رد tr تعریف می کنیم.

تعریف ۲-۵: رد tr را با در نظر داشتن دانش اولیه نفوذی IK معتبر می نامیم اگر یا تهی باشد و یا برای هر $1 \leq i \leq \text{length}(tr) - 1$ داشته باشیم:

$\text{last}(tr_{i+1}) = \langle a : m \triangleleft b \rangle$ implies that

$$\exists n \in \mathcal{F}(K(tr_i) \cup IK), \text{mcsu}(m, n) \neq \{\}$$

اعتبار عینی برای فرمول به شکل $\text{learn}(m)$ به صورت زیر تعریف می شود:

تعریف ۳-۵: فرمول $\text{learn}(m)$ را در نظر بگیرید. برای رد بدون متغیر tr و دانش اولیه نفوذی IK ، نماد $\langle tr, IK \rangle \models \text{learn}(m)$ را به صورت زیر تعریف می کنیم:

$\langle tr, IK \rangle \models \text{learn}(m)$ iff

$$\exists n \in \mathcal{F}(K(tr) \cup IK), \text{mcsu}(m, n) \neq \{\}$$

اعتبار نمادین برای فرمول $\text{learn}(m)$ به صورت زیر تعریف می شود:

تعریف ۴-۵: فرمول $\text{learn}(m)$ که m یک ترم است را در نظر بگیرید. برای رد tr و دانش اولیه نفوذی IK ، $\langle tr, IK \rangle \models \text{learn}(m)$ را به صورت زیر تعریف می کنیم:

$\langle tr, IK \rangle \models \text{learn}(m)$ iff

$$\exists n \in \mathcal{F}(K(tr) \cup IK), \sigma = \text{mcsu}(m, n),$$

$$\forall tr' = vi(tr\sigma, IK) = tr\sigma\gamma,$$

$$\langle tr', IK \rangle \models \text{learn}(m\sigma\gamma)$$

که $tr' = vi(tr\sigma, IK)$ نمونه معتبر رد $tr\sigma$ به ازای IK است.

تعریف ۵-۵: فرمول $m : K$ را در نظر بگیرید که m یک ترم و K مجموعه ای از ترم ها است. σ را یک جایگذاری در نظر بگیرید که تمام ترم های K را بدون متغیر می کند (به استثنای متغیر ویژه v). نماد $\sigma \models m : K$ را به صورت زیر تعریف می کنیم:

$$\sigma \models m : K \text{ iff } \exists n \in \mathcal{F}(K\sigma), \text{mcsu}(n, m\sigma) \neq \{\}$$

ابتدا نشان می دهیم الگوریتم \mathbf{P}_{Rev} پایان پذیر است.

قضیه ۵-۱: الگوریتم کاهش \mathbf{P}_{Rev} پایان پذیر است.

اثبات: رویکرد اثبات همان رویکرد مرجع [۱] است. ما اثبات را تنها برای قوانین اصلاح یافته انجام می دهیم. برای سایر قوانین به [۱] مراجعه کنید.

تابع مقیاس پایان پذیری^{۱۸} یک مجموعه قیود، را به صورت (N_v, N_s) تعریف می کنیم که N_v تعداد متغیرهای مجزای آن و N_s را مقیاس توسعه^{۱۹} می نامیم. ترتیب را ترتیب لغت نامه ای در نظر می گیریم.

برای تعریف مقیاس توسعه، اندازه ساختار ترم m ، $|m|$ ، را مجموع تعداد عملگرها، ثابتها و متغیرهای آن تعریف می کنیم. مقیاس توسعه یک مجموعه قیود را مجموع مقیاس توسعه های کلیه قیده های آن تعریف می کنیم. مقیاس توسعه یک قید به شکل $m : T$ به صورت $|m| \cdot \chi(T)$ است که χ به صورت شکل ۵ تعریف می شود (در این تعریف $+$ و بدون علامت، به ترتیب جمع و ضرب طبیعی را نشان می دهند):

$$\chi(t) = 2 \text{ if } t \text{ is a var or constant} \quad \chi([t]_k^*) = \chi(t)\chi(k) + 1$$

$$\chi(\{t_1, \dots, t_n\}) = \chi(t_1) \cdots \chi(t_n) \quad \chi([t]_k) = 1$$

$$\chi([t_1, t_2]) = \chi(t_1)\chi(t_2) + 1 \quad \chi([t]_k) = 1$$

$$\chi([t]_k) = \chi(t)\chi(k)(|k| + 1) \quad \chi(\text{sig}_k(t)) = \chi(t) + 1$$

$$\chi([t]_k^*) = \chi(t)\chi(k) + |k| + 1 \quad \chi(h(t)) = \chi(t) + 1$$

شکل ۵- تابع χ

نشان می دهیم هر قانون کاهش، مقدار تابع مقیاس پایان پذیری را کاهش

a به b فرستاده می شود، الگوریتم \mathbf{P} را در یک روند تکرار بر مجموعه قیودی به شکل $CS' \cup \{m : (K(tr') \cup IK)\}$ اثر می دهیم. این عمل به معنای تلاش برای یافتن رد مناسب tr' است که نفوذی با پیمودن آن قادر به کشف m خواهد شد؛ بدیهی است با ثابت در نظر گرفتن m ، حالتی را که در آن فرستنده پیام، مقداری به غیر از مقادیر ثابت را به عنوان مقدار محرمانه می فرستد را در نظر نگرفته ایم. از سوی دیگر در روش حل قیود در استفاده از متغیر با مشکلات زیر مواجه هستیم:

هر متغیر باید در هر رشته برای اولین بار، در رویداد $\langle a : m \triangleleft b \rangle$ ظاهر شود. دلیل وضع این شرط تامین کامل بودن الگوریتم کاهش \mathbf{P} است.

فرمول های منطق زمانی $PS\text{-LTL}$ که برای توصیف ویژگی های زمانی پروتکل های امنیتی به کار می رود متغیر آزاد ندارند. این شرط به این علت وضع شده است که پس از تبدیل فرمول ها به شکل نرمال، به شکل قیود منفی ای که متغیر در سمت چپ آن ظاهر شده باشد (قیده های به شکل $\neg(x:T), x \in \mathcal{V}$ ، درنیاید (با توجه به تمرکز ما در این مقاله بر محرمانگی، این مشکل خود بخود منتفی است زیرا شکل نرمال فرمول بیانگر محرمانگی تنها به صورت قید مثبت $x:T$ است).

متغیر مشترک در فرمول منطق زمانی و رد اجرایی جاری وجود نداشته باشد. این شرط به دلیل به وجود نیامدن ابهام وضع شده است.

نحوه استفاده ما از متغیر به این صورت است که از متغیر ویژه ای به جای اطلاع محرمانه m تنها در $\neg \text{learn}(m)$ و در رویداد $\langle a : f(m) \triangleright b \rangle$ که متعلق به رشته نمایانگر نقش a است استفاده می کنیم. دلیل استفاده از متغیر این است که می خواهیم بدانیم آیا مقداری (به غیر از مقادیر ثابت) وجود دارد که با فرستادن آن به عنوان اطلاع محرمانه، نفوذی با استفاده از دانش اولیه و توانایی استخراج دانش جدید، بتواند آن را کشف کند؟

برای غلبه بر مشکلات ناشی از افزودن متغیر، متغیر ویژه را به صورت متغیر سراسری در نظر می گیریم تا از ابهام پیشگیری شود (در زبان $TO\mathcal{V}$ [۱۵]، ما این متغیر را به صورت مقدار ثابتی از «نوع داده» variable پیاده سازی کرده ایم).

با این مقدمات لازم است دو تغییر بر الگوریتم حل قیود (شکل ۴) اعمال کنیم. اولین تغییر، به کارگیری الگوریتم کاهش \mathbf{P}_{Rev} به جای \mathbf{P} در گام های اول و سوم الگوریتم حل قیود است. دومین تغییر ناشی از استفاده از متغیر ویژه v است که باعث می شود گام سوم الگوریتم به صورت زیر تغییر کند:

6. Compute

$$\pi = T(\text{learn}(v\gamma), tr', IK) = v\gamma : (K(tr') \cup IK),$$

If v is not bounded by any value then

Continue the run from step 1

Otherwise

Evaluate $D(\pi, CS')$:

Apply \mathbf{P}_{Rev} to $CS' \cup \{v\gamma : (K(tr') \cup IK)\}$.

که v متغیر ویژه و γ جایگذاری است که در گام دوم به دست آمده است. در این گام بررسی می شود، آیا مقداری که در گام اول به متغیر v مقید شد (متغیر ویژه v ممکن است پس از محاسبه $K(tr')$ در قسمت b گام ۱ در سمت راست قید فعال ظاهر شود و با اعمال \mathbf{P}_{Rev} در گام ۱، مقید شود)، از دانش جدید نفوذی قابل دسترسی است. در صورت مقید نشدن متغیر ویژه v ، اجرا از گام ۱ ادامه می یابد.

۵- اثبات صحت و پایان پذیری

در این بخش صحت و پایان پذیری الگوریتم کاهش \mathbf{P}_{Rev} و الگوریتم حل قیود بهبود یافته را اثبات می کنیم. قبل از آن به چند تعریف می پردازیم.

تعریف ۵-۱: ترم بدون متغیر t' (رد بدون متغیر tr') را نمونه ترم t (رد tr)

لم ۵-۲: فرمول $\text{learn}(m)$ ، رد tr و دانش اولیه IK را در نظر بگیرید. فرض کنید σ جایگذاری بدون متغیر کننده‌ای باشد که $\text{var}(tr) \subseteq \text{dom}(\sigma)$ و $tr\sigma$ نمونه معتبر رد tr باشد. نشان دهید:

$$\langle tr\sigma, IK \rangle \models \text{learn}(m\sigma) \text{ iff } \sigma \models m : (K(tr) \cup IK)$$

اثبات:

$$\begin{aligned} &\langle tr\sigma, IK \rangle \models \text{learn}(m\sigma) \\ &\text{iff (def. of } \models) \\ &\exists n \in \mathcal{F}(K(tr\sigma) \cup IK), \\ &\quad mcsu(m\sigma, n) \neq \emptyset, tr\sigma \text{ is valid} \\ &\text{iff (IK is ground)} \\ &\exists n \in \mathcal{F}((K(tr) \cup IK)\sigma), \\ &\quad mcsu(m\sigma, n) \neq \emptyset, tr\sigma \text{ is valid} \\ &\text{iff (def. of } \models' \text{ and } tr\sigma \text{ is valid by hypothesis)} \\ &\sigma \models' m : (K(tr) \cup IK) \end{aligned}$$

لم ۵-۳: فرض کنید

$$\pi = T(\text{learn}(v\gamma), tr', IK) = v\gamma : (K(tr') \cup IK)$$

ثابت کنید اگر با اعمال \mathbf{P}_{Rev} بر $CS' \cup \{v\gamma : (K(tr') \cup IK)\}$ به جواب برسیم آنگاه $\sigma \models' \pi$.

اثبات: فرض کنید ρ_k جایگذاری باشد که با اعمال \mathbf{P}_{Rev} بر $CS' \cup \{v\gamma : (K(tr') \cup IK)\}$ حاصل شده است. نشان می‌دهیم $\rho_k \models' v\gamma : (K(tr') \cup IK)$.

چون $CS' \cup \{v\gamma : (K(tr') \cup IK)\}$ با جواب ρ_k قابل حل است. به دلیل صحت الگوریتم \mathbf{P}_{Rev} داریم:

$$\exists n \in \mathcal{F}((K(tr') \cup IK)\rho_k), mcsu(v\gamma\rho_k, n) \neq \{\}$$

که این دقیقاً تعریف $\rho_k \models' v\gamma : (K(tr') \cup IK)$ است.

قضیه ۵-۳: فرض کنید S_0 یک شبه کلاف، IK دانش اولیه نفوذی، $\varphi = \neg \text{learn}(v\gamma)$ ویژگی امنیتی و $\neg\varphi = \text{learn}(v\gamma)$ فرمول نشان دهنده حمله باشد. فرض کنید $\langle S', IK, CS', tr', \varphi \rangle$ حالتی از الگوریتم حل قیود بهبود یافته باشد. با فرض:

$$\pi = T(\text{learn}(v\gamma), tr', IK) = v\gamma : (K(tr') \cup IK)$$

اگر \mathbf{P}_{Rev} را حل کند، آنگاه $\langle tr', IK \rangle \models \text{learn}(v\gamma)$.

اثبات: چون \mathbf{P}_{Rev} بر $CS' \cup \{v\gamma : (K(tr') \cup IK)\}$ را حل می‌کند از لم ۵-۳ نتیجه می‌شود وجود دارد ρ_k که $\rho_k \models' v\gamma : (K(tr') \cup IK)$. با توجه به لم ۵-۱، $tr'\rho_k$ معتبر است. در نتیجه بنابر لم ۵-۲ $\langle tr', IK \rangle \models \text{learn}(v\gamma)$.

۶- پیاده‌سازی

در سیستم CS-TOL الگوریتم \mathbf{P}_{Rev} و الگوریتم حل قیود بهبود یافته به طور کامل در زبان برنامه نویسی تابعی منطقی TOL [۱۵] پیاده سازی شده است. ما در کل پیاده‌سازی تنها سه «نوع داده» را اعلان نموده‌ایم. نوع داده atom ، نوع داده variable و نوع داده term .

در شکل ۶ چگونگی اعلان سه «نوع داده» نشان داده شده است. از نوع داده atom برای نمایش مقادیر ثابتی چون نانس، کلیدهای مشترک و شناسه‌ها استفاده می‌شود. نوع داده variable برای نمایش متغیر ویژه v استفاده می‌شود. نوع داده term به منظور نشان دادن اجزای پیام به کار می‌رود. به منظور برقراری رابطه زیر نوعی بین نوع داده‌های atom و variable با نوع داده term به ترتیب ترم‌های atom و var variable را به عنوان مقادیر ثابت و متغیر در نظر می‌گیریم. به عنوان مثال کلید مشترک atm kab با atm نشان می‌دهیم. از lis [term] به عنوان لیستی از ترم‌ها استفاده می‌کنیم. msg atom [term]

می‌دهد. ما این مطلب را تنها برای قوانین کاهش اصلاح یافته نشان می‌دهیم. (برای سایر قوانین کاهش به [۱] مراجعه شود).

قانون R'_1 با وجودی که متغیر v را مستثنی کرده‌ایم هر جا اعمال شود، از N_s به مقدار $\chi(x) = 2$ می‌کاهد؛ R'_2 با وجود تبدیل mgv به $mcsu$ ، یا یک متغیر را با جایگذاری حذف می‌کند و بنابراین از مقدار N_v می‌کاهد، و یا در غیر این صورت مقدار ثابت در سمت چپ خود را بدون جایگذاری تطبیق می‌دهد و با اینکه ترم شامل متغیر سمت چپ خود را با ترمی شامل تنها متغیر موجود در سمت راست، v ، یکسان‌سازی جایجایی می‌کند که در هر دو حالت اخیر، N_v را تغییر نمی‌دهد اما N_s را با حذف قید، کاهش می‌دهد. R'_2 و R'_3 ، N_v را کاهش می‌دهند به این ترتیب که تغییری در T نمی‌دهند اما سمت چپ را به مولفه‌هایی با مجموع اندازه ساختار کمتر تقسیم می‌کنند. جالب‌ترین مورد R'_4 و R'_{15} است؛ برای مثال R'_{15} ، $m : T \cup [t]_{k_1}^{k_2}$ با مقیاس توسعه

$$|m| \chi(T) (\chi(t) \chi(k_1) (|k_1| + 1)) \chi(k_2) (|k_2| + 1)$$

را به $m : T \cup [t]_{k_2}^{k_1} \cup k_1$ و $k_1 : T \cup [t]_{k_2}^{k_1}$ با مجموع مقیاس توسعه

$$|k_1| \chi(T) + |m| \chi(T) \chi(t) \chi(k_2) (|k_2| + 1) \chi(k_1)$$

تبدیل می‌کند که مقداری کوچکتر است. R'_{14} نیز به همین صورت است. بنابراین \mathbf{P}_{Rev} پایان‌پذیر است.

قضیه ۵-۲: الگوریتم کاهش \mathbf{P}_{Rev} صحیح است.

اثبات: برای اثبات صحت کافی است نشان دهیم هر قانون کاهش صحیح است به عبارت دیگر اگر با اعمال قانون کاهش R'_i بر مجموعه قیود C جواب σ به دست آید: $\sigma \vdash C$ ، آنگاه $\sigma \vdash C$ ، ما این مطلب را تنها برای قوانین اصلاح یافته نشان می‌دهیم. برای سایر قوانین به [۱] مراجعه شود.

قانون R'_1 در حالتی که متغیر ویژه v در سمت راست ظاهر شود قابل اعمال نیست و در سایر حالات مشابه قانون R_1 است.

برای قانون R'_2 توجه کنید که اگر $\sigma \vdash TC_{<}, C_{>}$ آنگاه $\sigma \cup T \vdash C_{<}, C_{>}$. همچنین $\sigma \cup T \vdash m : t \cup T'$ (توجه کنید که $T = t \cup T'$)، زیرا $\sigma \cup T \vdash C$ و بنا بر تعریف $mcsu(m, t)$.

برای قوانین R'_{12} و R'_{13} صحت از آنجا ناشی می‌شود که \mathcal{F} تحت عملگر \rightarrow^l متناظر، بسته τ است. برای مثال در مورد R'_{13} اگر $\sigma \vdash C_{<}, C_{>}$ قابل حل باشد، آنگاه $\exists \sigma$ به گونه‌ای که $k_1 \sigma \in \mathcal{F}(T\sigma)$ و $[m\sigma]_{k_2\sigma} \in \mathcal{F}(T\sigma)$ ، که چون \mathcal{F} تحت

$$\rightarrow^{sec2} \text{ بسته است بنابراین } [m\sigma]_{k_1\sigma}^{k_2\sigma} \in \mathcal{F}(T\sigma)$$

نهایتاً نحوه عمل در مورد دو قانون R'_{14} و R'_{15} مشابه قانون R_{11} است که به [۱] رجوع شود.

پایان‌پذیری الگوریتم حل قیود بهبود یافته از متناهی بودن مجموعه قیود اولیه و پایان‌پذیری الگوریتم \mathbf{P}_{Rev} نتیجه می‌شود. اکنون به صحت الگوریتم حل قیود بهبود یافته می‌پردازیم.

لم ۵-۱: فرض کنید S_0 یک شبه کلاف، و IK دانش اولیه نفوذی باشد. در الگوریتم حل قیود بهبود یافته، فرض کنید $\langle S', IK, CS', tr', \varphi \rangle$ حالتی باشد که با شروع از حالت $\langle S_0, IK, \emptyset, \langle \cdot \rangle, \varphi \rangle$ به آن رسیده‌ایم. آنگاه برای هر حل σ از CS' ، $tr'\sigma$ به ازای IK معتبر است.

اثبات: اثبات دقیقاً مشابه اثبات در الگوریتم حل قیود [۴] است. با این تفاوت که از تعریف‌های متناظری که در اینجا ارائه شده است استفاده می‌کنیم.


```
(msg rcv [(M : ^^ (se A))]),
(msg send [(M : ^^ (se A)) : ^^ (se B)]),
(msg rcv [(M : ^^ (se B))]),
(msg end [B, A, (atm resp), M])
]
```

برای واری پروتکل در قبال محرمانگی، به صورت زیر عمل می کنیم (برای اختصار از نمادهای a, b و m به ترتیب به جای atm a, atm b و atm m و v, A, B, M، به ترتیب به جای v, var A, var B و var M، استفاده شده است):

```
semibundle1 =
(((a, Init1), [b, Resp1]), (neg (learn v))) <==
Init1 == initiator a b v m,
Resp1 == responder A B M
```

در اینجا v متغیر ویژه و a, b و m مقادیر ثابتند که به ترتیب بیانگر آغازگر، مخاطب و مقدار محرمانه دریافت شده توسط آغازگر است. این شبه کلاف شامل دو رشته آغازگر و مخاطب است. دانش اولیه نفوذی را (atm a), (atm b), (atm e) در نظر می گیریم.

با اجرای برنامه رد اجرایی به صورت زیر تولید می شود (برای اختصار از نمادهای a, b و m به ترتیب به جای atm a, atm b و atm m، و از A و B به ترتیب به جای var A و var B، استفاده شده است):

```
7. Output ->
8. [
9. [a, (msg start [a, b, init])],
10.[b, (msg start [_A, _B, resp])],
11.[a, (msg send [m : ^^ (se b) : ^^ (se a)])],
12.[a, (msg rcv [m : ^^ (se a) : ^^ (se b)])],
13.[a, (msg send [m : ^^ (se b)])],
]
```

که بیانگر حمله زیر است:

```
1. A -> I(B) : {M}Ka
2. I(B) -> A : {M}Ka
3. A -> I(B) : M
```

و $M = \{m\}Kb$ و $I(B)$ نفوذی است که خود را B جا زده است.

در این حمله که در یک نشست انجام می شود، نفوذی، در گذر دوم همان پیام فرستاده شده در گذر اول را به طرف B بر می گرداند. در گذر سوم پس از رمزگشایی از پیام گذر دوم، مقدار محرمانه توسط طرف A به نفوذی فرستاده می شود.

برای یافتن حمله دیگر ناقص ویژگی محرمانگی شبه کلاف را به صورت زیر در نظر می گیریم (برای اختصار از نمادهای a, b و m، به ترتیب به جای atm a, atm b, atm m، A, I, M، به ترتیب به جای var A, var I, var M، استفاده شده است):

```
semibundle2 =
(((a, Init1), [b, Resp1]), (neg (learn m))) <==
Init1 == initiator a I m m,
Resp1 == responder A I M
```

در شبه کلاف semibundle2 نیز یک رشته آغازگر و یک رشته مخاطب در نظر گرفته شده است. در اینجا دانش اولیه نفوذی به صورت (atm a), (atm b), (atm e), (se (atm e)) در نظر گرفته می شود؛ که atm a, atm b و atm e به ترتیب شناسه های طرف a، طرف b و نفوذی هستند و se (atm a) کلید محرمانه نفوذی است.

پس از اجرا، رد اجرایی به صورت زیر تولید می شود (برای اختصار از نمادهای a, e و m، به ترتیب به جای atm a, atm e و atm m، و از B به جای var B، استفاده شده است):

```
14. Output ->
15. [
16.[a, (msg start [a, e, (atm init)])],
17.[b, (msg start [e, _B, (atm resp)])],
```

نشانی دهنده یک رویداد است. برای مثال atm na) : ++ (atm kab) بیانگر یک رویداد ارتباطی receive است که نشان می دهد مقدار نانس atm na که با کلید مشترک atm kab رمز شده است، دریافت می شود (عملگر ++: که رمز کردن با کلید متقارن را نشان می دهد در ابتدا به صورت یک عملگر دودویی با اولویت ۴۰ تعریف شده است). در پیاده سازی یکسان سازی و یکسان سازی به پیمان جابجایی از [۱۳] و [۱۴] استفاده شده است. به علاوه، از پیاده سازی یکسان سازی، در زبان منطقی ALE [۱۶] نیز بهره گرفته شده است.

```
infix 40 :++
...
data atom = a | b | s | e | na |
kab | send | ...
data variable = v
data term =
atm atom | var variable |
lis [term] | msg atom [term] |
pk term | ... | term :++ term | ...
```

شکل ۶- نوع های داده

۷- واری پروتکل شامیر - ریوست - ادلمن

در این بخش پروتکل شامیر- ریوست- ادلمن [۷] را با استفاده از سیستم CS-TOT مورد واری قرار می دهیم. هدف پروتکل سه مرحله ای شامیر- ریوست- ادلمن برقراری ارتباط امن میان دو طرفی است که نه کلید متقارن مشترکی دارند و نه یک طرف کلید عمومی طرف مقابل را می داند. پروتکل در سه مرحله انجام می شود:

A, B: طرف های پروتکل، M: داده محرمانه، Ka, Kb: به ترتیب کلیدهای مخفی (کلیدی که تنها دارنده آن قادر به رمزگذاری و رمزگشایی پیام توسط آن است) به ترتیب متعلق به طرف های A و B، و عمل رمزکردن دارای خاصیت جابجایی است: $\{ \{M\}Ka \}Kb = \{ \{M\}Kb \}Ka$

```
1. A -> B : {M}Ka
2. B -> A : { {M}Ka }Kb
3. A -> B : {M}Kb
```

شرح پیام های پروتکل

در مرحله اول طرف A داده محرمانه M را با کلید مخفی خود Ka رمز می نماید و به طرف B می فرستد. در مرحله دوم طرف B پیام دریافت شده را با کلید مخفی خود Kb رمز نموده، به A می فرستد. در مرحله سوم طرف A پیام دریافت شده را با کلید خود رمزگشایی می کند و مقدار رمزگشایی شده را به B می فرستد. اکنون با توجه به اینکه B دارای کلید محرمانه Kb است می تواند پیام دریافت کرده را رمزگشایی کند و به M دست یابد.

محرمانگی استاندارد

رشته های پارامتری که نشان دهنده نقش های پروتکل (آغازگر و مخاطب) در روش حل قیوداند به صورت شکل ۷ نمایش داده می شوند.

```
initiator A B M1 M2 =
(lis [
(msg start [A, B, (atm init)]),
(msg send [(M1 : ^^ (se A))]),
(msg rcv [(M2 : ^^ (se A)) : ^^ (se B)]),
(msg send [(M2 : ^^ (se B))]),
(msg end [A, B, (atm init), M1, M2])
])
```

```
responder A B M =
(lis [
(msg start [B, A, (atm resp)]),
```

سوم نشست سوم، پیام $\{M\}Kb\}Kb$ که در گذر سوم نشست دوم به دست آمد را به طرف B می‌فرستد. نهایتاً نفوذی که خود را A جا زده است، نشست اول را با ارسال پیام $\{M\}Kb\}Kb$ که در گذر سوم نشست دوم به دست آمد، به طرف B، کامل می‌کند. در این حمله پس از انجام نشست اول، B گمان می‌کند که پیام محرمانه، $\{M\}Kb$ است در حالی که پیام محرمانه، M است. بنابراین این حمله نشان می‌دهد پروتکل، ناقض ویژگی تصدیق اصالت پیام است.

حمله ناقض تصدیق اصالت پیام دیگری را نیز می‌توان پیدا کرد. این بار شبه کلاف را به صورت زیر در نظر می‌گیریم (برای اختصار از نمادهای $a1, b1, a2, b2, m$ ، به ترتیب به جای $atm\ a, atm\ b, atm\ a2, atm\ b1, atm\ a1, atm\ b2$ و $atm\ m$ ، v و M ، به ترتیب به جای $var\ v$ و $var\ M$ ، استفاده شده است):

```
Semibundle3 =
([[a1,Init1], [b1, Resp1], [b2,Resp2]],
(neg (learn v))) <==
Init1 == initiator a b m m,
Resp1 == responder a b v,
Resp2 == responder a b M
```

این شبه کلاف شامل سه رشته است، یک رشته آغازگر و دو رشته مخاطب. پس از اجرا رد تولید شده حمله زیر را نمایش می‌دهد:

```
R1.1 A -> B : {M}Ka
R1.2 B -> I(A) : {{M}Ka}Kb
R2.1 I(A) -> B : {{M}Kb}Ka
R2.2 B -> A : {{M}Kb}Ka}Kb
R2.3 A -> B : {{M}Kb}Kb
R1.3 I(A) -> B : {{M}Kb}Kb
```

این حمله در دو نشست انجام می‌شود. گذر اول و دوم نشست اول به طور معمول انجام می‌شود، با این تفاوت که در گذر دوم، نفوذی، خود را به جای طرف A قرار می‌دهد. نشست دوم را نفوذی که خود را A جا زده است، با فرستادن پیامی که از گذر دوم نشست اول به دست آورده است، آغاز می‌کند (توجه کنید که عمل رمزکردن دارای خاصیت جابجایی است). گذرهای دوم و سوم نشست دوم مطابق پروتکل انجام می‌شود. نهایتاً در گذر سوم نشست اول، نفوذی پیامی را که از گذر سوم نشست دوم به دست می‌آورد را به طرف B می‌فرستد. در این حمله نیز مانند حمله قبل، پس از انجام نشست اول، B گمان می‌کند که پیام محرمانه، $\{M\}Kb$ است در حالی که پیام محرمانه، M است. بنابراین این حمله نشان می‌دهد پروتکل، ویژگی تصدیق اصالت پیام را حفظ نمی‌کند.

۸- نتیجه‌گیری

در روش حل قیود [۱، ۳، ۴، ۵] به دلیل اکتفا به مدل دولو-بائو برای مهاجم و به تبع آن، عدم مدل سازی خواص جبری عملیات رمزنگاری، حتی پروتکل ساده‌ای چون پروتکل سه مرحله ای شامیر-ریوست-ادلمن قابل توصیف و تحلیل نمی‌باشد.

در این مقاله روش حل قیود را توسعه داده و برای واری پروتکل‌های امنیتی دو قابلیت مهم را به آن اضافه کرده ایم. اولین قابلیت افزودن عملیات رمزگذاری با قابلیت جابجایی است که ما را در توصیف و تحلیل پروتکل‌های امنیتی در یک سطح انتزاع جزئی تر توانا می‌سازد. دومین قابلیت استفاده از متغیر در توصیف ویژگی‌های امنیتی برای شناسایی حمله احتمالی است. حمله ای که با جایگذاری مقدار محرمانه فرستاده شده توسط فرستنده با مقداری به غیر از مقادیر ثابت، ممکن است صورت گیرد. صحت و پایان پذیری الگوریتم حل قیود بهبود یافته با افزودن این قابلیت‌ها را اثبات کردیم؛ به علاوه، روش پیشنهادی را در ابزار $TOY-CS$ پیاده سازی و با استفاده از آن، چهار حمله را بر ضد پروتکل شامیر-ریوست-ادلمن شناسایی کردیم. تا آنجا که می‌دانیم دو حمله ناقض ویژگی تصدیق اصالت پیام برای اولین بار گزارش می‌شود.

```
18.[a, (msg send [m :^^ (se a)]),
19.[a, (msg recv [m :^^ (se a) :^^ (se e)]),
20.[a, (msg send [m :^^ (se e)]),
21.]
```

که بیانگر حمله زیر است:

```
1. A -> I(B) : {M}Ka
2. I(B) -> A : {{M}Ka}Ke
3. A -> I(B) : {M}Ke
```

در این حمله که در یک نشست انجام می‌شود، نفوذی، خود را B جا می‌زند و در گذر دوم پیام فرستاده شده در گذر اول را با کلید محرمانه خود رمز می‌کند و به طرف A بر می‌گرداند. در گذر سوم پس از رمزگشایی از پیام گذر دوم توسط طرف A، مقدار محرمانه که با کلید نفوذی رمز شده است به نفوذی فرستاده می‌شود.

تصدیق اصالت پیام

در این قسمت دو حمله ناقض ویژگی تصدیق اصالت پیام را، تا آنجا که می‌دانیم برای اولین بار، شناسایی می‌کنیم.

برای یافتن اولین حمله ناقض ویژگی تصدیق اصالت پیام، شبه کلاف را به صورت زیر در نظر می‌گیریم (برای اختصار از نمادهای $a1, a2, a, b1, b, a, m$ ، به ترتیب به جای $atm\ a, atm\ b, atm\ a, atm\ b1, atm\ a2, atm\ a1, atm\ b$ و v, B ، M ، به ترتیب به جای $var\ v, var\ B, var\ M$ ، استفاده شده است):

```
semibundle2 =
([[a1,Init1],[a2, Init2],[b1,Resp1]],
(neg (learn v))) <==
Init1 == initiator a B m m,
Init2 == initiator a B v v,
Resp2 == responder a b M
```

در این شبه کلاف، v متغیر ویژه و a, b و m مقادیر ثابتند که به ترتیب بیانگر آغازگر، مخاطب و «مقدار محرمانه فرستاده شده در یک نشست پروتکل» هستند. این شبه کلاف شامل سه رشته است: دو رشته بیانگر نقش آغازگر و یک رشته بیانگر نقش مخاطب.

با اجرای برنامه رد تولید شده نشان دهنده حمله زیر است:

```
R1.1 A -> B : {M}Ka
R1.2 B -> I(A) : {{M}Ka}Kb
R2.1 A -> I(B) : {M'}Ka
R3.1 I(A) -> B : {{M}Kb}Ka
R3.2 B -> I(A) : {{M}Kb}Ka}Kb
R2.2 I(B) -> A : {{M}Kb}Ka}Kb
R2.3 A -> I(B) : {{M}Kb}Kb
R3.3 I(A) -> B : {{M}Kb}Kb
R1.3 I(A) -> B : {{M}Kb}Kb
```

این حمله در سه نشست انجام می‌شود. گذر اول و دوم نشست اول به طور معمول انجام می‌شود، به جز اینکه در گذر دوم نفوذی خود را A جا می‌زند. نشست دوم را نیز طرف A آغاز می‌کند. در این نشست پیام رمز شده $\{M'\}Ka$ به نفوذی، که خود را به جای B جا زده است فرستاده می‌شود. نشست سوم را نفوذی که خود را A جا زده است، شروع می‌کند. در این نشست پیام $\{M\}Ka$ که در گذر دوم نشست اول به دست آمد به طرف B فرستاده می‌شود (به این نکته توجه شود که عمل رمزکردن دارای خاصیت جابجایی است). در گذر دوم نشست سوم، پیامی که در گذر اول نشست سوم دریافت کرده است را با کلید محرمانه خود رمز می‌کند و به نفوذی، که خود را A جا زده است می‌فرستد. در گذر دوم نشست دوم، نفوذی که خود را B جا زده است، پیامی را در گذر دوم نشست سوم به دست آورده است را به A می‌فرستد. در گذر سوم نشست دوم، پیامی را که در گذر دوم نشست دوم به دست آورده است را با کلید محرمانه خود رمزگشایی می‌کند و پیام $\{M\}Hb\}Kb$ را به نفوذی که خود را B جا زده است می‌فرستد. سپس نفوذی که خود را A جا زده است، در گذر

Analyzer", *Electronic Notes in Theoretical Computer Science*, vol. 171, pp. 23-36, 2007.

[11] S. A. Modersheim, *Models and Methods for the Automated Analysis of Security Protocols*, PhD Thesis, Freiburg ETH Zurich, p. 205, 2007.

[12] M. Turuani, "The CL-Atse Protocol Analyser", In *Proceedings of the RTA 2006*, pp. 277-286, vol. LNCS 4098, 2006.

[13] F. Baader, and W. Snyder, *Unification Theory, Handbook of Automated Reasoning*, pp. 445-532, 2001.

[14] J. H. Siekmann, "Matching under commutativity", In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, Springer-Verlag, vol. LNCS 72, pp. 531-545, 1979.

[15] P. Arenas, A. Fernández, A. Gil, F. J. López, M. Rodríguez, and F. Sáenz, "TOY, A Multiparadigm Declarative Language Version 2.3.1", p. 264, October 2007, R. Caballero and J. Sánchez (Eds.), Available at <http://www.fdi.ucm.es/profesor/fernan/TOY/>.

[16] G. Penn, and M. Haji-Abdolhosseini, "ALE The Attribute Logic Engine User's Guide with TRALE extensions, version 4.0 Beta", p. 219, University of Toronto, Toronto December 2003, Available at <http://www.ale.cs.toronto.edu/docs/>.



سعید جلیلی مدرک کارشناسی ارشد مهندسی نرم‌افزار را در سال ۱۳۶۴ از دانشگاه صنعتی شریف و دکترای هوش مصنوعی را در سال ۱۳۷۰ از دانشگاه برادفورد انگلستان دریافت کرد. ایشان از سال ۱۳۷۱ عضو هیات علمی دانشکده مهندسی برق و کامپیوتر دانشگاه تربیت مدرس است و هم‌اکنون دانشیار مهندسی کامپیوتر می‌باشد. زمینه‌های پژوهشی ایشان واریسی رسمی حین اجرای نرم‌افزار، تشخیص ناهنجاری در شبکه‌های اقتصایی، طراحی نرم‌افزار جستجو بنیان، و محاسبات نوری است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

sjalili@modares.ac.ir



سیدمهدی سجادی مدرک کارشناسی خود را در سال ۱۳۸۱ از دانشگاه صنعتی اصفهان در رشته ریاضی کاربردی و مدرک کارشناسی ارشد خود را در سال ۱۳۸۶ در رشته مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه تربیت مدرس اخذ نموده و هم‌اکنون عضو هیات علمی دانشگاه آیت‌الله العظمی بروجردی است. زمینه تحقیقاتی مورد علاقه ایشان امنیت شبکه و روش‌های رسمی می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

m_sajjadi@modares.ac.ir

از سه ابزار تحلیل خودکار پروتکل که از عملیات جبری پشتیبانی می‌کنند، تنها ابزار OFMC [۱۱] قادر به شناسایی حمله ناقص ویژگی محرمانگی بر ضد پروتکل شامیر-ریوست-ادلمن است، اما هیچ حمله ناقص ویژگی تصدیق اصالت پیام، با این ابزار، گزارش نشده

افزودن عملیات جبری دیگر مانند XOR، استفاده از متغیر در توصیف سایر ویژگی‌های امنیتی، استفاده از نوع برای هر یک از اجزای ثابت پیام در جایی که در پیاده‌سازی پروتکل‌ها برای اجزای پیام، نوع در نظر گرفته شده باشد، و تحلیل پروتکل‌های امنیتی دیگر در یک سطح انتزاع جزئی‌تر، از جمله کارهای آینده در این مقوله است.

مراجع

[1] J. Millen, and V. Shmatikov, "Constraint solving for bounded-process cryptographic protocol analysis," In *Proceedings of the 8th ACM Conference on Computer and Communication Security*, pp. 166-175. ACM SIGSAC, November. 2001.

[2] F. Thayer Fabrega, J. Herzog, and J. Guttman, "Strand spaces: Proving security protocols correct," *Journal of Computer Security*, vol. 7, pp. 191-230, 1999.

[3] R. Corin, A. Saptawijaya, and S. Etalle, "PS-LTL for Constraint-Based Security Protocol Analysis," In *Proceedings of the 21st Int. Conf. on Logic Programming (ICLP)*, Springer-Verlag, vol. LNCS 3668, pp. 439-440, Oct. 2005.

[4] R. Corin, *Analysis Models for Security Protocols*, PhD Thesis, University of Twente, p. 191, 2006.

[5] R. Corin, S. Etalle, and A. Saptawijaya, "A Logic for Constraint-based Security Protocol Analysis," In *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, California, IEEE Computer Society, pp. 155-168, May. 2006.

[6] L. C. Pualson, "Mechanized proofs for a recursive authentication protocol," In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pp. 84-95, June 1997.

[7] V. Cortier, S. Delaune, and P. Lafourcade, "A survey of algebraic properties used in cryptographic protocols", *Journal of Computer Security*, vol. 14, no. 1, pp. 1-43, 2006.

[8] H. Comon-Lundh, and V. Shmatikov, "Intruder Deductions: Constraint Solving and Insecurity Decision in Presence of Exclusive or", In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*, pp. 271-280, 2003.

[9] Y. Chevalier, R. Kuesters, M. Rusinowitch, and M. Turuani, "An NP Decision Procedure for Protocol Insecurity with XOR", In *Proceedings of the Logic in Computer Science Conference LICS'03*, pp. 261-270, June 2003.

[10] S. Escobar, C. Meadows, and J. Meseguer, "Equational Cryptographic Reasoning in the Maude-NRL Protocol

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۰/۲/۲۵

تاریخ اصلاح: ۹۰/۱۲/۱۰

تاریخ قبول شدن: ۹۰/۱۲/۲۵

نویسنده مرتبط: دکتر سعید جلیلی، دانشکده مهندسی برق و کامپیوتر،
دانشگاه تربیت مدرس، تهران، ایران.

-
- ¹ Dolev-Yao
 - ² Strand Space Model
 - ³ Parametric Strands
 - ⁴ Constraint Solving
 - ⁵ Self-Cancellation
 - ⁶ Shamir-Rivest-Adleman Three-Pass Protocol
 - ⁷ Substitution
 - ⁸ Event
 - ⁹ Role
 - ¹⁰ Parametric Strands
 - ¹¹ Strand
 - ¹² Semibundle
 - ¹³ Multiset
 - ¹⁴ Ground Term
 - ¹⁵ Reduction
 - ¹⁶ Trace
 - ¹⁷ Commutative Unification
 - ¹⁸ Termination Measure
 - ¹⁹ Expansion Measure
 - ²⁰ Closed

بهینه‌سازی سبد سهام با محدودیت به کمک شبکه عصبی بر روی گرید

علی غفاری‌نژاد محمد کاظم اکبری فرهنگ نصرت‌ماکویی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

چکیده

مسئله انتخاب سهام بدین صورت است که تعدادی سهام در دسترس می‌باشند و هدف آن است که با انتخاب بهترین روش سرمایه‌گذاری بر روی آنها و لحاظ کمترین ریسک، سود مشخصی حاصل گردد. این مسئله از نوع مسائل بهینه‌سازی در رابطه با ارضای محدودیت‌ها برای انتخاب تعدادی سهام از سبد سهام می‌باشد. روش‌های ریاضی برای یافتن راه حل بهینه مسئله انتخاب سبد سهام استاندارد یا فاقد محدودیت، براساس ریسک/بازده که توسط مارکوویتز ارائه شده است، وجود دارد. اضافه کردن بعضی محدودیت‌ها به مدل استاندارد مشکل است و باعث می‌شود این الگوریتم‌ها کارایی خود را از دست بدهند و نیاز به معرفی الگوریتم‌های جدیدی برای انتخاب بهینه سبد سهام داشته باشیم. در این مقاله روشی جدید برای حل مسئله بهینه‌سازی سبد سهام استاندارد و بهینه‌سازی سبد سهام همراه با محدودیت با استفاده از شبکه عصبی ریاضی ارائه می‌گردد. محدودیت‌هایی که مد نظر قرار می‌دهیم عبارتند از (۱) حد بالا و پایین در انتخاب وزن هر سهم در سبد سهام و (۲) محدودیت در تعداد سهام‌های موجود در سبد سهام. محدودیت سوم هم این است که مجموع وزن سهام‌های موجود در سبد سهام برابر یک گردد. در قدم بعدی گرید محاسباتی برای افزایش سرعت الگوریتم پیشنهادی استفاده می‌گردد. نتایج پیاده‌سازی الگوریتم، نشان می‌دهد مدل ارائه شده قادر به انتخاب بهینه سبد سهام دارای محدودیت با شیوه‌ای کارآمد و در زمان کوتاه می‌باشد.

کلمات کلیدی: بهینه‌سازی، سبد سهام، شبکه عصبی، جبهه کارا، گرید.

۱- مقدمه

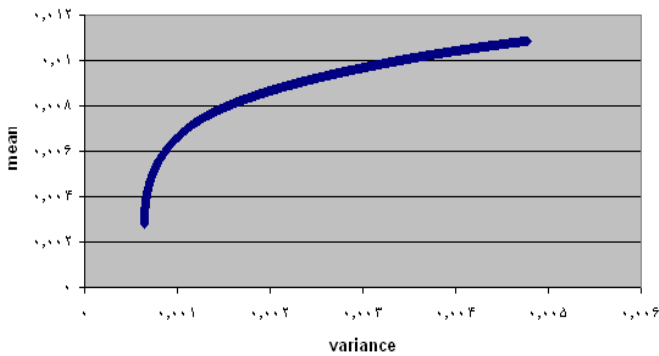
عایدی مورد انتظار^۲ به صورت میانگین وزنی عایدی هر یک سهام تشکیل دهنده سبد سهام محاسبه می‌شود. وزن هر کدام از سهام نسبت سرمایه‌گذاری بر روی آن سهم در سبد سهام است. معادله (۱) فرمول محاسبه عایدی مورد انتظار سهام را نشان می‌دهد.

$$\mu_p = E\left(\sum_{i=1}^n w_i \rho_i\right) = \sum_{i=1}^n w_i \rho_i \quad (1)$$

واریانس یک سبد سهام از معادله (۲) محاسبه می‌شود. با استفاده از روش‌های تحلیلی می‌توان مدل استاندارد ریسک/عایدی مورد انتظار را کاملاً تحلیل نمود. مشکل این مدل در اینست که هیچ محدودیتی را در نظر نمی‌گیرد [۲]. مسئله‌ای که ما در این مقاله مدنظر قرار می‌دهیم بدست آوردن سبد سهام بهینه با در نظر گرفتن محدودیت‌هایی بدین شرح است:

پس از آنکه هاپفیلد در سال ۱۹۸۲ مدل شبکه عصبی خود را ارائه نمود، این مدل با موفقیت در زمینه‌های مختلف بهینه‌سازی مانند مسئله فروشنده دوره‌گرد، چند و زیر و زمان‌بندی منابع بکار گرفته شده است [۱]. بسیاری از سهامداران یک سبد سهام را انتخاب می‌کنند تا ریسک سرمایه‌گذاری را کاهش دهند. در مسئله بهینه‌سازی سبد سهام با محدودیت، فرض بر این است که تعدادی سهام برای انتخاب وجود دارند. هدف ما انتخاب بهینه یک سبد سهام با توجه به ریسک و عایدی مورد انتظار است. سبدهای سهامی که به ازای عایدی مورد انتظار مشخص، کمترین ریسک را دارا هستند جبهه‌های تشکیل می‌دهند که به نام جبهه کارا^۱ شناخته می‌شود [۲، ۳].

حل مسئله بهینه‌سازی پورتفولیوی با محدودیت مورد استفاده قرار دادیم در شکل ۱ آورده شده است.



شکل ۱- جبهه کارای استاندارد داده‌های تست

۳- انتخاب سبد سهام با محدودیت

فرض کنید علاوه بر پارامترهای تعریف شده در بخش ۲ داشته باشیم:

K تعداد سهامهای مورد دلخواه در سبد سهام

ε_i حداقل درصد یک سهم در سبد سهام

δ_i حداکثر درصد یک سهم در سبد سهام

با توجه به مفروضات مسئله داریم: $0 \leq \varepsilon_i \leq \delta_i \leq 1$

$z_i = 1$ در صورتیکه سهم i ام در سبد سهام موجود باشد، و

$z_i = 0$ در صورتیکه سهم i ام در سبد سهام موجود نباشد. در اینصورت مسئله

انتخاب پورتفولیوی با محدودیت بصورت زیر تعریف می‌شود:

$$\min \lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - (1-\lambda) \left[\sum_{i=1}^N w_i \mu_i \right]$$

$$\text{subject to } \sum_{i=1}^N w_i = 1$$

$$\sum_{i=1}^N z_i = K$$

$$\varepsilon_i z_i \leq w_i \leq \delta_i z_i \quad i = 1, \dots, N$$

$$z_i \in [0, 1] \quad i = 1, \dots, N$$

(۴)

مسئله فوق یک مسئله ترکیبی از برنامه‌ریزی خطی و برنامه‌ریزی درجه دوم است که الگوریتم کارایی برای حل آن وجود ندارد [۲، ۴].

چانگ در سال ۱۹۹۹ الگوریتم‌های ژنتیک، جستجوی ممنوع و آنیلینگ شبیه‌سازی شده را برای حل این مسئله بکار برد و نتایج این روش‌ها را با یکدیگر مقایسه نمود [۲]. فرناندز نیز در سال ۲۰۰۴ الگوریتمی مبتنی بر شبکه عصبی هاپفیلد را برای حل مسئله پیشنهاد نمود. فرناندز این محدودیت را که مجموع وزن‌ها در سبد سهام باید برابر یک باشند را لحاظ نکرده بود و برای ارضای این محدودیت از الگوریتم حریمانه معرفی شده توسط چانگ، استفاده نموده بود [۴].

در این مقاله ما قصد داریم با بهبود مدل بکار رفته توسط فرناندز، محدودیت فوق را در شبکه هاپفیلد اعمال کنیم. در انتها صحت عملکرد مدل ارائه شده را مورد بررسی قرار خواهیم داد.

۱- تعداد سهام‌های موجود در سبد سهام برابر عدد K باشد

۲- وزن هر سهم در سبد سهام در محدوده یک مقدار کمینه و یک مقدار

بیشینه قرار گیرد.

$$\sigma_p^2 = E(|\rho_p - \mu_p|^2) = E\left(\left|\sum_{i=1}^n w_i (\rho_i - \mu_i)\right|^2\right) =$$

$$E\left(\left(\sum_{i=1}^n w_i (\rho_i - \mu_i)\right)\left(\sum_{j=1}^n w_j (\rho_j - \mu_j)\right)\right) =$$

$$\sum_{i,j=1}^n w_i w_j E((\rho_i - \mu_i)(\rho_j - \mu_j)) =$$

$$\sum_{i,j=1}^n w_i w_j \sigma_{i,j}$$

(۲)

کارهای قبلی انجام شده در این زمینه، مسئله فوق را با استفاده از الگوریتم‌های تکاملی، جستجوی ممنوع^۳، آنیلینگ شبیه‌سازی^۴ شده و شبکه هاپفیلد^۵ حل کرده‌اند. ما در این مقاله روش جدیدی مبتنی بر شبکه هاپفیلد برای حل مسئله فوق ارائه می‌دهیم.

تفاوت روش ما با شبکه هاپفیلدی که فرناندز و گومز در مرجع [۲] ارائه نموده‌اند در اینست که آنها این محدودیت را که مجموع وزنی سهام‌ها در سبد سهام باید برابر یک باشد را در نظر نگرفته‌اند و برای ارضای این شرط از الگوریتم حریمانه چانگ [۳] استفاده کرده‌اند. در این مقاله ما از روش ریاضی شبکه‌های عصبی که در مرجع [۱] معرفی شده است استفاده کرده‌ایم.

۲- انتخاب سبد سهام

در ابتدا مدل ریسک/عایدی مورد انتظار برای انتخاب پورتفولیوی که توسط مارکوویتز معرفی گردید، ارائه می‌شود [۳]. فرض کنید N تعداد سهام‌های مختلف، μ_i عایدی مورد انتظار برای سهام i ام و σ_{ij} کواریانس بین عایدی سهام‌های i و j باشد. ضمناً فرض کنید $\lambda \in [0, 1]$ پارامتری باشد که وزن ریسک و عایدی مورد انتظار را در انتخاب سبد سهام بهینه تعیین می‌کند. مدل استاندارد ریسک/عایدی مورد انتظار برای مسئله انتخاب سبد سهام عبارتست از:

$$\min \lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - (1-\lambda) \left[\sum_{i=1}^N w_i \mu_i \right]$$

$$\text{subject to } \sum_{i=1}^N w_i = 1$$

$$0 \leq w_i \leq 1 \quad i = 1, \dots, N$$

(۳)

حالتی که $\lambda = 0$ ، وضعیتی را نشان می‌دهد که قصد داریم بدون در نظر گرفتن ریسک، عایدی مورد انتظار را بیشینه کنیم. حالتی که در آن $\lambda = 1$ است نیز نشان دهنده وضعیتی است که ما بایلیم بدون در نظر گرفتن عایدی مورد انتظار، ریسک را کمینه نماییم. مسئله انتخاب سبد سهام بهینه جزء مسائل بهینه‌سازی چندهدفه است.

در این مسائل، یک راه‌حل را بهینه گوئیم اگر هیچ راه حل امکان‌پذیر دیگری وجود نداشته باشد که یک هدف را بهبود ببخشد مگر آنکه باعث بدتر شدن هدف/اهداف دیگر گردد [۴، ۸].

معمولاً مسائل بهینه‌سازی چندهدفه بجای یک جواب، دارای یک مجموعه جواب هستند که به این مجموعه جواب، جبهه کارا اطلاق می‌گردد. بعنوان مثال جبهه کارای مدل استاندارد ریسک/عایدی مورد انتظار برای داده‌هایی که ما برای

۴- شبکه عصبی هاپفیلد

حل مسئله سبد سهام نشان دهیم مسئله را با دو محدودیت حل می‌کنیم. اولین محدودیت این است که مجموع وزن‌ها باید برابر یک باشد. محدودیت دوم این است که وزن هر یک از سهام باید بین صفر و یک باشد. برای حل مسئله انتخاب بهینه پورتفلیو باید تابع هدف را معادل تابع انرژی شبکه هاپفیلد قرار دهیم تا وزن‌های شبکه عصبی بدست آیند. تابع هدف ما طبق رابطه زیر تعریف می‌شود.

$$E_T = A\lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - A(1-\lambda) \left[\sum_{i=1}^N w_i \mu_i \right] + B \left(\sum_{i=1}^N w_i - 1 \right)^2 \quad (10)$$

در رابطه فوق، $B \left(\sum_{i=1}^N w_i - 1 \right)^2$ محدودیتی را بیان می‌کند که در آن باید مجموع وزن هر کدام از سهام برابر یک باشد. بدیهی است هر قدر مجموع وزنهای سهام به یک نزدیکتر باشد، E_T کوچکتر خواهد بود. از طرفی دیگر هر چقدر مجموع وزن سهام از یک فاصله داشته باشد مقدار E_T افزایش خواهد یافت و در نتیجه مقدار تابع انرژی افزایش می‌یابد. در ادامه برای تطبیق این تابع انرژی با رابطه (۹)، جمله آخر آن را باز می‌کنیم و به رابطه (۱۱) می‌رسیم.

$$E_T = A\lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - A(1-\lambda) \left[\sum_{i=1}^N w_i \mu_i \right] + B \left(\sum_{i=1}^N w_i \right)^2 - 2B \left(\sum_{i=1}^N w_i \right) + B \quad (11)$$

با برخی عملیات ساده ریاضی (ترکیب جمله سوم و چهارم)، رابطه فوق به رابطه (۱۲) تبدیل خواهد شد:

$$E_T = A\lambda \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j \sigma_{ij} \right] - A(1-\lambda) \left[\sum_{i=1}^N w_i \mu_i \right] + B \left(\sum_{i=1}^N \sum_{j=1}^N w_i w_j \right) - 2B \left(\sum_{i=1}^N w_i \right) + B \quad (12)$$

در نهایت برای تطبیق رابطه فوق با رابطه (۹) پس از ساده‌سازی به رابطه (۱۳) را خواهیم رسید. که رابطه (۱۳) بسیار شبیه رابطه (۹) است.

$$E_T = \left[\sum_{i=1}^N \sum_{j=1}^N w_i w_j (A\lambda \sigma_{ij} + B) \right] - \left[\sum_{i=1}^N w_i (A\mu_i - A\lambda \mu_i + 2B) \right] + B \quad (13)$$

آخرین جمله در رابطه (۱۳) اثری در تابع مینیمم‌کننده ندارد زیرا یک مقدار ثابت است. با برابر قرار دادن جملات اول و دوم رابطه‌های (۹) و (۱۳) ما می‌توانیم وزن نرونهاي شبکه هاپفیلد را به محاسبه کنیم. وزن نرن‌های شبکه هاپفیلد در رابطه (۱۴) نشان داده شده است.

شبکه عصبی هاپفیلد مدلی است که برای حل مسائل بهینه‌سازی ترکیبی مورد استفاده قرار می‌گیرد. دو نوع مدل شبکه عصبی هاپفیلد که مورد استفاده قرار می‌گیرند مدل گسسته و مدل پیوسته هستند. در مدل گسسته، هر نرون دو وضعیت دارد. ورودی‌ها به هر نرون از دو منبع سرچشمه می‌گیرند. یکی از منابع عبارتست از ورودی‌های خارجی و منبع دیگر عبارتست از سایر نرون‌ها.

$$U_i = \sum_{j=1, j \neq i}^N T_{ij} V_j + I_i \quad (5)$$

در این رابطه U_i کل ورودی به نرون i ، T_{ij} وزن سیگنال ارسالی از نرون j به نرون i ، I_i ورودی خارجی به نرون i و V_j خروجی نرون j است. تابع انرژی شبکه عصبی هاپفیلد بصورت رابطه زیر تعریف می‌شود.

$$E = -\frac{1}{2} \sum_{j=1}^N \sum_{i=1, i \neq j}^N T_{ij} V_i V_j - \sum_{i=1}^N I_i V_i + \sum_{i=1}^N \theta_i V_i \quad (6)$$

تغییر در انرژی ΔE ناشی از تغییر در وضعیت نرون i از رابطه زیر بدست می‌آید.

$$\Delta E = - \left[\sum_{i=1, i \neq j}^N T_{ij} V_i + I_i - \theta_i \right] \Delta V_i \quad (7)$$

در رابطه فوق، ΔV_i تغییرات خروجی نرون i را نشان می‌دهد.

۵- شبکه عصبی هاپفیلد پیوسته

مدل پیوسته شبکه عصبی هاپفیلد بر پایه متغیرهای پیوسته است. تابع ورودی/خروجی یک تابع افزایشنده و پیوسته است. مدل پیوسته، برای حل مسئله انتخاب سبد سهام بهینه مورد استفاده قرار خواهد گرفت.

خروجی هر نرون از روی رابطه $V_i = g_i(U_i)$ بدست می‌آید. $g_i(U_i)$ تابع ورودی/خروجی نرون I است و توسط رابطه زیر بدست می‌آید.

$$g_i(U_i) = \frac{1}{(1 + e^{(-\beta U_i)})} \quad (8)$$

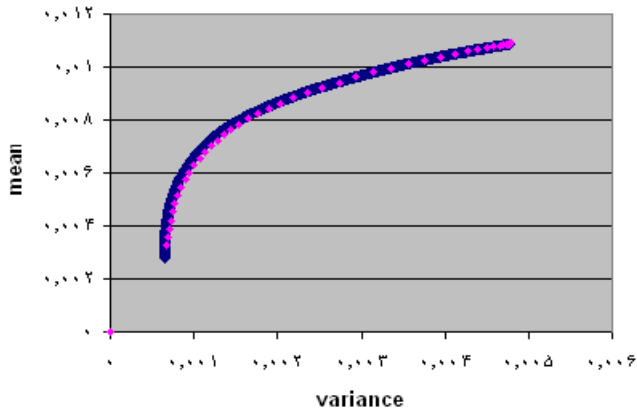
β شکل تابع سیگموئید را مشخص می‌کند.

تابع انرژی شبکه عصبی هاپفیلد طبق رابطه زیر تعریف می‌شود [۵، ۶، ۷].

$$E = -\frac{1}{2} \sum_{j=1}^N \sum_{i=1, i \neq j}^N T_{ij} V_i V_j - \sum_{i=1}^N I_i V_i \quad (9)$$

۶- تطبیق مسئله بر روی شبکه هاپفیلد

حل مسئله بهینه‌سازی با ارضای محدودیت نیازمند انتخاب یک بازنمایی مناسب از مسئله و انتخاب وزن‌های مناسب است. برای اینکه قابلیت شبکه‌های عصبی در



شکل ۲- نتایج بدست آمده روی شاخص Hang Seng

۸- نتیجه‌گیری

همانطور که قبلاً اشاره شد، فرناندز و گومز برای ارضای محدودیت (۵) از الگوریتم حریمانه چانگ استفاده می‌کنند. این الگوریتم بهینه نیست و همیشه دارای جواب نمی‌باشد [۲]. بنابراین با جایگزینی الگوریتمی که در این مقاله معرفی گردید، قادر خواهیم بود اولاً مسئله ارضای محدودیت را توسط شبکه عصبی در نظر بگیریم و نیاز به الگوریتم جداگانه برای این منظور نخواهد بود و ثانیاً مشکلات الگوریتم حریمانه در مدل ارائه شده در این مقاله وجود ندارد.

مراجع

- [1] Y. Takefuji, *Neural Network Parallel Computing*, Kluwer Academic Publishers, 1992.
- [2] A. Fernandez, *Portfolio Selection Using Neural Network*, Computers and Operation Research, 2004.
- [3] T. J. Chang, *Heuristics for Cardinality Constrained Portfolio Optimisation*, Computers And Operation Research, 2000.
- [4] H. Markowitz, "Portfolio Selection," *Journal Of Finance*, 1952.
- [5] F. Fausett, *Fundamentals of Neural Networks*, Prentice Hall, 1994.
- [6] J. H. Parker, "Economic Load Dispatch For Piecewise Quadratic Cost Function Using Hopfield Neural Network," *IEEE Transactions On Power System Apparatus and Systems*, vol 8, no 3, pp.1030-1038, 1993.
- [7] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *proceedings of National Academy Of Science, USA*, vol 81, pp 3088-3092, 1984.
- [8] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods And Applications*, Swiss Federal Institute Of Technology, 1999.

$$\begin{aligned} T_{ij} &= -2(A\lambda\sigma_{ij} + B) \\ I_i &= (A\mu_i - A\lambda\mu_i + 2B) \end{aligned} \quad (14)$$

با توجه به تابع انرژی، معادله حرکت عبارت خواهد بود از:

$$\frac{du_i}{dt} = \sum_{j=1}^N T_{ij}V_j + I_i \quad (15)$$

برای بهنگام سازی هر نرون از رابطه اولر مرتبه اول استفاده می‌شود.

$$u_i(t+1) = u_i(t) + \Delta u_i \quad (16)$$

بدین ترتیب با توجه به معلوم بودن σ_{ij} و μ_i در صورت بدست آوردن مقدار مناسب A و B ، وزن‌های مورد نیاز شبکه بدست خواهد آمد. مقادیر A و B از روی داده‌های تاریخی بدست می‌آیند.

به دلیل اینکه وزن‌های سهام در مسئله به صورت پیوسته تغییر می‌کنند ما نیازمند یک تابع ورودی- خروجی پیوسته هستیم. بنابراین ما می‌توانیم از تابع سیگموئید به عنوان تابع ورودی برای ارضای محدودیت دوم مسئله که در آغاز این بخش به آن اشاره شده استفاده کنیم. این محدودیت وزن هر یک از نرون‌ها محدود می‌کند.

$$v_i = g_i(u_i) = \varepsilon_i + \frac{\delta_i - \varepsilon_i}{1 + \exp(\beta u_i)} \quad (17)$$

این رابطه باعث می‌شود خروجی هر نرون بین مقادیر ε_i و δ_i قرار گیرد. β شیب تابع سیگموئید را مشخص می‌نماید.

۷- نتایج

با توجه به اینکه مسئله انتخاب پورتفلیو بدون محدودیت دارای روش‌های تحلیلی برای حل مسئله است، برای بررسی صحت الگوریتمی که ارائه کرده‌ایم، آنرا برای حل مسئله پورتفلیوی بدون محدودیت، بکار بردیم یعنی با فرض $K=N$ و $\varepsilon_i = 0$ و $\delta_i = 1$ مسئله انتخاب پورتفلیو را با الگوریتم ارائه شده حل نمودیم.

برای اینکار سری داده‌ها مربوط به شاخص Hang Seng هنگ کنگ را مورد استفاده قرار دادیم. داده‌های تست از مسیر <http://mscmga.ms.ic.ac.uk/jeb/orlib/portinfo.html> قابل دستیابی است.

برای این منظور ۵۱ مقدار مختلف λ بین صفر و یک با طول گام ۰.۰۲ انتخاب گردید و برای هر یک از این مقادیر، مسئله انتخاب پورتفلیو با الگوریتم ارائه شده، حل شد و ۵۱ جواب برای این مسئله بدست آمد. شکل ۲ مجموعه جواب‌های بدست آمده برای این مسئله را نشان می‌دهد.

خط پیوسته مجموعه کارا که توسط روش‌های تحلیلی بدست آمده و از طریق آدرس فوق قابل دستیابی است را نشان می‌دهد. نقاط گسسته، مجموعه جواب‌های بدست آمده با توجه به الگوریتم پیشنهادی ما می‌باشد که همانطور که مشاهده می‌شود روی مجموعه کارا قرار گرفته‌اند و این مسئله کارایی الگوریتم پیشنهادی را نشان می‌دهد.



علی غفاری نژاد دارای مدرک کارشناسی مهندسی کامپیوتر گرایش نرم افزار از دانشگاه صنعتی امیرکبیر و کارشناسی ارشد مهندسی فناوری اطلاعات گرایش تجارت الکترونیک از دانشگاه صنعتی امیرکبیر است. وی در حال حاضر دانشجوی دکترا مهندسی فناوری اطلاعات در دانشگاه صنعتی امیرکبیر است.

آدرس پست الکترونیکی ایشان عبارت است از:

ghaffarinejad@aut.ac.ir



محمد کاظم اکبری دارای مدرک دکترا مهندسی کامپیوتر از دانشگاه Case Western Reserve و دانشیار و عضو هیئت علمی دانشگاه امیرکبیر، دانشکده کامپیوتر و فن آوری اطلاعات است. زمینه های پژوهشی ایشان عبارتند از: معماری کامپیوتر، شبکه های عصبی، پردازش موازی، طراحی VLSI، آزمون و تشخیص خطا.

آدرس پست الکترونیکی ایشان عبارت است از:

akbarif@aut.ac.ir



فرهنگ نصرت ماکویی دارای مدرک کارشناسی مهندسی کامپیوتر گرایش سخت افزار از دانشگاه صنعتی شریف و کارشناسی ارشد مهندسی فناوری اطلاعات گرایش تجارت الکترونیک از دانشگاه صنعتی امیرکبیر است. زمینه های پژوهشی مورد علاقه وی عبارتست از: محاسبات نوری، سیستم عامل، شبکه های کامپیوتری، سیستم عامل، امنیت شبکه و سیستم های پایگاه داده.

آدرس پست الکترونیکی ایشان عبارت است از:

makuee@aut.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۰/۲/۲۸

تاریخ اصلاح: ۹۰/۱۲/۱

تاریخ قبول شدن: ۹۰/۱۲/۱۰

نویسنده مرتبط: دکتر محمد کاظم اکبری، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران.

¹ Efficient Frontier

² Expected Return

³ Taboo Search

⁴ Simulated Annealing

⁵ Hopfield

ارائه مدل جدیدی از الگوریتم فازی حرکت دسته جمعی ذرات با افزایش دقت و صحت همگرایی در بهینه‌سازی توابع (FPSO)

محمد فیوضی^۱ جواد حدادنیا^۱ سیدکمال‌الدین موسوی مشهدی^۲ محمد علی‌پور^۳

^۱ دانشکده مهندسی برق و کامپیوتر، دانشگاه حکیم سبزواری، سبزوار، ایران
^۲ دانشکده مهندسی برق، دانشگاه علم و صنعت ایران، تهران، ایران
^۳ دانشکده مهندسی پزشکی، دانشگاه صنعتی چالمرز، چالمرز، سوئد

چکیده

الگوریتم حرکت دسته جمعی ذرات (PSO^۱) با تنظیم مسیر حرکت یک جمعیت از ذرات در فضای مساله بر پایه اطلاعات مربوط به بهترین کارایی قبلی در هر ذره (P_best) و بهترین کارایی قبلی مربوط به همسایگان هر ذره (g_best, N_best) عمل جستجو را در فضای مساله انجام می‌دهد. در این روش هر یک از ذرات سعی می‌کنند به سمت بهترین تجربه‌های فردی و گروهی که در آن نقاط روی داده است همگرا شوند، بدنبال این فرآیند نقطه بهینه تابع هدف مشخص می‌شود. مشکل اصلی الگوریتم PSO استاندارد افتادن در دام بهینه محلی است. تنظیم و کنترل پارامترهای الگوریتم استاندارد حرکت دسته جمعی ذرات (PSO) نقش تعیین کننده و موثری در همگرایی و رسیدن به پاسخ بهینه در حل مسائل پیچیده با دامنه گسترده دارد. در این مقاله با استفاده از منطق فازی روشی برای کنترل پارامترهای (W, C₁, C₂) در الگوریتم PSO ارائه شده است. هر ذره در هر بار اجرای الگوریتم، مقادیر C₁, C₂, W را با توجه به ورودی‌های کنترلر (نرمالیزه تکرار فرایند، تفاوت نقاط بهینه سراسری و محلی در هر تکرار و پارامتر NCBPE) با استفاده از یک سیستم فازی تعیین کرده و در بدست آوردن موقعیت بعدی ذرات از آن استفاده می‌کند. روش پیشنهادی در حل چند مساله با الگوریتم‌های PSO, CPSO, CFPSO مقایسه و نتایج آن ارائه می‌شود. نتایج آزمایش‌ها، برتری روش پیشنهادی به روش‌های مذکور را نشان می‌دهد.

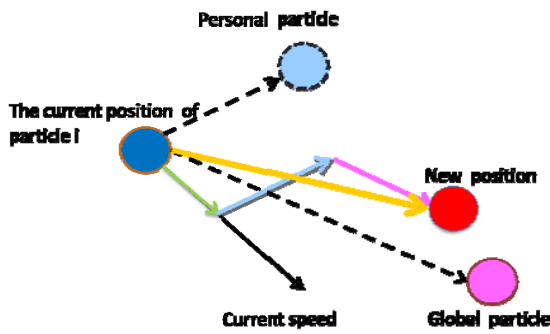
کلمات کلیدی: حرکت دسته جمعی ذرات، منطق فازی، بهینه محلی، بهینه سراسری، همگرایی.

۱- مقدمه

است [۳]. بهینه‌سازی گروه ذرات یکی از تکنیک‌های بهینه‌سازی ابتکاری می‌باشد که بر مبنای جمعیت کار می‌کند. ایده اصلی این روش در سال ۱۹۹۵ توسط دکتر ابرهات و دکتر کندی [۴] مطرح گردید که از رفتار دسته جمعی ماهی و پرندگان برای یافتن غذا الهام گرفته شده بود. فرض بر این است که یک گروه از پرندگان بصورت تصادفی در یک منطقه به دنبال غذا می‌گردند در حالیکه تنها در یک ناحیه جستجو غذا وجود دارد [۵]. پرندگان از مکان غذا اطلاعی ندارند و تنها میزان فاصله خود تا آن محل را می‌دانند [۶]. استراتژی بکار رفته این است که پرندگان بدنبال پرندایی حرکت می‌کنند که نزدیکترین فاصله تا غذا را دارد [۷]. در PSO استاندارد هر جواب مساله یک ذره نام دارد [۸]. هر ذره دارای یک مقدار

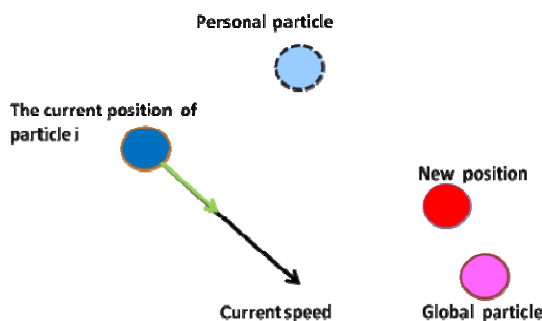
امروزه با بزرگ شدن مسائل و مشخص شدن اهمیت سرعت بالا برای رسیدن به پاسخ و همگرایی، استفاده از الگوریتم‌های جستجوی تصادفی به جای جستجوی همه جانبه در فضای مساله و یا الگوریتم‌های جستجوی هیروستیک^۲ (شهودی)، رشد چشمگیری داشته است [۱]. در مسائل بهینه سازی، یکی از اهداف مهم علاوه بر صحت و دقت نتایج، کاهش زمان جستجو، مهم می‌باشد [۲]. امروزه با بزرگ و پیچیده شدن مسائل، استقبال از روش‌های ابتکاری بطور چشم گیری افزایش یافته

در معادله ۱، W ضریب اینرسی است، برای اولین بار در [۲۳، ۲۴] پیشنهاد شد تا در طول برنامه بطور خطی بین ۰.۹ تا ۰.۴ کاهش یابد. C_1 و C_2 ، میزان تجربه شخصی و گروهی در بازه [۰-۱.۵] انتخاب می‌شود [۳، ۱۷، ۲۴]. دو عدد r_1 و r_2 نیز اعداد تصادفی در بازه [۰-۱] می‌باشد. همچنین مقدار نهایی سرعت هر پرنده، برای جلوگیری از واگرایی الگوریتم به یک بازه محدود می‌شود و V عضوی از دامنه $[-V_{min}, V_{max}]$ می‌باشد. معادله ۲ نیز بردار موقعیت فعلی ذره را با توجه به سرعت جدید آن بروز رسانی می‌کند، نحوه بروز رسانی یک ذره در شکل ۱ نشان داده شده است.



شکل ۱- روند بهینه‌سازی توسط معادلات (۱) و (۲)

حرکت پرندگان بدون قسمت اول معادله ۱، فرایندی خواهد بود که طی آن فضای جستجو به تدریج کوچک می‌شود، اما در مقابل، قسمت اول معادله ۱ باعث حرکت ذرات در مسیر عادی خود خواهد شد. با ترکیب این دو عامل سعی می‌شود تا موازنه‌ای بین جستجوی محلی و سراسری برقرار گردد. برای برقراری بهتر این موازنه برای اولین بار در [۲۳، ۱۷] W پیشنهاد گردید که ضریب حرکت در جستجوی سراسری را مشخص می‌کند، دو پارامتر C_1, C_2 نیز ضریب حرکت در جستجوی محلی را مشخص می‌کنند. W رفته رفته کاهش می‌یابد ولی C_1, C_2 همواره ثابت هستند، بنابراین با ادامه جستجو از میزان جستجوی سراسری کم شده و جستجوی محلی افزایش می‌یابد تا بهینه‌ترین جواب بدست آید. یکی از مشکلات شناخته شده الگوریتم PSO این است که برای پرنده‌ای که شامل اطلاعات g_best است قسمت‌های دوم و سوم معادله ۱، صفر می‌شوند [۳]. بنابراین پرنده در راستای بردار حرکت قبلی خود حرکت خواهد کرد شکل ۲، ذره شامل g_best ثابت خواهد شد و سایر پرندگان نیز به این پرنده همگرا می‌شوند.



شکل ۲- روند بهینه‌سازی الگوریتم با در نظر گرفتن مشکل اول

برای رفع این مشکل ابتدا در سال ۲۰۰۲، الگوریتم جدیدی با نام GCP SO ارائه گردید [۲۵]، در این روش یک پارامتر جدید به الگوریتم اضافه شده بود که

شایستگی است که توسط تابع شایستگی مساله بدست می‌آید [۹]. پرنده‌ای که به غذا نزدیکتر است شایستگی بیشتری دارد [۱۰]. در این زمینه، با توجه به اهمیت مساله، تحقیقات گسترده‌ای انجام شده است. در [۱۱] برای نرم کردن منحنی خروجی الگوریتم، تنها از گزاره‌های منطق فازی به صورت شرطی در بدنه الگوریتم استفاده شده است. توسط محققین [۱۲، ۱۳] از منطق فازی برای بهبود کارایی این الگوریتم در چند تابع محک استفاده شده است. در [۱۴] مدل جدیدی به نام CFPSO³ توسط مولفین ارائه شد که تنها معادله سرعت، توسط توابع عضویت فازی کنترل می‌شود. در [۱۵] از یک الگوریتم پیشنهادی توسط الگوریتم‌های خوشه بندی استفاده شده است، بطوریکه یک تعریف جدید برای همسایگی پیشنهاد می‌شود که در همگرایی نهایی الگوریتم نقش موثری دارد [۱۶]. می‌توان گفت که از جمله مشکلات مهم این الگوریتم یکی افتادن در دام بهینه محلی و دیگری پایین بودن سرعت همگرایی است. در [۱۷] برای اجتناب از افتادن در دام بهینه محلی در حالتی که سرعت پرنده از یک حد مشخص کمتر و شایستگی بدست آمده قابل قبول نباشد، از یک تابع برای ایجاد شک در سرعت پرنده استفاده می‌شود.

در [۱۸] از یک تابع غیر خطی برای تغییر ضریب اینرسی استفاده می‌شود، در صورتی که میزان شایستگی بدست آمده قابل قبول نباشد این تابع، موجب افزایش ضریب اینرسی شده و با افزایش سرعت ذره‌ها امکان جستجوی سراسری بیشتر می‌شود. در [۱۹، ۲۰] سیستم‌های معرفی شده فازی ورودی‌هایی مانند (دفعات تکرار و اختلاف هر ذره از g_best) و خروجی (ضریب اینرسی^۴، شتاب یا دخالت جمعی^۵ و یا فردی^۶) یا ترکیبی از این پارامترها بوده‌اند. در روش پیشنهادی سیستم فازی دارای ۴ ورودی (نرمالیزه تکرار فرایند، تفاوت نقاط بهینه سراسری و محلی در هر تکرار و پارامتر NCBPE) و ۳ خروجی (ضریب اینرسی، دخالت (شتاب) جمعی و فردی) می‌باشد. در بخش دوم این تحقیق الگوریتم PSO استاندارد معرفی می‌شود، سپس در بخش سوم مقدمه‌ای از منطق فازی و کاربردهای آن در زمینه‌های مختلف ارائه می‌شود. در بخش چهارم الگوریتم پیشنهادی ارائه شده تا نهایتاً در بخش پنجم ارزیابی‌های مختلف بر روی الگوریتم پیشنهادی ارائه شود.

۲- الگوریتم PSO استاندارد

PSO استاندارد از جمله الگوریتم‌های جستجوی موازی مبتنی بر جمعیت است که با یک گروه از جواب‌های تصادفی (پرندگان) شروع به کار می‌کند [۲۱]. سپس برای یافتن جواب بهینه در فضای مسئله با به روز کردن مکان پرنده‌ها به جستجو ادامه می‌دهد. هر پرنده به صورت چند بعدی (بسته به نوع مساله) با دو بردار $V_i[t]$ و $X_i[t]$ که به ترتیب معرف موقعیت فعلی و سرعت در لحظه فعلی (t ام) از پرنده نام هستند مشخص می‌شود.

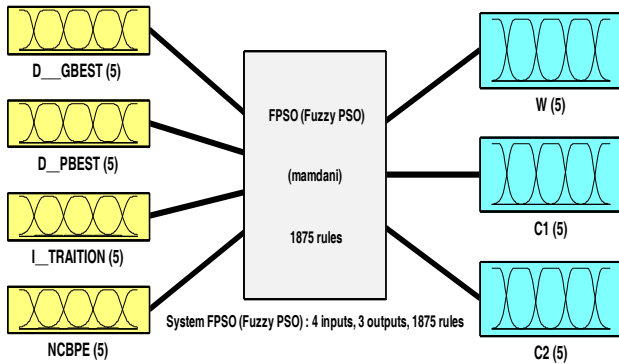
در هر مرحله از حرکت جمعیت، مکان هر پرنده با دو مقدار بهترین به روز می‌شود. اولین مقدار، بهترین تجربه‌ای است که خود پرنده تا به حال بدست آورده است و با p_best نشان داده می‌شود دومین مقدار، بهترین تجربه‌ای است که در بین تمامی پرنده‌ها بدست آمده است و با g_best نشان داده می‌شود. در هر تکرار، الگوریتم بعد از یافتن دو مقدار بالا، سرعت و موقعیت جدید پرنده را براساس معادلات ۱ و ۲ بروز رسانی می‌کند [۲۲].

$$V_i[t+1] = W V_i[t] + C_1 r_1 (x_i^{pbest}[t] - x_i[t]) + C_2 r_2 (x_i^{gbest}[t] - x_i[t]) \quad (1)$$

$$X_i[t+1] = X_i[t] + V_i[t+1] \quad (2)$$

۴-۱- مشکلات الگوریتم

با توجه به مطالب ارائه شده در بخش قبل و آشنایی با الگوریتم پرندگان می‌توان گفت که دو مشکل مهم روش استاندارد یکی گیر افتادن ذرات در دام بهینه محلی و دیگری پایین بودن سرعت همگرایی است. در اینجا روشی را ارائه می‌کنیم که در واقع یک تابع غیرخطی برای ضرب اینرسی، ضرب تجربه فردی و ضرب تجربه گروهی توسط سیستم فازی ایجاد و تنظیم شود. بطور کلی می‌توان سیستم فازی طراحی شده را توسط شکل ۴ نشان داد.



شکل ۴- سیستم فازی طراحی شده

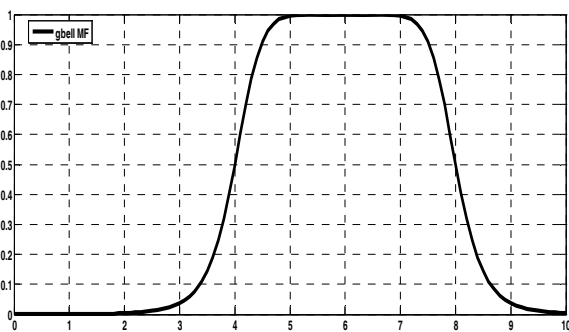
۴-۲- متغیرهای ورودی

متغیرهای ورودی شامل، نرمال شده دفعات تکرار و اختلاف با بهترین بهینه سراسری و گروهی و همینطور پارامتر NCBPE در هر تکرار می‌باشند. برای فازی سازی هر کدام از متغیرهای فازی از توابع زنگوله‌ای بخاطر حساسیت کمتر به نویز طبق معادله ۳ و شکل ۵ استفاده می‌شود.

(۳)

$$f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

در این معادله a نقطه شروع، b نقطه اوج زنگوله و c نقطه مرکز زنگوله می‌باشد.



شکل ۵- تابع عضویت زنگوله‌ای

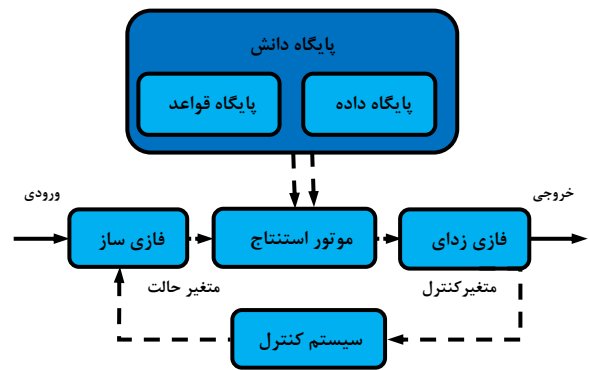
ورودی اول توسط کاربر تعیین می‌شود، بهمین خاطر از نرمال شده‌ی آن بصورت معادله ۴ استفاده می‌کنیم که حاصل عددی بین [0-1] می‌باشد.

باعث جستجوهای تصادفی پرنده best حول جواب بهینه خواهد شد. دومین مشکلی که الگوریتم PSO استاندارد دارد، همگرایی خطی آن می‌باشد که ممکن است بسیار زمان بر باشد [۲۶]. امروزه این الگوریتم در بسیاری از کاربردها از قبیل انتخاب ویژگی [۲۷]، تعیین وزن شبکه‌های عصبی [۲۹]، جایابی قطعات و مسیر یابی ادوات FPGA [۳۰]، طراحی کنترل کننده PID [۳۱]، جایابی ادوات SVC و TCSC در شبکه‌های قدرت استفاده می‌شود [۳۲، ۳۳]. در اینجا از روش ترکیبی الگوریتم PSO استاندارد و منطق فازی برای حل مشکلات مذکور استفاده می‌کنیم.

۳- سیستم‌های فازی

منطق فازی تکنولوژی نسبتاً جدیدی است که در مقابل روش‌های مرسوم برای طراحی و مدل سازی سیستم‌هایی که نیازمند ریاضیات و احتمالات نسبتاً پیشرفته هستند به کار می‌رود. در منطق کلاسیک همه متغیرها در عبارت باینری بیان می‌شوند. منطق فازی گزاره‌های صحیح بولی را با درجه درستی بیان می‌کند. تابع عضویت یک مجموعه فازی، در حالت خاص یک منحنی بوده و نشان دهنده چگونگی نگاشت هر یک از نقاط فضا به درجه عضویت (عددی بین ۰ یا ۱) می‌باشد [۳۴، ۳۵].

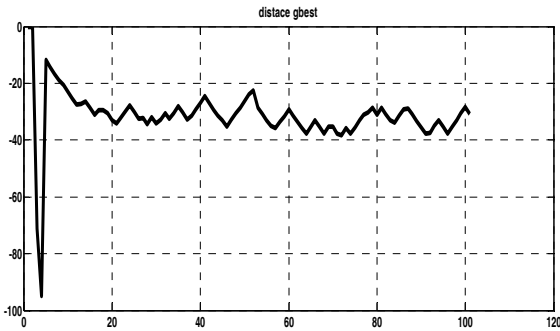
هر سیستم فازی تعدادی قانون دارد که بصورت عبارات IF_THEN در بانک قواعد آن ذخیره می‌گردد. مرحله پردازش که به آن موتور استنتاج گفته می‌شود براساس بانک قواعد فازی عمل می‌کند. ۲ نوع سیستم استنتاج فازی ممدانی و سوگنو وجود دارد. این ۲ نوع سیستم تا حدودی در روش تعیین فازی سازی ورودی‌ها و عملگرهای فازی شبیه هم هستند. اما در روش تعیین خروجی با هم تفاوت دارند [۳۶]. ما در طراحی سیستم مورد نظر، بر حسب نتایج مورد انتظار از سیستم استنتاج ممدانی استفاده کرده‌ایم. تمامی مراحل سیستم فازی در شکل ۳ مشخص شده است.



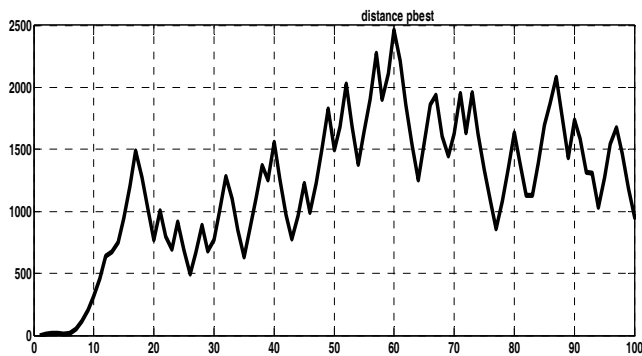
شکل ۳- ساختار یک کنترل کننده فازی

۴- روش ترکیبی فازی یا (Fuzzy PSO) FPSO

در قسمت ۴-۱ ابتدا مساله توضیح داده می‌شود. در قسمت ۴-۲ متغیرهای ورودی‌ها و خروجی‌ها به همراه توابع عضویت شان تعریف می‌شوند و در قسمت ۴-۳ قوانین فازی تعریف و در انتها براساس توابع عضویت، قوانین فازی و موتور استنتاج، خروجی را استخراج کرده و بر روی surface بحث می‌کنیم.



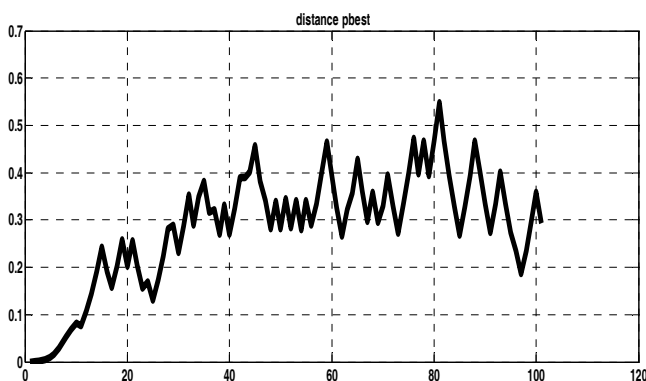
شکل ۷- رفتار تابع نرمال کننده D_g best در هر تکرار در مسیر همگرایی



شکل ۸- رفتار تابع نرمال کننده D_p best در هر تکرار

در معادله ۷ NCBPE^۶ مقدار نرمال "۳" مقدار بهترین، بدترین و حالت فعلی شایستگی تابع می‌باشد که این ورودی نیز به ۵ مجموعه فازی به نام‌های N_FAR, N_NEAR, PROPER, P_NEAR, P_FAR در بازه [0 1] در شکل ۹ مقدار به خود می‌گیرد [۲، ۳۸].

$$NCBPE = \frac{CBPE - CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \quad (۷)$$

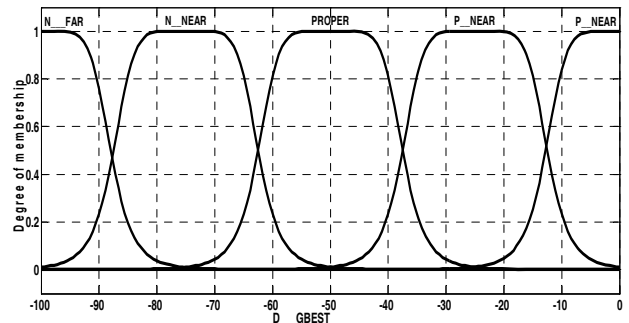


شکل ۹- رفتار تابع NCBPE در هر تکرار در مسیر همگرایی

در الگوریتم بهینه‌سازی ذرات در حالت عادی به خاطر ماهیت الگوریتم سعی بر پیدا کردن نقطه بهینه در صورت پیدا شدن نقاط بهینه سراسری را دارد.

$$\frac{iter(i)}{\max iter} \quad (۴)$$

این ورودی به ۵ مجموعه فازی به نام‌های FIRST, FIRST_MED, MED, END_MED, END به ترتیب به معنی ابتدا، میانه ابتدا، میانه، میانه تا انتها و انتهای تکرار در شکل ۶ تقسیم می‌گردد. البته ناگفته نماند از آنجا که در این سیستم تمام توابع عضویت یک نوع (زنگوله‌ایی) و متغیرهای ورودی و خروجی همه دارای ۵ تابع عضویت هستند از ترسیم این مجموعه‌ها خود داری می‌نماییم. ورودی دوم اختلاف بهینه سراسری (D_gbest) است. در الگوریتم بهینه‌سازی ذرات در حالت عادی به خاطر ماهیت الگوریتم سعی بر پیدا کردن نقطه بهینه در صورت پیدا شدن نقاط بهینه سراسری را دارد.



شکل ۶- تابع عضویت ورودی بعنوان نمونه

اختلاف بهینه سراسری بدست آمده در هر تکرار با تکرار قبلی به سمت صفر میل می‌کند، تا جایی که اختلاف بهینه سراسری هر مرحله با مرحله قبل صفر شود که در این صورت نقطه بدست آمده حتماً بهینه سراسری است. توسط معادله ۵.

$$D_gbest = \frac{1}{N} \sum_{i=1}^N (X_i - X_{gbest}) \quad (۵)$$

مشخص شد که نرمال شده این ورودی در بازه [0 -100] می‌تواند باشد، که دارای ۵ مجموعه فازی با نام‌های N_FAR, N_NEAR, PROPER, P_NEAR, P_FAR در شکل ۷ به ترتیب به معنی منفی دور، منفی نزدیک، مناسب، مثبت نزدیک و مثبت دور توسط توابع عضویت زنگوله‌ایی تقسیم می‌گردد. ورودی سوم اختلاف بهینه محلی (D_pbest) است. این ورودی نیز به ۵ مجموعه فازی به نام‌های N_FAR, N_NEAR, PROPER, P_NEAR, P_FAR در بازه [0 +100] توسط معادله ۶ (شکل ۸ بیانگر رفتار نرمال شده اختلاف بهینه محلی در هر تکرار می‌باشد) توسط توابع عضویت زنگوله‌ایی به ۵ قسمت تقسیم می‌شود.

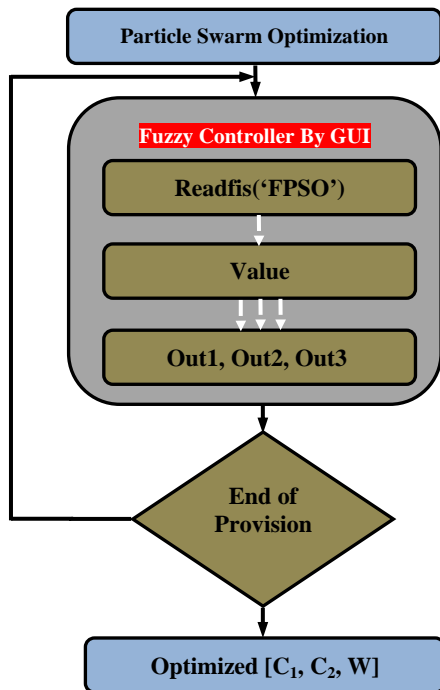
$$D_pbest = \frac{1}{N} \sum_{i=1}^N (X_i - X_{pbest}) \quad (۶)$$

ورودی چهارم پارامتر NCBPE می‌باشد. CBPE میزان شایستگی ذره در حالت فعلی را نشان می‌دهد و CBPE_{min} بهترین شایستگی می‌باشد که ذره تا بحال بدست آورده است و CBPE_{max} بدترین شایستگی هست که ذرات تا بحال به دست آورده‌اند.

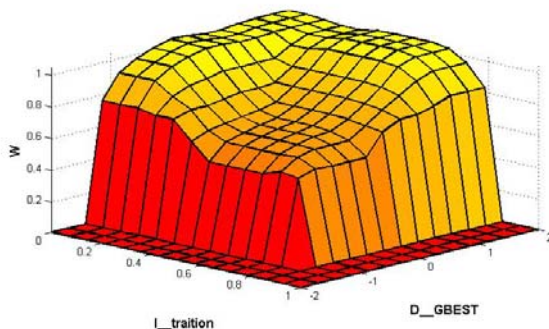
۳-۴- متغیرهای خروجی

سیستم فازی طراحی شده، در استلزام از ممدانی، اشتراک از minimum، در اجتماع از maximum و در عملیات استنتاج از اجتماع نتایج حاصل از استلزام هر یک از قوانین (correlation-minimum) استفاده می‌نماید. همچنین به منظور بدست آوردن یک مقدار Crisp (عددی برحسب ورودی) از دی فازی ساز مرکز ثقل استفاده می‌گردد، این مقادیر به صورت خروجی از طریق تابع evalu مجدداً به الگوریتم برنامه پاس داده می‌شود (شکل ۱۰).

پس از اعمال قوانین براساس توابع عضویت‌ها، surface کنترل بدست می‌آید. هر surface نشان دهنده میزان تغییرات یک خروجی براساس دو ورودی است. در این سیستم به ازای هر خروجی یک surface تعریف می‌شود، البته surface ها در این سیستم همانند یکدیگر هستند که برای اجتناب از تکرار فقط یک surface با توجه به شکل‌های بدست آمده، صحت قوانین و توابع عضویت متغیرهای ورودی و خروجی و نیز نرمی و عملکرد آنها در شکل ۱۱ به نمایش درآمده است.



شکل ۱۰- بلوک دیاگرام اجرایی الگوریتم پیشنهادی



شکل ۱۱- نمایشگر سطح ورودی‌ها براساس خروجی اول

۵- ارزیابی الگوریتم پیشنهادی

در الگوریتم PSO اساس کار بر روی تغییر وضعیت فعلی پرنده می‌باشد. برای بررسی درست الگوریتم‌ها، در نرم افزار MATLAB[®] الگوریتم‌های پیشنهادی

متغیرهای این سیستم ضریب اینرسی (w)، شتاب یا ضرایب یادگیری (C_1, C_2) می‌باشد (شکل ۴). مقادیر بزرگ W امکان جستجوی فرامحلی و مقادیر کوچک آن امکان جستجوی محلی را فراهم می‌کند [۳۷]. در زمینه کنترل ضریب اینرسی با استفاده از سیستم فازی می‌توان به کارهای انجام شده (تا آنجا که مولفین اطلاع دارند) از جمله [۳۸] اشاره کرد که این متغیر در بازه [0-1.2] بصورت استاندارد تعریف می‌شود که به ۵ مجموعه فازی با توابع عضویت زنگوله‌ای به نام‌های $AF(w), AFM(w), ALM(w), AL(w)$ ، به ترتیب به معنی: کم، میانه‌کم، میانه، میان‌زیاد و زیاد تقسیم می‌شود. C_1, C_2 ضرایبی هستند که به ترتیب بیانگر بیشترین گام حرکت به سمت جواب محلی و جواب سراسری می‌باشد. متغیر دوم خروجی سیستم C_1 یا ضریب شتاب یادگیری (تجربه شخصی) است. این متغیر در بازه [0-1, 1.5] بصورت استاندارد تعریف می‌شود، دارای ۵ تابع عضویت با نام‌های $AF(c_1), AFM(c_1), M(c_1), ALM(c_1), AL(c_1)$ ، به ترتیب به معنی: کم، میانه‌کم، میانه، میان‌زیاد و زیاد می‌باشد. متغیر سوم خروجی C_2 می‌باشد، این متغیر در بازه [0-1, 1.5] بصورت استاندارد دارای ۵ تابع عضویت گوسی با نام‌های $F(c_2), AFM(c_2), M(c_2), ALM(c_2), AL(c_2)$ ، به ترتیب به معنی: کم، میانه‌کم، میانه، میان‌زیاد و زیاد می‌باشد. برقراری تعادل بین این ۲ ضریب می‌تواند نقش به‌سزایی در سرعت پاسخ الگوریتم داشته باشد [۳۸]. با استفاده از منطق فازی در این تحقیق ضرایب C_1, C_2, W در هر تکرار الگوریتم کنترل می‌شود.

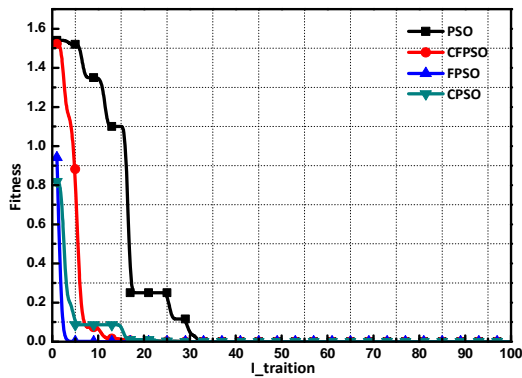
۴-۴- قواعد فازی

مهمترین قسمت از یک سیستم فازی، بانک قواعد آن می‌باشد. با داشتن قوانینی درست، می‌توان یک کنترل مناسب را از سیستم انتظار داشت. در این تحقیق تمام حالات ممکن بین ورودی‌ها و خروجی‌ها در بانک قوانین در نظر گرفته شده است. براساس قاعده Decouple ۲۵ مجموعه تحت عنوان بانک قواعد، هر کدام در ۷۵ قانون ایجاد شد، در کل $1875 (5 \times 5 \times 5 \times 3)$ قانون بدست آمد که در جدول ۱ تنها قواعد بانک اول ارائه شده است. ابتدا مقادیر ورودی توسط الگوریتم حرکت دسته جمعی پرنده‌گان تولید می‌شوند، سپس وارد سیستم فازی می‌شوند.

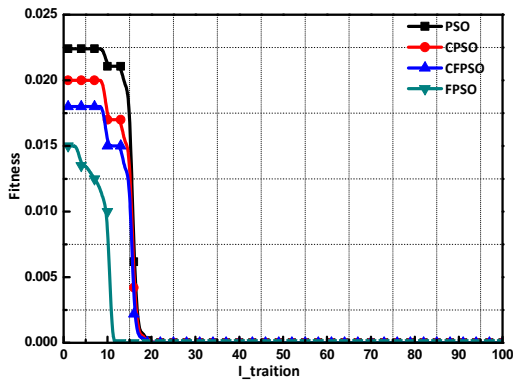
جدول ۱- بانک قواعد با ساختار Index

5 5 1 5, 5 5 1 (1) : 1	4 5 3 5, 3 3 3 (1) : 1
4 5 1 5, 5 4 2 (1) : 1	5 5 3 5, 3 3 3 (1) : 1
3 5 1 5, 5 3 3 (1) : 1	5 5 4 5, 1 2 4 (1) : 1
2 5 1 5, 5 4 2 (1) : 1	4 5 4 5, 2 2 5 (1) : 1
1 5 1 5, 5 5 1 (1) : 1	3 5 4 5, 2 3 3 (1) : 1
5 5 2 5, 4 1 5 (1) : 1	2 5 4 5, 2 2 4 (1) : 1
4 5 2 5, 4 2 4 (1) : 1	1 5 4 5, 2 1 5 (1) : 1
1 5 2 5, 4 5 2 (1) : 1	5 5 5 5, 1 1 5 (1) : 1
2 5 2 5, 4 4 2 (1) : 1	4 5 5 5, 1 1 5 (1) : 1
3 5 2 5, 4 3 3 (1) : 1	3 5 5 5, 1 1 5 (1) : 1
1 5 3 5, 3 3 3 (1) : 1	2 5 5 5, 1 1 5 (1) : 1
2 5 3 5, 3 3 3 (1) : 1	1 5 5 5, 1 1 5 (1) : 1
3 5 3 5, 3 3 3 (1) : 1	

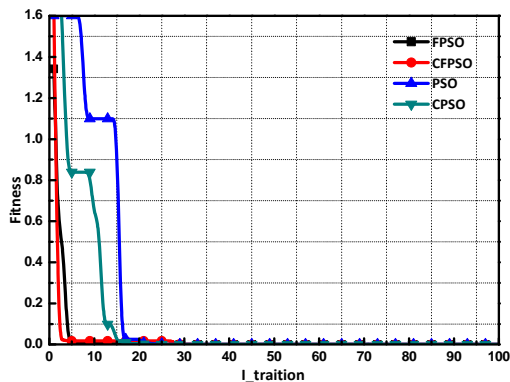
غیرگویایی می‌شود. لذا از نرم‌افزار "ORIGIN[®]" برای ترسیم این نمودارها استفاده شده است براساس نتایج این شبیه‌سازی‌ها برای توابع محک مورد نظر، با ابعاد مختلف (۲۰ و ۵۰ و ۱۰۰ و ۲۰۰)، بهترین جواب و متوسط جواب بدست آمده بیان شده‌اند. محروافقی "I_traction" معرف نقطه همگرایی در تکرار فرآیند سیستم های معرفی شده و محور عمودی مشخص کننده ارزیابی (Fitness) توابع معرفی شده می‌باشند که در نمودارهای (۱ و ۳ و ۴) بیان می‌شوند. با توجه به نمودارهای (۱ و ۳ و ۴) و جداول (۲ و ۳ و ۴) و در نظر داشتن ماهیت تصادفی الگوریتم به خوبی نتیجه گرفته می‌شود که روش ارائه شده براساس سیستم فازی دارای سرعت همگرایی بالاتر در تعداد دفعات تکرار کمتر در مقایسه با الگوریتم استاندارد و الگوریتم‌های بررسی شده می‌باشد. این موضوع به خوبی بهینه بودن روش و مهم‌تر از آن پیاده‌سازی صحیح سیستم فازی و قوانین را بیان می‌کند.



شکل ۱۲- نمودار همگرایی تابع همگرایی Shepherd ۱۰۰ بعدی



شکل ۱۳- نمودار همگرایی تابع همگرایی Rastregin ۱۰۰ بعدی



شکل ۱۴- نمودار همگرایی تابع همگرایی Schawefel ۱۰۰ بعدی

(PSO, CPSO, CFPSO, FPSO (Proposed ALG)) شبیه‌سازی شدند و با استفاده از توابع محک استاندارد مورد ارزیابی قرار گرفته‌اند. این توابع دارای نقاط بهینه محلی فراوان و از طرف دیگر دارای نقطه بهینه سراسری نیز هستند. به همین منظور از ۴ تابع مشهور، Rosenber (۸-۱۱) معرفی شده‌اند استفاده می‌کنیم که هدف مینیمم سازی این توابع در نقطه صفر است.

$$f_{sph}(x) = \sum_{i=1}^{\dim(x)} x_i^2 \quad (8)$$

$$f_{sds}(x) = \sum_{i=1}^{\dim(x)} \sum_{j=1}^i x_j^2 \quad (9)$$

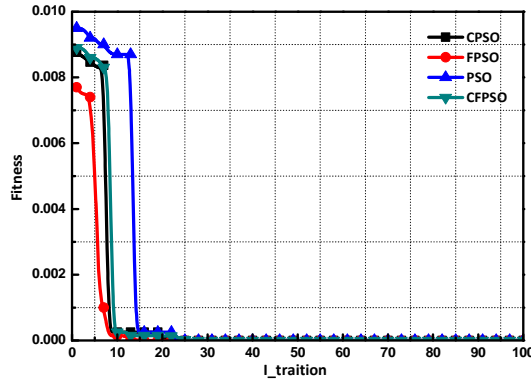
$$f_{ros}(x) = \sum_{i=1}^{\dim(x)-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2] \quad (10)$$

$$f_{ras}(x) = \sum_{i=1}^{\dim(x)} [x_i^2 + 10 - 10\cos(2\pi x_i)] \quad (11)$$

در این ارزیابی، ۵۰ پرند با ۲۰، ۵۰، ۱۰۰، ۲۰۰ بعد (متغیر)، و $C_1=C_2=1.5$ و $W=1.2$ بصورت پیش فرض برای شروع اولیه در نظر گرفته شده‌اند، سپس الگوریتم وارد قسمت فازی شده و از این مرحله تا انتهای الگوریتم کار به صورت فازی انجام می‌شود. بعد از ۱۰۰ بار تکرار نتایج مورد نظر بدست می‌آید. برای مقایسه بهتر شبیه سازی های بالا را ۵۰ بار اجرا کرده، نتایج ارائه شده در جداول (۲ و ۳ و ۴) میانگین نتایج بدست آمده و نمودارهای (۱ و ۳ و ۴) بهترین مقادیر از این ۱۰۰ بار تکرار می‌باشند. این جداول بیانگر مقایسه بین الگوریتم فازی پیشنهادی (FPSO⁸)، الگوریتم PSO استاندارد، الگوریتم CPSO⁹ در [۱۲] و الگوریتم CFPSO³ در [۱۳، ۲۷] می‌باشد. در الگوریتم CPSO [۱۲] با اضافه کردن یک قسمت به معادله ۲، رابطه جدیدی برای سرعت در معادله ۱۱ بدست آمده است.

$$v_{i+1} = k[av_i + c_1 r_1 (x_{ipbest} - x_i)] + \sum \varphi(h) c_2 r_2 (x_{igbest} - x_i) \quad (12)$$

که در این معادله $\varphi(h)$ همان توابع عضویت فازی بوده که در بدنه اصلی برنامه معرفی شده است. الگوریتم معرفی شده در [۱۳، ۲۷] نیز به همین ترتیب با اضافه کردن یک قسمت در معادله ۲، معادله ۱۲ دوباره بدست آمده است، با این تفاوت که در این معادله، قسمت $\varphi(h)$ یا تابع عضویت فازی از یک نوع متفاوت از [۱۲] می‌باشد. برای ارزیابی الگوریتم پیشنهادی از نرم افزار Matlab استفاده شد، قسمت الگوریتم PSO بر پیشنهاد Eberhart نوشته شد، سپس توسط واسط کاربری (GUI) قسمت فازی پیاده‌سازی شد، به منظور ارتباط بین قسمت فازی و واسط کاربری از توابع "Readfis" برای فراخوانی قسمت فازی (با نامگذاری توسط یک متغیر) و Evalfis به منظور برآورد سازی مقادیر محاسبه شده ورودی الگوریتم PSO (چهار ورودی سیستم فازی) مقدار Crisp (مقدار عددی منتج از ورودی‌های PSO و سیستم فازی) بدست می‌آید. سپس با توجه به خروجی‌های سیستم فازی مقادیر مناسب W ، C_1 ، C_2 به الگوریتم PSO به منظور (هدایت صحیح ذرات)؛ ارزیابی تابع شایستگی پاس داده می‌شود. ارزیابی توابع منجر به تولید بهترین مقدار شایستگی توابع و نمودارهایی می‌شود. ترسیم این نمودارها توسط نرم‌افزار Matlab به همراه تابع hold on منجر به نمودارهای نسبتاً



شکل ۱۵- نمودار همگرایی تابع همگرایی Rosenberg ۱۰۰ بعدی

جدول ۲- نتایج همگرایی توابع محک در ۲۰ بعد

Function	FPSO average	FPSO best	CFPSO average	CFPSO best	CPSO average	CPSO best	PSO average	PSO best
F _{Shepher}	0.059	6.66e-16	0.326	3.51e-9	0.118	3.51e-10	0.337	0.337
F _{Rastregin}	0.033	6.66e-16	0.0935	4.82e-11	0.237	2.83e-15	0.225	0.225
F _{Schawefel}	0.019	5.26e-12	0.998	6.35e-2	0.998	5.425e-4	0.156	0.156
F _{Rosenberg}	0.025	4.325e-15	0.860	3.25e-7	2.658	8.352e-5	0.125	0.125

جدول ۳- نتایج همگرایی توابع محک در ۵۰ بعد

Function	FPSO average	FPSO best	CFPSO average	CFPSO best	CPSO average	CPSO best	PSO average	PSO best
F _{Shepher}	0.057	7.24e-10	0.638	5.214e-6	0.0995	5.512e-10	0.558	0.585
F _{Rastregin}	0.046	8.223e-8	0.259	5.326e-9	0.345	3.458e-8	0.325	0.289
F _{Schawefel}	0.119	6.758e-9	1.258	8.548e-5	1.897	7.231e-5	0.875	0.358
F _{Rosenberg}	0.025	5.275e-12	1.325	4.289e-6	3.587	5.736e-4	0.536	3.003

جدول ۴- نتایج همگرایی توابع محک در ۱۰۰ بعد

Function	FPSO average	FPSO best	CFPSO average	CFPSO best	CPSO average	CPSO best	PSO average	PSO best
F _{Shepher}	0.116	8.330e-15	0.038	2.834e-14	0.0258	8.331e-13	0.923	9.283e-13
F _{Rastregin}	0.226	1.546e-12	0.208	1.565e-12	0.548	1.817e-08	0.485	1.6493e-06
F _{Schawefel}	0.230	4.102e-19	0.658	1.58e-15	0.835	1.266e-11	0.975	1.37543e-06
F _{Rosenberg}	0.672	6.221e-13	0.8275	3.17e-09	0.897	5.052e-08	0.776	6.036e-07

جدول ۵- نتایج همگرایی توابع محک در ۲۰۰ بعدی

Function	FPSO average	FPSO best	CFPSO average	CFPSO best	CPSO average	CPSO best	PSO average	PSO best
F _{Shepher}	0.569	4.251e-3	1.0019	3.531e-2	1.564	5.145e-2	1.0563	1.852
F _{Rastregin}	0.437	2.001e-5	0.528	7.564e-3	0.9214	4.553e-3	0.891	0.852
F _{Schawefel}	0.620	5.802e-4	3.5483	1.648e-8	2.654	2.423e-2	0.9915	2.0127
F _{Rosenberg}	0.782	2.587e-6	5.312	8.565e-3	3.423	5.569e-1	0.753	3.645

optimization," *the 3rd International Conference on Machine Learning and Cybernetics*, Shanghai, pp. 398-407, 2004.

[7] B. Al-kazemi, and C. K. Mohan, "Training feed forward neural networks using multi-phase particle swarm optimization," *the 9th International conference on Neural Information*, Bambamie, pp. 25-28, 2002.

[8] D. Merwe, and A. Engelbrecht, *Data clustering using particle swarm optimization*; <http://cirg.cs.up.ac.za/publications/CEC2003d.pdf>.

[9] T. R. Machado, and H. S. Lopes, "A hybrid particle swarm optimization model for the traveling salesman problem," *Adaptive and Natural Computing Algorithms Springer*, vol. 12, no. 10, pp. 255-258, 2005.

[10] V. G. Gudise, and G. K. Venayagamoorthy, "FPGA placement and routing using particle swarm," *Proc. IEEE Int Conf. Computer Society Annual Symposium on VLSI Emerging trends in VLSI Design (ISLVI 2004)*, pp. 514-518, 2004.

[11] T. Niknam, and H. D. Mojarrad, "A new fuzzy adaptive particle swarm optimization for non smooth," *Economic dispatch journal Energy*, vol. 21, no. 3, pp. 1746-1778, 2010

[12] X. Meng, and L. Jia, "A New Kind of PSO Convergent Fuzzy Particle Swarm Optimization and Performance Analysis," *Proc. Fourth Int Conf. Networked Computing and Advanced Information Management*, pp. 1251-1257, 2008.

[13] X. Meng, and L. Jia, "Study on Convergent Fuzzy Particle Swarm Optimization and Performance Analysis," *Proc. The 3rd Int IEEE Conf. Innovative Computing Information and Control (ICICIC'2008)*, pp. 9780-9785, 2008.

[14] X. Meng, and L. Jia, "A New Kind of PSO-Convergent Fuzzy Particle Swarm Optimization and Performance Analysis," *Proc. Fourth Int Conf. Networked Computing and Advanced Information Management*, pp. 25-28, 2008.

[15] M. Alizadeh, E. Fotoohi, V. Roshanae, and E. Safavieh, "Clustering Based Fuzzy Particle Swarm Optimization," *Proc. 28th North American Fuzzy Information Processing Society, Annual Conf. (NAFIPS2009)*, pp 14-17, 2009.

[16] N. S. Niasar, J. Shanbezade, M. M. Perdam, and M. Mohajeri, "Discrete Fuzzy Particle Swarm Optimization for Solving Traveling Salesman Problem," *International Conference on Information and Financial Engineering*, pp. 75-81, Iran, 2009.

[17] X. Cui, "Document, Clustering using particle swarm optimization," *IEEE Journal*, vol. 10, no. 4, pp. 185-191, 2005.

[18] L. Hongbo, and M. Abraham, "Fuzzy Adaptive Turbulent Particle Swarm Optimization," *IEEE Conference*, pp. 39-47, Turkey, 2005.

[19] J. Liebowitz, *The Handbook of Applied Expert Systems*, CRC Press, ISBN: 0849331064, pp. 131-135, 1999.

براساس نمودارهای حاصل از ارزیابی توابع به خوبی دیده می‌شود که ۲ مشکل اصلی الگوریتم استاندارد در ۴ تابع محک به طور قابل ملاحظه‌ای برطرف شده و برتری واقعی روش ارائه شده به الگوریتم استاندارد، الگوریتم CPSO در مرجع [۱۲] الگوریتم CFPSO در مراجع [۱۳، ۲۶] را توسط نمودارهای فوق به اثبات رساندیم.

۶- نتیجه‌گیری

با توجه به نتایج بدست آمده از جداول (۲ و ۳ و ۴ و ۵)، نتیجه می‌گیریم که الگوریتم پیشنهادی در تمامی توابع محک استفاده شده بهترین بوده و اختلاف محسوسی براساس نمودارهای (۱ و ۲ و ۳ و ۴) نسبت به الگوریتم‌های مذکور وجود دارد که برتری الگوریتم پیشنهادی را نشان می‌دهد. این الگوریتم در تشخیص بهینه‌های محلی موفق بوده و توانسته خود را از دام بهینه‌های محلی رها کند و بهینه سراسری را پیدا کند. در بعضی از نمودارها همانند نمودارهای (۴ و ۲) رفتار نسبتاً غیرخطی و حتی قابل تاملی از توابع دیده می‌شود که این بخاطر مقادیر کاملاً تصادفی و غیر قابل پیش‌بینی تولیدی توسط الگوریتم در تلاش برای همگرایی است. می‌توان گفت که استفاده از توابع غیر خطی رفتار الگوریتم استاندارد را با توجه به ماهیت تصادفی‌اش تغییر می‌دهد. از طرفی چون چندین پارامتر در تعیین نقطه بهینه سراسری دخیل می‌باشند، منطق فازی یکی از بهترین روش‌ها برای ایجاد تناسب و تعادل در بین پارامترهای الگوریتم استاندارد حرکت دسته جمعی ذرات برای حل توابع غیر خطی به منظور دستیابی به نقطه بهینه سراسری تابع هدف می‌باشد. اگر چه محققین تمایل زیادی در استفاده از ابزار منطق فازی داشته اما چالش جدی فهم دقیق منطق و پیاده‌سازی قوانین فازی مساله می‌باشد. ترکیب دو ایده هوش جمعی و منطق فازی می‌تواند پاسخی برای این چالش محسوب شود. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم بهینه‌سازی جمعی پرندگان استعداد گیر افتادن در بهینه محلی را دارد. استفاده از منطق فازی به شرط فهم دقیق موضوع و پیاده‌سازی صحیح قوانین فازی می‌تواند صحت پاسخ دقیق مساله را افزایش، و زمان حل را به شدت کاهش دهد.

مراجع

- [۱] م. رستمی شهر بابکی، و م. نظام‌آبادی پور، "روش جدیدی برای الگوریتم pso باینری"، پردازش تصویر، بخش مهندسی برق، دانشگاه شهید بهنر کرمان، چهاردهمین کنفرانس ملی برق ایران (ICE 2006)، ص ۱۲۱-۱۲۰، ۱۳۸۵.
- [۲] م. نورزوی بیرامی، و م. میبیدی، "بهبود بهینه‌سازی گروه ذرات با استفاده از منطق فازی"، دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر، کنفرانس داه‌کاوی (ICDM 2008)، تهران، ص ۶۱-۵۴، ۱۳۸۷.
- [۳] ح. نظام‌آبادی پور، و م. رستمی شهر بابکی، "تعمیمی بر الگوریتم GCBPSO". دوازدهمین کنفرانس مهندسی کامپیوتر ایران، ص ۲۹-۳۵، ۱۳۸۵.
- [4] R. Kennedy, "Eberhart, Particle Swarm Optimization," *IEEE, 33rd Applied Imagery*, pp. 1942-1948, 1995.
- [5] H. A. Firip, and E. Goodman, "Swarmed feature selection," *IEEE Conf of the 33rd Applied Imagery Pattern Recognition, AIPR'04*, pp. 251-259, 2004.
- [6] H. B. Lip, Y. Y. Tang, J. Meng, and Y. Jp, "Neural networks learning using best model particle swarm

Proc. 3rd International Conf. Technical and physical Problems in Power Engineering, pp. 5-9, 2006.

[34] M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier, Book, New York, 1985.

[35] J. Liebowitz, *The Handbook Of Applied Expert Systems*, CRC Press, ISBN: 0849331064, 1999 Krishnamoorthy.

[36] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang, "Improved Catfish Particle Swarm Optimization with Fuzzy Adaptation," *IEEE FUZZY Proceeding conference*, 20-24, KORAE, 2009.

[37] Y. Shi, and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," *IEEE proceedings Conference*, pp. 101-106, 2001.

[38] F. Herrera, and M. Lozano, "Fuzzy adaptive genetic algorithm: design," *Journal of Soft Computing*, vol. 7, no. 4, pp. 545-562, 2009.



محمد فیوضی در حال حاضر در دانشگاه حکیم سبزواری (تربیت معلم سبزواری سابق) در آزمایشگاه پردازش سیگنال، بخش مهندسی پزشکی، دانشکده مهندسی برق و کامپیوتر مشغول به فعالیت می‌باشد. مدرک کارشناسی را از دانشگاه آزاد اسلامی واحد ساوه در رشته مهندسی برق - قدرت در سال ۱۳۸۶ اخذ نموده و در حال حاضر دانشجوی ترم آخر رشته مهندسی پزشکی - بیو الکترونیک در دانشگاه حکیم سبزواری می‌باشد. زمینه‌های حرفه‌ای مورد علاقه ایشان، پردازش سیگنال (تصویرهای دیجیتال و بیولوژیکی)، بازشناسی الگو، شبکه‌های عصبی مصنوعی و تطبیقی فازی عصبی، بهینه‌سازی، منطق فازی، محاسبات نرم و هوش مصنوعی می‌باشد. از ایشان تا به حال تعدادی مقاله در مجلات داخلی، خارجی و کنفرانس‌های مرتبط به چاپ رسیده است. آدرس پست‌الکترونیکی ایشان عبارت است از:

mohammad.fiuzy@yahoo.com



جواد حدادنیایا در حال حاضر استاد دانشکده مهندسی برق و کامپیوتر دانشگاه حکیم سبزواری (تربیت معلم سبزواری سابق) می‌باشد. او مدرک کارشناسی را در رشته مهندسی برق - الکترونیک در سال ۱۳۷۲ و مدارک کارشناسی ارشد و دکتری را از دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی امیرکبیر تهران به ترتیب در سال‌های ۱۳۷۴ و ۱۳۸۱ اخذ نموده است. ایشان از سال ۱۳۸۱ با مرتبه استادیاری در دانشگاه حکیم سبزواری (تربیت معلم سبزواری سابق) مشغول به فعالیت می‌باشند. او در سال‌های ۱۳۸۰-۱۳۸۱ در دانشگاه Windsor در کشور کانادا به عنوان محقق مشغول به فعالیت بوده‌اند. علایق مطالعاتی ایشان شامل: شبکه‌های عصبی مصنوعی، پردازش تصاویر دیجیتال، بینایی ماشین و انسان، بازشناسی الگو، هوش مصنوعی و محاسبات نرم می‌باشد. از ایشان تا به حال شمار زیادی مقالات در ژورنال‌های بین‌المللی به چاپ رسیده است. ضمن اینکه هم‌اکنون سر دبیر مجله هوش مصنوعی در دانشگاه مربوطه می‌باشند، ایشان هم‌اکنون عضو دائم انجمن‌های بین‌المللی IEEE, SPIE, CIPPR, and IEICE می‌باشند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

haddadnia@sttu.ac.ir

[20] C. Krishnamoorthy, and S. Rajeev, *Artificial Intelligence and Expert Systems for Engineers*, CRC Press LLC ISBN: 0849391253, pp. 24-29, 1996.

[21] J. Kennedy, and R. Mendes, "Population Structure and Particle Swarm Performance," *Proc. IEEE Conf. Evol. Computing*, pp. 1671-1676, 2002.

[22] Y. Shi. Eberhart, "Parameter selection in particle swarm optimization," *Proc. Annual Conf. Evolutionary Programming*, pp. 174-78, March 1998.

[23] Y. Shi. Eberhart, "Empirical, study of particle swarm optimization," Conference on Evolutionary Programming Washington DC, pp. 21-27, USA, 6-9 July, 1999.

[24] L. Hongbo, and M. Abraham, "Fuzzy Adaptive Turbulent Particle Swarm Optimization," *IEEE Conference*, pp. 39-47, Slovakia, 2005.

[25] X. Feng, J. Zhang, and Z. Yang, *Adaptive, Particle Swarm Optimization on Individual Level*, IEEE, conference, China, pp. 1215- 1218, 2002.

[26] X. Meng, and L. Jia, "Study on Convergent Fuzzy Particle Swarm Optimization," *Proc. Fourth International Conf. Networked Computing and Advanced Information Management*, 87-94, 2008.

[27] H. A. Firip, and E. Goodman, "Swarmed feature selection," *IEEE Conference Proceedings of the 33rd Applied Imagery Pattern Recognition Workshop*, pp. 112-118, Canada, 2004.

[28] H. B. Liu, Y. Tang, J. Meng, and Y. Jp, "Neural networks learning using vbest model particle swarm optimization," *Proc. The 3rd International Conf. Machine Learning and Cybernetics*, pp. 3157-3159, 2004.

[29] B. Al-kazemi, and C. K. Mohan, "Training feed forward neural networks using multi-phase particle swarm optimization," *Proceedings of the 10th International conference on Neural Information*, pp. 2615-2619, Sweden, 2002.

[30] V. G. Gudise, and G. K. Venayagamoorthy, "FPGA placement and routing using particle swarm optimization" *the IEEE Computer Society*, pp. 307-308, Iran, 2005.

[31] M. Nasri, H. Nezamabadi-pour, and M. M. Farsangi, "Design of a PID controller using PSO algorithm Incorporating fuzzy objective function," *Proc. The 6th Iranian Conf. Fuzzy Systems and 1st Islamic Conference on Fuzzy Systems*, pp. 157-168, 2006.

[32] M. M. Farsangi, H. Nezamabadi-pour, and K. Y. Lee, "Multi-objective VAr planning with SVC for a large power system using PSO and GA," *IEEE Power Systems Conference & Exposition*, pp. 274-279, Pakistan, 2006.

[33] S. Mollazei, M. M. Farsangi, and H. Nezamabadi-pour "Allocation of TCSC to enhance total transfer capability using guaranteed convergence particle swarm optimization,"



سیدکمال‌الدین موسوی مشهدی در سال ۱۳۴۸ در مشهد متولد شده‌اند. ایشان مدارک کارشناسی ارشد و دکتری خویش را از دانشگاه فردوسی مشهد به ترتیب در سال‌های ۱۳۷۸ و ۱۳۸۵ اخذ نموده‌اند. ایشان هم‌اکنون در دانشگاه علم و صنعت ایران در تهران با سمت استادیاری در دانشکده مهندسی برق مشغول به فعالیت می‌باشند. علایق مطالعاتی ایشان، کنترل خطی و غیر خطی، کنترل فازی، مدل‌سازی غیرخطی و کاربردهایش، آشوب، کنترل تطبیقی، کنترل فرآیندهای اتفاقی، کنترل چند متغیره و محاسبات نرم می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

sk_mousavi@iust.av.ir



محمد علی پور متولد ۱۳۶۲ و دانش آموخته کارشناسی مهندسی پزشکی و کارشناسی ارشد الکترونیک - پردازش سیگنال به ترتیب از دانشگاه‌های صنعتی سهند تبریز و تربیت معلم سبزوار می‌باشد. او هم‌اکنون در رشته مهندسی پزشکی و در مقطع دکتری در دانشگاه صنعتی چالمرز - سوئد مشغول به تحصیل است پردازش سیگنال و تصویر (عموماً تصاویر پزشکی)، شناسایی الگو و بهینه‌سازی از زمینه‌های تحقیقاتی مورد علاقه وی می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

mohammad.alipoor@ymail.com

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۰/۷/۱۵

تاریخ اصلاح: ۹۱/۸/۲۴

تاریخ قبول شدن: ۹۱/۹/۵

نویسنده مرتبط: محمد فیوضی، دانشکده مهندسی برق و کامپیوتر، دانشگاه

حکیم سبزوار، سبزوار، ایران.

موضوعات تحقیقاتی مطرح در مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل

احمد عبدالله‌زاده بارفروش شیوا وفادار

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

چکیده

مهندسی نرم‌افزار مبتنی بر عامل، به عنوان رویکرد بعدی تولید نرم‌افزار (بعد از شی گرای) مطرح است. پس از گذشت یک دهه از معرفی این حوزه تحقیقاتی، در این مقاله جایگاه مهندسی نرم‌افزار مبتنی بر عامل بررسی و تحلیل شده است. بدین منظور، در این مقاله یک روش سیستماتیک برای تبیین فضای تحقیقاتی موجود معرفی شده و با پیروی از این روش، حوزه‌های تحقیقاتی در زمینه مهندسی نرم‌افزار مبتنی بر عامل شناسایی گردیده است. در این راستا، تحقیقات انجام شده در هر یک از حوزه‌های متدولوژی و فرآیند، مدل‌سازی و ابزارهای پشتیبانی مهندسی نرم‌افزار مبتنی بر عامل، به صورت اجمالی معرفی شده است. همچنین، با بررسی سیر تکاملی تحقیقات انجام شده در زمینه‌های مختلف، گرایش‌های موجود در این حوزه‌ها مشخص و تحلیل و بر این مبنای موضوعات مطرح برای تحقیقات آینده در این حوزه معرفی شده است.

کلمات کلیدی: مهندسی نرم‌افزار، عامل، مهندسی نرم‌افزار مبتنی بر عامل.

۱- مقدمه

مصنوعی به نرم‌افزار و افزایش توانایی حل مسایل پیچیده، مهندسی نرم‌افزار مبتنی بر عامل رویکرد بعدی مهندسی نرم‌افزار پس از شی‌گرایی معرفی شده است [۲]. پس از معرفی دیدگاه عامل در اواسط دهه ۵۰ میلادی [۳] و کاربردی شدن این دیدگاه با معرفی هوش مصنوعی توزیع شده، در اواسط دهه ۷۰ میلادی [۴]، استفاده از عامل‌ها در ساخت سیستم‌های نرم‌افزاری گسترش یافت. بدین ترتیب در طی دو دهه، تجربیات متعددی در ساخت سیستم‌های مبتنی بر عامل حاصل گردید [۵]. در سال ۱۹۹۹، نظریه مهندسی نرم‌افزار مبتنی بر عامل مطرح گردید. براساس این نظریه، به علت فراگیر شدن دیدگاه مبتنی بر عامل و وجود تجربیات موفق برای تولید سیستم‌های مبتنی بر عامل، امکان ورود به مرحله سوم از سیر تکاملی دیدگاه مبتنی بر عامل احساس شد. بدین ترتیب حوزه تحقیقاتی مهندسی نرم‌افزار مبتنی بر عامل معرفی گردید [۱].

بنابر تعریف IEEE مهندسی نرم‌افزار به کارگیری یک رویکرد سیستماتیک، باقاعده و کمی برای تولید، عملیاتی نمودن و نگهداری نرم‌افزار است. روشی که منجر به تولید نرم‌افزار به صورت مهندسی می‌گردد [۶]. مهندسی نرم‌افزار مبتنی بر عامل به جنبه مهندسی سیستم‌هایی می‌پردازد که از تکنولوژی عامل در ساخت آنها استفاده می‌شود و سعی در ارائه روش‌ها و ابزارهایی دارد که به صورت خاص

با بررسی سیر تکاملی دیدگاه‌های تولید نرم‌افزار می‌توان سه مرحله اصلی برای تکامل دیدگاه جدید در تولید نرم‌افزار در نظر گرفت [۱]: ۱) معرفی دیدگاه جدید ۲) به کارگیری ایده در تولید سیستم و جمع‌آوری تجربیات ۳) فراگیر شدن دیدگاه و ورود به حوزه مهندسی نرم‌افزار. در مرحله ورود به حوزه مهندسی نرم‌افزار، براساس تجربیات موفق تولید کنندگان نرم‌افزار، متدولوژی‌ها و فرآیندهای تولید مشخص می‌شوند و به صورت مدون و سیستماتیک ارائه می‌گردند. روش‌های مختلف تحلیل، طراحی، تست و مدل‌سازی مورد استفاده به صورت مشخص و استاندارد برای دیدگاه مذکور تهیه می‌گردند و ابزارهای مختلف برای پشتیبانی فرآیند تولید نرم‌افزار براساس دیدگاه جدید توسعه می‌یابند.

با توجه به قابلیت‌های دیدگاه مبتنی بر عامل از جنبه مهندسی نرم‌افزار، از قبیل نزدیک بودن به دیدگاه انسان برای شکستن مساله، افزایش سطح تجرید و بسته‌بندی بلوکهای پایه، افزایش امکان استفاده مجدد، افزودن یافته‌های هوش

معیار اصلی انتخاب مجلات، میزان ارتباط آنها با موضوع عامل و سیستم‌های مبتنی بر عامل و اعتبار و عمومیت آنها در جامعه محققان سیستم‌های مبتنی بر عامل بوده است.

کنفرانس‌های مرتبط با موضوع یکی دیگر از مراجع مورد استفاده در این تحقیق بوده است. مهمترین معیارهای انتخاب کنفرانس‌های مربوطه سرفصل مطالب مورد توجه در کنفرانس، نرخ پذیرش مقالات و تعداد مقالات دریافت شده توسط کنفرانس‌ها بوده است. با توجه به اینکه موضوع اصلی این مقاله، مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل بوده است، برخی کارگاه‌های خاص مرتبط با موضوع نیز در نظر گرفته شده اند که در آنها به صورت خاص به موضوع مهندسی نرم‌افزار در این سیستم‌ها پرداخته شده است. جهت بررسی تحقیقات داخلی انجام شده در ایران در زمینه عامل و سیستم‌های مبتنی بر عامل، کنفرانس انجمن کامپیوتر ایران به عنوان معتبرترین و عمومی‌ترین کنفرانس داخلی در زمینه مباحث مرتبط با علوم کامپیوتر نیز در این بررسی در نظر گرفته شده است. لیست کامل مجلات و کنفرانس‌های مرجع در [۱۷] ارائه شده است.

برای جمع‌آوری مقالات منبع، مجموعه مقالات منتشر شده در مراجع و کنفرانس‌های تعیین شده در سال‌های اخیر، گردآوری شده است. در مجموع ۴۶ مقاله مجله و ۴۰۶ مقاله کنفرانس براساس منابع فوق جمع‌آوری شده و به عنوان مقالات منبع مورد استفاده قرار گرفته است.

پس از جمع‌آوری مجموعه مقالات منبع، گام بعدی دسته‌بندی منابع و تعیین زمینه‌های تحقیقاتی بوده است. در این دسته‌بندی، هدف مشخص کردن موضوعات تحقیقاتی مختلفی است که در سیستم‌های مبتنی بر عامل وجود دارد.

گام بعدی فرایند تحقیق، مطالعه منابع و گزارش تحقیقات انجام شده می‌باشد. نتایج این مرحله در بخش‌های ۴ تا ۶ مقاله به صورت مفصل در زمینه متدولوژی و فرایندهای تولید نرم‌افزار، روش‌های مدل‌سازی و ابزارهای اتوماتیک برای پشتیبانی فرایند تولید نرم‌افزار تشریح شده است. حاصل این مرحله گزارش آخرین تحقیقات انجام شده در زمینه هر یک از موضوعات می‌باشد.

تحلیل فضای تحقیق و دسته‌بندی تحقیقات انجام شده در هر یک از زمینه‌های مهندسی نرم‌افزار مبتنی بر عامل، آخرین مرحله در روش تحقیق معرفی شده است. معیارهای تحلیلی انتخاب شده گرایش‌های موجود در هر یک از زمینه‌های تحقیقاتی در سال‌های مختلف و نوع تحقیقات انجام شده در سال‌های اخیر و سال‌های اولیه می‌باشد. همچنین با استفاده از این تحلیل‌ها، کمبودهای موجود مشخص و زمینه‌های موجود برای تحقیقات آینده مشخص شده است. نتایج این تحلیل‌ها در پایان بخش‌های ۴ تا ۶ ارائه شده است.

۳- تحلیل آماری جایگاه مهندسی نرم‌افزار مبتنی بر عامل

براساس روش تحقیقی که در بخش ۲ تشریح گردید، ۴۶ مقاله مجله و ۴۰۶ مقاله کنفرانس جمع‌آوری گردیده است. از میان این مقالات ۲۳۸ مقاله به موضوع عامل و ویژگی‌های عامل پرداخته‌اند که در آنها مباحثی مانند مفاهیم شناختی، یادگیری، جستجو، استنتاج، برنامه‌ریزی، محیط عامل، همکاری و هماهنگی میان عامل‌ها، نحوه ارتباط در سیستم‌های مبتنی بر عامل، ویژگی‌های اجتماعی عامل، ساختارهای اجتماعی سیستم‌های مبتنی بر عامل، مذاکره میان عامل‌ها و واژگان شناختی پرداخته شده است. تحقیقاتی که در آنها نتایج پیاده‌سازی سیستم با استفاده از تکنولوژی عامل گزارش شده است، مشتمل بر ۹۶ مقاله می‌باشند. در مجموع ۱۱۹ مقاله از مقالات جمع‌آوری به مبحث مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل پرداخته‌اند. لیست کامل این مقالات در [۱۷] ارائه شده است. شکل

در ساخت سیستم‌های مبتنی بر عامل مورد نیاز است و از ویژگی‌های عامل و جامعه عامل‌ها ناشی می‌شود.

با گذشت بیش از یک دهه از معرفی مهندسی نرم‌افزار مبتنی بر عامل، تبیین و تحلیل جایگاه آن به عنوان رویکرد بعدی تولید نرم‌افزار، انگیزه این تحقیق بوده است. بدین منظور مجموعه‌ای از تحقیقات انجام شده در این زمینه، در بین سال‌های ۱۹۹۹-۲۰۰۸ گردآوری و گزارش شده است. علت انتخاب این دوره، شکوفایی رویکرد مبتنی بر عامل در طی دهه اول ظهور آن بوده است. بدیهی است در سال‌های بعد نیز تحقیقات مختلفی در این حوزه صورت گرفته است که از نمونه‌های آن می‌توان از [۷]، [۸]، [۹] در زمینه متدولوژی و فرآیند، [۱۰]، [۱۱] در زمینه روش‌های مدل‌سازی و [۱۲]، [۱۳] در زمینه ابزارها نام برد. در این مقاله، تحلیلی آماری بر روی تحقیقات انجام شده در این زمینه صورت گرفته است. همچنین گرایش‌های موجود برای تحقیق در این زمینه مشخص و نقاط ضعف، قوت و کاستی‌های تحقیقات در این حوزه معرفی می‌گردد.

برای دستیابی به اهداف فوق و گزارش فرایند انجام تحقیق، ساختار این مقاله به صورت زیر تنظیم شده است. در بخش ۲، روش مورد استفاده در انجام تحقیق و نحوه گردآوری مجموعه منابع مورد استفاده، معرفی می‌شود و فضای تحقیق در این حوزه ترسیم می‌گردد. تحلیل آماری تحقیقات انجام شده در بخش ۳ ارائه شده است. با دسته‌بندی تحقیقات انجام شده، هفت زمینه تحقیقاتی اصلی در مهندسی نرم‌افزار مبتنی بر عامل تشخیص داده شده است؛ متدولوژی‌ها و فرایندهای تولید نرم‌افزار، روش‌های مدل‌سازی، مباحث مرتبط با نیازمندی‌ها، روش‌های تحلیل و طراحی، پیاده‌سازی، تست و ابزار. در این مقاله تحقیقات مرتبط با متدولوژی و فرآیند (بخش ۴)، مدل‌سازی (بخش ۵) و ابزارها (بخش ۶) مورد تحلیل و بررسی قرار گرفته است. در هر یک از این بخش‌ها، تحقیقات انجام شده در هر یک از این مباحث، گرایش‌های موجود و سیر تکاملی تحقیق براساس دسته‌بندی تحقیقات مورد بررسی صورت گرفته است. در بخش ۷ نیز، تحلیل فضای تحقیق و تعیین کاستی‌های موجود انجام شده است.

۲- روش تحقیق

در این قسمت، روش تحقیق استفاده شده برای تبیین فضای تحقیق در زمینه مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل ارائه می‌گردد. فرآیند تحقیق را می‌توان در پنج مرحله اصلی خلاصه نمود:

۱- تعیین فضای تحقیق

۲- جمع‌آوری مقالات منبع

۳- دسته‌بندی منابع و تعیین زمینه‌های تحقیقاتی

۴- مطالعه منابع و گزارش تحقیقات انجام شده

۵- تحلیل مباحث مطرح در هر یک از زمینه‌های تحقیقاتی

برای تعیین فضای تحقیق، تحقیقات انجام شده به دو نیمه - قبل از سال ۲۰۰۵ و پس از آن - تقسیم شده است. برای بررسی تحقیقات انجام شده در قبل از سال ۲۰۰۵ از تعدادی مقاله مروری استفاده شده است [۲، ۱۴، ۱۵، ۱۶] این مقالات مجموعه‌ای از تحقیقات در زمینه مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل را در سال‌های مختلف گزارش نموده‌اند. مجموعه مقالات مروری، به عنوان منبع اصلی گزارش تحقیقات انجام شده قبل از سال ۲۰۰۵ در نظر گرفته شده‌اند. برای تکمیل مجموعه مقالات منبع لازم است سایر تحقیقات انجام شده در این زمینه، در دوره زمانی مورد بررسی نیز در نظر گرفته شوند. برای گزارش تحقیقات جدیدی که در زمینه مهندسی نرم‌افزار مبتنی بر عامل پس از سال ۲۰۰۵ انجام شده است، تعدادی از مجلات و کنفرانس‌های معتبر مرتبط با عامل و سیستم‌های مبتنی بر عامل (و مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل) انتخاب شده‌اند.

سال‌های اولیه ارائه شد. حجم بالایی از Surveyهای موجود در مهندسی نرم‌افزار که در سال‌های اولیه ارائه شده است، به متدولوژی‌ها اختصاص یافته است [۱۴، ۱۵]. اما در سال‌های اخیر، تمرکز و توجه از تولید متدولوژی به بهبود و اصلاح آن سوق یافته است. مجموعه مقالاتی که در گروه متدولوژی و فرآیندهای مهندسی نرم‌افزار مبتنی بر عامل دسته‌بندی شده‌اند را می‌توان به دسته‌های زیر تقسیم کرد:

- بهبود و توسعه متدولوژی‌های موجود
- ارائه متدولوژی‌های خاص منظوره
- ارزیابی متدولوژی‌های موجود
- تلفیق و استفاده مجدد از اجزای متدولوژی

در ادامه تحقیقات انجام شده در زمینه متدولوژی‌های مهندسی نرم‌افزار مبتنی بر عامل به صورت خلاصه گزارش می‌گردند.

۴-۱- متدولوژی Gaia و توسعه‌های آن

یکی از نخستین متدولوژی‌های ارائه شده، Gaia نام دارد که در آن مراحل تحلیل و طراحی پشتیبانی شده است. تحلیل در متدولوژی Gaia شامل مراحل مدل‌سازی نقش‌ها و مدل‌سازی تعامل است. در مرحله طراحی مدل عامل‌ها، مدل سرویس و مدل شناخت سیستم تولید می‌گردد [۱۸]. در این متدولوژی عامل‌ها در دو سطح مختلف، سطح اجتماع عامل‌ها و ساختار سازمانی سیستم و سطح ساختار عامل مورد بررسی قرار می‌گیرند. با توجه به اینکه متدولوژی Gaia یکی از نخستین متدولوژی‌های ارائه شده برای سیستم‌های مبتنی بر عامل بوده است، در نسخه اصلی این متدولوژی کمبودهایی وجود داشته و تحقیقات مختلفی برای بهبود آن، صورت گرفته است.

یکی از بهبودهایی که برای متدولوژی Gaia ارائه شده است، توانایی مدل‌کردن سیستم‌های قابل پیاده‌سازی در اینترنت را به این متدولوژی اضافه نموده است [۱۹]. در این تحقیق به ویژگی‌هایی از قبیل بازبودن و تضاد اهداف عامل‌های شرکت کننده در سیستم توجه شده است. بدین منظور قابلیت مدل‌سازی روابط اجتماعی عامل به متدولوژی اضافه شده است. در این بهبود، براساس توصیف نیازمندی‌های سیستم، مدل تعامل و قوانین اجتماعی به مرحله تحلیل متدولوژی اضافه شده است. همچنین در مرحله طراحی، مدل رفتار رسانه هماهنگ‌کننده به متدولوژی اضافه شده است. علاوه بر اینکه مدل قوانین اجتماعی و مدل تعامل که در مرحله تحلیل اضافه شده‌اند، بر سایر مدل‌های طراحی مانند مدل عامل و مدل سرویس نیز تاثیرگذار است. بدین ترتیب رفتار سیستم در قبال تعامل‌هایی که در آنها شرکت می‌کند، مشخص می‌شود.

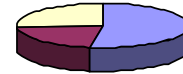
با استفاده از بهبود دیگری که بر روی متدولوژی Gaia صورت گرفته است، متدولوژی جدیدی با عنوان ROADMAP ارائه شده است. در این متدولوژی جدید توانایی مدل‌کردن سیستم‌های باز به متدولوژی Gaia اضافه شده است [۲۰]. در این تحقیق، چهار بهبود در متدولوژی Gaia ارائه شده است. مدل‌های رسمی برای دانش و محیط سیستم، ساختار سلسله مراتبی نقش‌ها، ارتباط صریح ساختارهای اجتماعی و توانایی مدیریت تغییرات پویا.

بهبود دیگری که بر روی متدولوژی Gaia صورت گرفته است [۲۱]، با استفاده از ترکیب آن با روش مدل‌سازی AUML می‌باشد. در این بهبود، مرحله طراحی در متدولوژی Gaia در نظر گرفته شده است و در چهار مرحله پروتکل، تعامل، عامل و سازمان مدل‌های توسعه یافته UML برای عامل‌ها بهبود یافته است.

در [۲۲] دو بهبود دیگر برای متدولوژی پیشنهاد شده است. نخست آنکه مرحله طراحی عامل به متدولوژی اضافه شده است. براساس این بهبود، مرحله طراحی عامل که در متدولوژی اولیه به صورت یک جعبه سیاه در نظر گرفته شده

۱، سهم هر یک از موضوعات فوق را در فضای تحقیق در سیستم‌های مبتنی بر عامل نشان می‌دهد.

مهندسی نرم‌افزار مبتنی بر عامل ۲۶٪

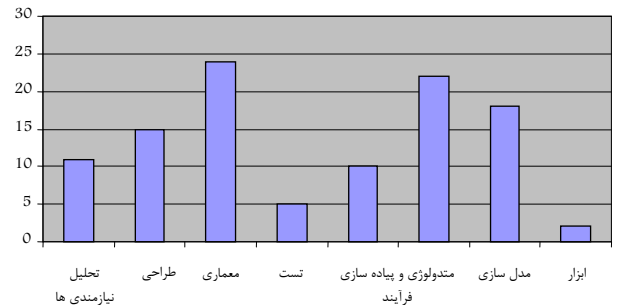


ویژگی‌های عامل و سیستم های چند عامله ۵۳٪

نرم افزارهای کاربردی مبتنی بر عامل ۲۱٪

شکل ۱- سهم موضوعات تحقیقاتی در هوش مصنوعی توزیع شده

با دسته‌بندی مقالات در حوزه مهندسی نرم‌افزار مبتنی بر عامل، سهم هر یک از موضوعات مهندسی نرم‌افزار در این حوزه مشخص می‌شود. نمودار توزیع تحقیقات انجام شده در این زمینه در شکل ۲ نشان داده شده است. با بررسی شکل می‌توان دید که از میان تحقیقات انجام شده در این حوزه، بیشترین سهم به ترتیب مربوط به موضوعات مرتبط با معماری عامل و سیستم چندعامله، متدولوژی و فرآیند و مدل‌سازی است. درحالی که تعداد مقالاتی که در آنها موضوع اصلی ابزارهای یکپارچه مهندسی نرم‌افزار مبتنی بر عامل است، در این بین کمترین سهم را داشته‌اند.



شکل ۲- تحقیقات گزارش شده در موضوعات مختلف مهندسی نرم‌افزار مبتنی بر عامل

در این مقاله، تحقیقات انجام شده در موضوعات مرتبط با متدولوژی و فرآیند، مدل‌سازی و ابزارهای اتوماتیک برای پشتیبانی فرآیندهای مهندسی نرم‌افزار، به عنوان مباحث عمومی مطرح در این حوزه تحقیقاتی مورد بررسی و تحلیل قرار می‌گیرند. در ادامه زمینه‌های تحقیقاتی مورد توجه در هر یک از این موضوعات به تفصیل ارائه می‌گردد.

۴-۲- متدولوژی و فرآیندهای تولید سیستم‌های مبتنی

بر عامل

هدف از ارائه متدولوژی‌ها آن است که طراحان سیستم بتوانند به صورت سیستماتیک نیازمندی‌های کاربران سیستم را به مدل طراحی تبدیل کنند، به طوری که این طراحی دارای جزئیات کاملی از سیستم باشد و در نتیجه به راحتی قابل پیاده‌سازی باشد. پس از معرفی دیدگاه مهندسی نرم‌افزار مبتنی بر عامل، متدولوژی‌های مهندسی نرم‌افزار مبتنی بر عامل یکی از زمینه‌های تحقیقاتی بودند که بسیار مورد توجه قرار گرفتند. به طوری که متدولوژی‌های مختلفی در طی

افزوده شده است. همچنین مرحله ساخت کلاس‌های عامل در متدولوژی به مرحله ساخت «کلاس-عامل-شی» تغییر یافته است [۲۷].

همچنین به منظور افزودن قابلیت توانایی پشتیبانی مفاهیم مرتبط با مدل‌سازی سیستم‌های مبتنی بر عامل که بر ساختار سازمانی بنا می‌شود، مفاهیمی مانند هدف، نقش، عامل، توانایی‌ها و انتساب عامل به نقش‌ها به عنوان الزامات موردنیاز تشخیص داده شده‌اند و متدولوژی برای پشتیبانی این مفاهیم توسعه داده شده است [۲۸].

بهبود دیگری که در متدولوژی MaSE صورت گرفته است، استفاده از روش Method Fragment برای ارائه یک فرآیند تولید نرم‌افزار است [۲۹]. در این بهبود، فرآیند قابل سازمان‌دهی تعریف شده است که با استفاده از آن می‌توان متدولوژی را برای کاربردهای مختلف اختصاصی‌سازی نمود. بدین منظور، Meta Model مورد نیاز فرآیند، مجموعه‌ای از Method Fragment‌ها و مجموعه‌ای از راهنمایی‌ها برای اختصاصی نمودن و استفاده از آنها ارائه شده است. در این فرآیند، سه مرحله مهندسی نیازمندی‌ها، تحلیل و طراحی در نظر گرفته شده است. فعالیت‌های مهندسی نیازمندی‌ها، مدل‌سازی اهداف و اصلاح اهداف هستند که با استفاده از درخت اهداف AND_OR مدل‌سازی می‌شود. در مرحله تحلیل، مدل‌سازی ساختار سازمانی، مدل‌سازی نقش‌ها، مدل‌سازی دامنه و اصلاح نقش‌ها انجام می‌شود. در مرحله طراحی، فعالیت‌های مدل‌سازی کلاس‌ها، مدل‌سازی پروتکل‌ها، مدل‌سازی برنامه‌ها، مدل‌سازی توانایی‌ها، مدل‌سازی فعالیت‌ها و مدل‌سازی سرویس‌ها صورت می‌گیرد.

۴-۳- متدولوژی Tropos و توسعه‌های آن

Tropos یکی از متدولوژی‌های نسبتاً جدید در مهندسی نرم‌افزار مبتنی بر عامل است [۳۰]. تفاوت اصلی این متدولوژی، با سایر متدولوژی‌ها تمرکز آن بر روی تحلیل نیازمندی‌ها است به طوری که در مرحله اول دینفغان سیستم و تمایلات آنها مورد بررسی و تحلیل قرار می‌گیرد. فرآیند تولید نرم‌افزار در این متدولوژی شامل پنج مرحله تحلیل اولیه نیازمندی‌ها، تحلیل ثانویه نیازمندی‌ها، طراحی معماری، طراحی جزئی و پیاده‌سازی است. در طراحی متدولوژی Tropos، از تئوری‌های ساختارهای اجتماعی مانند تئوری سازمانی، طراحی فرآیند و پیوستگی استراتژیک الهام گرفته شده است.

با وجود اینکه متدولوژی Tropos، به عنوان یک متدولوژی نسبتاً جدید در سیستم‌های مبتنی بر عامل ارائه شده است، توسعه‌های متعددی برای این متدولوژی پیشنهاد شده است. در [۳۱]، یک مدل رسمی تحلیل هدف به متدولوژی Tropos اضافه شده است که باعث تقویت این متدولوژی در مرحله تحلیل نیازمندی‌ها گردد. این تحقیق، با استفاده از ابزارها و مدل‌های استدلال رو به جلو و رو به عقب، و با به کارگیری روش تحلیل هدف رسمی، در مرحله تحلیل نیازمندی‌ها، امکان یافتن اهداف متضاد تعریف شده در مرحله تحلیل نیازمندی‌ها و تعیین اهدافی که در صورت برآورده شدن، هدف سیستم به صورت کلی برآورده می‌شود را فراهم نموده است. همچنین با استفاده از این روش، این امکان وجود دارد که با استفاده از استدلال رو به جلو مشخص گردد، برای اینکه هدف سیستم برآورده شود، کدام مجموعه اهداف سیستم باید برآورده شوند. با به کارگیری این روش، امکان وجود گزینه‌های مختلف در مرحله تحلیل سیستم فراهم می‌شود و تحلیلگر سیستم امکان انتخاب میان اهداف مختلف را خواهد داشت. بدین منظور گراف هدف و نمادهای ارتباط میان اهداف برای توصیف هدف‌ها و ارتباط میان آنها به صورت رسمی، ارائه شده است.

همچنین در [۳۲] تحلیل اهداف سیستم‌ها و گزینه‌های مختلف برای اختصاص هدف به نقش در سیستم مبتنی بر عامل در مرحله تحلیل اهداف سیستم به

است، براساس معماری‌های شناخته شده برای عامل، گسترش یافته است. بدین منظور، مدل‌ها و فعالیت‌هایی در نظر گرفته شده است تا مشخص شود که چگونه عامل، اهدافی که به آن تخصیص یافته است را برآورده می‌نماید. مدل ساختاری (شامل کلاس‌هایی که یک نقش به آنها نیاز دارد) و مدل عملکرد (شامل سناریوهایی که برای انجام فعالیت‌های موردنیاز نقش، مورد استفاده قرار می‌گیرد) به متدولوژی اضافه شده است. این مدل‌ها در فعالیت‌های انتخاب معماری عامل (براساس الگوهای طراحی)، ایجاد نمودار کلاس، ایجاد سناریوها در مرحله طراحی تولید می‌شوند. همچنین برای بهبود فرایند تولید سیستم مبتنی بر عامل با استفاده از Gaia یک رویکرد مبتنی بر تکرار در متدولوژی در نظر گرفته شده است. براساس این پیشنهاد، برای جلوگیری از تولید سیستمی که با نیازهای کاربر منطبق نیست و برای غلبه بر پیچیدگی سیستم، بهتر است سیستم در چندین تکرار تولید شود.

۴-۲- متدولوژی MaSE و توسعه‌های آن

متدولوژی MaSE (Multiagent System Engineering) یک متدولوژی برای تحلیل و طراحی سیستم‌های مبتنی بر عامل می‌باشد و از نقاط قوت آن این است که در آن ابزاری به نام agentTool ارائه شده است که کلیه مراحل تحلیل و طراحی سیستم در متدولوژی MaSE را پشتیبانی می‌نماید [۲۳]. مرحله تحلیل در MaSE، شامل سه مرحله اصلی تعیین اهداف سیستم، اعمال موارد کاربرد و بازبینی نقش‌ها است. مرحله طراحی در MaSE شامل چهار مرحله ایجاد کلاس‌های عامل، ساخت گفتگوها، ترکیب کلاس‌های عامل و طراحی سیستم می‌باشد.

متدولوژی MaSE نیز از زمان ارائه نسخه اولیه تاکنون، به صورت‌های مختلف بهبود و توسعه یافته است. در یکی از بهبودهای متدولوژی MaSE، مرحله مدل‌سازی هستان‌شناسی به مرحله تحلیل متدولوژی اضافه شده است [۲۴]. در این توسعه برای متدولوژی، ابتدا مشخص می‌گردد که هدف و محدوده هستان‌شناسی موردنیاز عامل چیست. سپس داده‌های موجود در دامنه سیستم که بدین منظور مورد استفاده قرار می‌گیرند، جمع‌آوری می‌گردد.

همچنین امکان مدل‌سازی روابط سازمانی میان عامل‌ها نیز در توسعه دیگری به متدولوژی MaSE افزوده شده است [۲۵]. در این توسعه، پس از مرحله مدل‌سازی هستان‌شناسی عامل در مرحله تحلیل، تحلیل و مدل‌سازی ساختار سازمانی انجام می‌شود. در این مرحله، قوانین سازمانی که محدودیت‌های رفتار عامل و ارتباط و تعامل میان عامل‌های یک سازمان است، مشخص می‌شود. بدین ترتیب قوانین سازمانی در قالب تعدادی محدودیت بر روی رفتار نقش (در مرحله تحلیل) ارائه می‌شود. برای این منظور، نمادهایی برای توصیف این محدودیت‌ها در نظر گرفته شده است.

افزودن قابلیت حرکت به عامل‌های سیستم از دیگر بهبودهای متدولوژی MaSE می‌باشد [۲۶]. بدین منظور، در مرحله تحلیل، در مدل‌سازی فعالیت‌ها که در قالب نمودار حالت انجام می‌شود، دستور حرکت به متدولوژی اضافه شده است. در این مرحله، نتیجه حرکت و علت شکست آن (در صورت شکست) مشخص می‌شود تا امکان بازیابی عامل از این شکست، فراهم شود. همچنین در مرحله طراحی، این امکان به عامل‌ها اضافه شده است که با دریافت دستور حرکت، وضعیت داخلی همه اجزای خود را ذخیره نمایند تا پس از انتقال به مقصد امکان شروع مجدد فعالیت‌ها از وضعیتی که در آن قرار داشته‌اند، فراهم شود.

همچنین در توسعه دیگری برای این متدولوژی که منجر به ارائه متدولوژی Ex-MaSE شده است، مدل محیط و مدل دانش به مرحله تحلیل متدولوژی

نگرفته‌اند. بلکه دیدگاه‌های خاصی در آنها وجود داشته است که کل مراحل تولید متدولوژی را تحت تاثیر قرار داده است.

دیدگاه در نظر گرفتن ذینفعان سیستم در فرآیند تولید نرم‌افزار موجب معرفی متدولوژی شده است که رشد عامل نرم‌افزاری مبنای فرآیند توسعه سیستم قرار می‌گیرد. براساس این دیدگاه که ذینفعان سیستم باید در تولید سیستم دخالت داشته باشند، شرایطی که در آن عامل در سیستم عمل می‌کند، شبیه‌سازی می‌شود. بدین ترتیب شرایط مختلفی که عامل در آن قرار می‌گیرد، مورد بررسی قرار می‌گیرد. یکی از این شرایط وضعیتی است که عامل خود قادر به اجرای رفتار خاصی نیست و برای تصمیم‌گیری به کمک نیاز دارد. در این حالت ذینفع سیستم موظف به راهنمایی عامل برای تصمیم‌گیری است [۴۳].

برای افزایش قابلیت تغییر سیستم‌های تولید شده، متدولوژی ارائه شده است که در آن امکان نگاشت میان اهداف سیستم (به عنوان نیازمندی‌ها) و رفتارهای عامل وجود دارد. بدین ترتیب، این امکان وجود دارد که با ارزیابی رفتارهای عامل و میزان موثر بودن آنها، در صورت نیاز به تغییر رفتار عامل، اهداف متناظر با رفتار را نیز تغییر داد و برعکس. بدین ترتیب امکان اصلاح طراحی براساس تغییر نیازهای سیستم، در یک فرآیند تکرار شونده به وجود می‌آید [۴۴].

برای افزایش سهم انسان در تولید سیستم‌های مبتنی بر عامل، متدولوژی ارائه شده است که در آن روش‌های جهان مجازی سه‌بعدی و موسسه‌های الکترونیکی، ترکیب شده است [۴۵].

دیدگاه دیگری که برای تولید سیستم‌های مبتنی بر عامل ارائه شده است، عامل‌ها را براساس نیاز کاربر در زمان اجرا ایجاد می‌نماید. براساس این دیدگاه، لزوماً هر آنچه در زمان طراحی به عنوان عامل در نظر گرفته شده است، در زمان اجرا ایجاد نمی‌شود بلکه با در نظر گرفتن هزینه‌های ایجاد عامل و نیاز کاربر، عامل‌های فعال در سیستم انتخاب می‌شوند [۴۶].

۴-۶- ارزیابی متدولوژی‌های موجود

از آنجا که متدولوژی‌های مختلفی برای سیستم‌های مبتنی بر عامل ارائه شده است، تحقیقات متعددی نیز برای ارزیابی متدولوژی‌های موجود صورت گرفته است. در این تحقیقات سعی شده است که متدولوژی‌های ارائه شده براساس معیارهای مختلف مورد ارزیابی قرار گیرند.

در یکی از تحقیقاتی که در این زمینه انجام شده است، سه معیار اصلی میزان مستندات موجود در ارتباط با متدولوژی، میزان شناخته شده بودن در جامعه محققان و کاربران و خاص منظوره نبودن متدولوژی برای ارزیابی متدولوژی‌ها انتخاب شده است. بنابر ارزیابی انجام شده در این تحقیق، براساس این معیارها متدولوژی‌های Gaia, MaSE, Tropos, Agent SE, MASSIVE, Prometheous, MAS-CommonKADS, PASSI, MESSAGE متدولوژی‌های برتر هستند. در این تحقیق همچنین متدولوژی‌های مذکور براساس معیارهای مرتبط با پشتیبانی ویژگی‌های عامل بودن، مدل‌سازی، ارتباط، فرآیند، برنامه کاربردی و کاربر نیز مورد ارزیابی قرار گرفته‌اند. اگرچه در ارزیابی هر یک از معیارها، متدولوژی‌های مختلف در رده‌های مختلف قرار می‌گیرند، اما براساس نتایج ارزیابی براساس کلیه معیارها، متدولوژی MaSE برترین متدولوژی معرفی شده است [۴۷].

در تحقیق دیگری که در [۴۸] گزارش شده است، متدولوژی‌های مبتنی بر عامل از دیدگاه فرآیندی مورد ارزیابی قرار گرفته‌اند. براساس نتایج این ارزیابی، متدولوژی‌های مبتنی بر عامل بیشتر براساس دیدگاه آشنایی و تکرار شونده و تکاملی شکل گرفته‌اند و دیدگاه‌های دیگری مانند دیدگاه Agile، دیدگاه‌های مبتنی بر ارائه چندین جنبه از سیستم، فرآیندهای تلفیق متدولوژی‌های موجود،

متدولوژی Tropos در نظر گرفته شده است. بدین منظور یک روش تخصیص هدف به نقش ارائه شده است که امکان تولید گزینه‌های مختلف برای نقش‌های سیستم و انتخاب میان آنها را فراهم می‌کند.

در تحقیق دیگری که در ارتباط با متدولوژی Tropos انجام شده است، به نیازمندی‌های تعامل میان عامل در سیستم‌های مبتنی بر عامل و متدولوژی Tropos پرداخته است [۳۳]. بنابر نظر نویسندگان مقاله نیازمندی‌های تعامل عامل در این متدولوژی در مراحل اولیه مورد توجه قرار نمی‌گیرد و پروتکل‌های ارتباطی تا مرحله طراحی جزئی سیستم مشخص نمی‌شوند. از طرف دیگر، در نظریه Commitment Protocol پروتکل‌های تعامل و زمینه اجرای آنها در سیستم کاربردی مشخص نمی‌شود. برای حل این مساله در این مقاله، از ترکیب این دو موضوع استفاده شده است به طوری که Commitment Protocol‌ها به متدولوژی Tropos اضافه شده است. بدین منظور راهنمایی‌هایی برای تعیین تعامل میان بازیگران مختلف سیستم براساس مفاهیم ارائه شده در متدولوژی (مانند وابستگی میان بازیگران) ارائه شده است. براساس این راهنمایی‌ها به صورت سیستماتیک می‌توان مشخص کرد که چه پروتکل‌هایی میان بازیگران مختلف سیستم باید تعریف گردد.

۴-۴- سایر متدولوژی‌ها

علاوه بر متدولوژی‌های فوق که در مهندسی نرم‌افزار مبتنی بر عامل بسیار مورد توجه و استفاده هستند، متدولوژی‌های دیگری نیز در این حوزه معرفی شده‌اند، اما تحقیقات زیادی برای توسعه و بهبود آنها گزارش نشده است. در این بخش به معرفی اجمالی این متدولوژی‌ها می‌پردازیم.

- متدولوژی MASSIVE که در آن هفت دیدگاه محیط، وظیفه، نقش، تعامل، جامعه، معماری و سیستم مدل‌سازی می‌شود [۳۴].
- متدولوژی MESSAGE که در آن مفاهیم نقش، سازمان، منابع در قالب موجودیت‌های سیستم، وظیفه و پروتکل‌های تعامل در قالب فعالیت‌های موجود در سیستم و هدف به عنوان حالت ذهنی عامل تحلیل می‌شوند، [۳۵]. و در پی آن متدولوژی INGENIAS که در آن پنج مدل عامل، تعامل، وظیفه و هدف، سازمان، محیط تولید می‌شود [۳۶].
- متدولوژی MAS-CommonKADS که توسعه متدولوژی‌های مبتنی بر دانش است و در آن فعالیت‌های مدل‌سازی عامل، مدل کردن وظایف، مدل کردن هماهنگی، مدل کردن دانش، مدل کردن ساختار سازمانی در مرحله تحلیل صورت می‌گیرد. طراحی در این متدولوژی شامل طراحی شبکه عامل‌ها، طراحی عامل و طراحی ساختار زیربنایی عامل است [۳۷].
- متدولوژی PASSI که در آن مدل‌های نیازمندی‌های سیستم، جامعه عامله، پیاده‌سازی عامل، کد برنامه و آرایش ساختاری ایجاد می‌شود [۳۸].
- متدولوژی Premetheus که در آن مراحل توصیف سیستم، طراحی معماری و طراحی جزئیات و خطایابی در نظر گرفته شده است [۳۹].
- RICA که برپایه نقش، تعامل و ارتباط بنا شده است [۴۰].
- ODAC [۴۱] که در آن پنج دیدگاه Enterprise، اطلاعات، محاسبات، تکنولوژی و مهندسی [۴۲] مدل می‌شوند.

۴-۵- دیدگاه‌های خاص در تولید متدولوژی

در این قسمت، متدولوژی‌هایی که براساس دیدگاه‌های خاص ایجاد شده‌اند، معرفی می‌شوند. این متدولوژی‌ها براساس دیدگاه عمومی در تولید متدولوژی‌ها شکل

صورت تلفیقی براساس نیازهای سیستم بیشتر مورد توجه قرار گرفته است. بدین ترتیب می‌توان نتیجه گرفت که در زمینه متدولوژی‌های مهندسی نرم‌افزار مبتنی بر عامل تحقیقات از تنوع و تکثر روش‌ها در سال‌ها اولیه به سمت استانداردسازی و تلفیق متدولوژی‌ها در حرکت است. علاوه بر اینکه دیدگاه‌های خاص در تولید متدولوژی که منجر به ارائه شیوه‌های نو یا توجه به مسایل خاص در متدولوژی می‌گردد، هنوز نیز زمینه تحقیقاتی مناسبی است.

همچنین در این بخش، فرآیندهای مهندسی نرم‌افزار مبتنی بر عامل نیاز به تحقیقات گسترده‌ای دارند. چنانکه نتایج این تحقیق نیز نظرات ارائه شده در [۴۸] را تایید می‌نماید، فرآیندهای مهندسی نرم‌افزار در سیستم‌های مبتنی بر عامل چندان مورد توجه قرار نگرفته‌اند (در این تحقیق ۲ مقاله از ۳۸ مقاله که در حدود ۵ درصد از تحقیقات انجام شده است، به موضوع فرآیندهای مهندسی نرم‌افزار پرداخته است).

در میان متدولوژی‌هایی که در این زمینه ارائه شده‌اند، متدولوژی MaSE بیشترین حجم تحقیقات را (۷ مورد از ۳۸ که در حدود ۱۸ درصد کل تحقیقات مورد بررسی در این زمینه در این گزارش می‌باشد) در زمینه توسعه و بهبود به خود اختصاص داده است. این موضوع را می‌توان تأییدی بر ارزیابی انجام شده در [۴۷] دانست که موجب توجه بیشتر محققان به این متدولوژی شده است.

۵- مدل‌سازی در سیستم‌های مبتنی بر عامل

استفاده از زبان مدل‌سازی UML یکی از پرکاربردترین روش‌های مدل‌سازی در سیستم‌های مبتنی بر عامل است. براساس نتایج حاصل از تجربیات اولیه در به کارگیری UML، نیاز به توسعه آن برای مدل‌سازی ویژگی‌های خاص عامل مشخص شد. با استفاده از قابلیت‌های زبان UML برای توسعه، تحقیقات گوناگونی در زمینه توسعه آن برای کاربرد در مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل صورت گرفته است.

بنابر روش تحقیقی که در بخش ۲ تشریح گردید، مجموعه مقالاتی که در ارتباط با بحث مدل‌سازی جمع‌آوری شده‌اند، را می‌توان در موضوعات زیر طبقه‌بندی نمود:

- استفاده از زبان مدل‌سازی UML و ارائه توسعه‌هایی بر آن جهت مدل‌سازی ویژگی‌های عامل و معرفی AUML
 - استفاده از زبان مدل‌سازی *i به عنوان یکی از روش‌های مدل‌سازی عامل‌ها و روابط اجتماعی آنها
 - AML به عنوان یک زبان مدل‌سازی بر مبنای UML
 - ارائه چارچوب‌های خاص برای مدل‌سازی عامل
- در ادامه تحقیقات انجام شده در زمینه مدل‌سازی سیستم‌های مبتنی بر عامل تشریح می‌گردند.

۵-۱- UML و AUML

در تحقیقات اولیه مرتبط با مدل‌سازی سیستم‌های مبتنی بر عامل، از نمودارهای استاندارد UML استفاده می‌شد. در تحقیقی که در [۵۷] گزارش شده است، برای مدل‌سازی عامل به عنوان بالاترین سطح تجرید در سیستم‌های مبتنی بر عامل، چهار نمودار واژگان شناختی، معماری، پروتکل و نقش پیشنهاد شده است. در این مدل‌ها از نمودارهای UML استاندارد (به ترتیب نمودار کلاس، نمودار آرایش ساختاری، نمودار همکاری و نمودار کلاس) برای بازنمایی این ویژگی‌های سیستم استفاده شده است.

افزایش کیفیت و ابزارها از مسایلی هستند که نیاز به تحقیقات بیشتری در آنها وجود دارد.

ارزیابی دیگری که بر روی متدولوژی‌های مبتنی بر عامل صورت گرفته است، میزان پیچیدگی متدولوژی‌های ارائه شده را براساس نمودارها، جداول و الگوهای متنی در چهار سطح، تحلیل سطح صفر، تحلیل سطح یک، طراحی کلی و طراحی جزئی مقایسه نموده است [۴۹].

۴-۷- حرکت به سمت تلفیق متدولوژی‌های موجود

مهندسی متدولوژی (Method Engineering) یک رویکرد جدید برای تولید متدولوژی‌های مهندسی نرم‌افزار است که در آن به جای ایجاد یک متدولوژی که در آن اجزای متدولوژی دارای اجزایی هستند که امکان عملکرد آنها با هم وجود دارد، می‌توان یک متدولوژی را به اجزا یا قسمت‌هایی تجزیه کرد که دارای واسطه‌هایی هستند که امکان ترکیب آنها با هم وجود دارد. بدین ترتیب می‌توان از اجزای متدولوژی برای تولید متدولوژی‌های مختلف استفاده کرد.

تعدادی از تحقیقات انجام شده در زمینه مهندسی نرم‌افزار سیستم‌های مبتنی بر عامل نیز به مهندسی متدولوژی پرداخته است. برای شکستن متدولوژی به اجزای کوچکتر از چارچوب ارائه شده در OPF (Open Process Framework) استفاده شده است که در آن فرامدل، مجموعه‌ای از اجزای موجود و راهنمایی برای نمادهای مورد استفاده و ابزارهای لازم برای شکستن متدولوژی ارائه شده است و سعی شده است که اجزای لازم برای مهندسی متدولوژی در سیستم‌های مبتنی بر عامل مشخص گردند. در این تحقیقات متدولوژی‌های MaSE [۵۰]، Gaia [۵۱]، Premethous [۵۲] و PASSI [۵۳] به تعدادی اجزا شکسته شده است و براساس آن انباره اجزای متدولوژی برای متدولوژی‌های مبتنی بر عامل تهیه شده است. بدین ترتیب یک متدولوژی جامع براساس این دیدگاه ارائه شده است [۵۴].

در تحقیق دیگری که با هدف مهندسی متدولوژی انجام شده است، یک فرامدل به عنوان زیربنایی برای بازنمایی جهت یکسان‌سازی متدولوژی‌های موجود و حرکت به سمت یک توصیف واحد از متدولوژی ارائه شده است. در این تحقیق در یک فرامدل مفاهیم مرتبط با طراحی و زمان اجرا و طراحی داخلی عامل به صورت مدل کلاس بازنمایی شده است [۵۵].

در تحقیق دیگری که در زمینه method Fragment انجام شده است، تحقیقات FIPA برای استانداردسازی مورد بررسی قرار گرفته است و براساس آن، اجزای اصلی متدولوژی‌ها مبتنی بر چندین دیدگاه (شامل فرآیند، استفاده مجدد، ذخیره‌سازی و پیاده‌سازی) مشخص شده است [۵۶].

۴-۸- تحلیل تحقیقات انجام شده در زمینه متدولوژی‌ها

و فرآیندها

در زمینه متدولوژی‌های تولید سیستم‌های مبتنی بر عامل، در این تحقیق ۳۸ مقاله گزارش شده است. نیمی از این تحقیقات مربوط به سال‌های قبل از ۲۰۰۵ و نیمی از آن مربوط به سال‌های پس از ۲۰۰۵ است. با بررسی این مقالات می‌توان نتیجه گرفت که در سال‌های اولیه (۲۰۰۰ تا ۲۰۰۵) حجم بالایی از مقالات به معرفی متدولوژی‌های جدید و توسعه و بهبود آنها اختصاص داشته است. این در حالی است که در سال‌های بعد بحث ارزیابی متدولوژی‌ها بیشتر مورد توجه محققان قرار گرفته است. این در حالی است که مباحث مرتبط با Method Fragment و استفاده مجدد از اجزای متدولوژی و حرکت به سمت استفاده از متدولوژی‌ها به

گرفتن نقش به عنوان توسعه‌ای برای نمودار ترتیب (و همچنین نمودار ارتباط) لازم است.

یکی دیگر از توسعه‌هایی که برای UML ارائه شده است، امکان مدل‌سازی قابلیت حرکت یک عامل است. در این زمینه مسایلی همچون ایجاد عامل جدید، مسیر حرکت و وضعیت فعلی عامل در نظر گرفته شده‌اند و برای مدل‌سازی این ویژگی‌ها چهار مدل مختلف (شامل Sterotype, Swimlaned, State representation, framefragmentation) ارائه شده است. این مدل‌ها را می‌توان توسعه‌هایی برای نمودارهای ترتیب UML در نظر گرفت. بنابر پیشنهاد ارائه شده در این تحقیق، هر یک از این مدل‌ها امکان بازنمایی ویژگی‌های منحصر به فرد یک عامل متحرک در محیط‌های خاص را دارا هستند. بنابراین لازم است که به نوع کاربرد هر یک از نمودارها با توجه به شرایط خاص سیستم توجه شود [۶۵].

مدل‌سازی رفتارهای عامل از قبیل همزمانی، انطباق پذیری و حرکت نیز با استفاده از توسعه نمودارهای پویای UML مورد توجه قرار گرفته است. بدین منظور با توسعه نمودارهای ترتیب و نمودار فعالیت، راهنمایی جهت مدل‌سازی ویژگی‌های مختلف عامل ارائه شده است. به عنوان نمونه پیشنهاد شده است که برای تعامل میان عامل‌ها از نمودار ترتیب برای مدل‌سازی استفاده شود. در حالی که استفاده از نمودار فعالیت در مدل‌سازی هدف‌گرایی، ارسال و دریافت پیام‌ها و انطباق اهداف مناسب‌تر است. در مواردی مانند رفتارهای اجتماعی، انطباق هدف، توزیعی بودن، قابلیت حرکت و اجرای همزمان می‌توان از هر دو نمودار استفاده کرد [۶۶].

۵-۲-۱*

استفاده از مفاهیم و ویژگی‌های عامل برای مدل‌سازی ویژگی‌های موجودیت‌های سیستم (شامل افراد، نرم‌افزارها و سخت‌افزارها)، مبنای بسیاری از روش‌های مبتنی بر عامل است [۶۷]. در مدل i^* که شاخص‌ترین مدل در این روش‌هاست، از مفاهیمی مانند هدف، تمایلات، ارتباطات و توانایی‌های عامل استفاده می‌شود. تا آنچه برای هر یک از ذینفعان سیستم حائز اهمیت است، مشخص شود. بدین ترتیب، هر یک از موجودیت‌های سیستم (از قبیل ذینفعان، موجودیت‌های سخت‌افزاری، نرم‌افزاری، سیستم) در قالب تعدادی عامل دیده می‌شوند که ویژگی‌هایی از قبیل هدف، تمایلات، ساختارهای سازمانی که در آنها شرکت دارند، براساس این دیدگاه مدل می‌شوند و میزان انطباق، سازگاری و تضادهای آنها با یکدیگر مشخص می‌شوند [۶۸]. در چارچوب i^* دو نوع مدل وجود دارد. مدل ارتباطات استراتژیک که بازیگران سیستم، ارتباط میان آنها و وابستگی‌های آنها با یکدیگر را نشان می‌دهد. در مدل‌های عقلانیت استراتژیک، اهداف اصلی بازیگران و شکست آنها به اهداف کوچکتر و نحوه ارتباط و وابستگی‌های این اهداف مدل می‌شود. با استفاده از این دو مدل در کنار هم، یک دیدگاه جامع از سیستم ارائه می‌گردد.

روش i^* مبنای متدولوژی‌های مختلفی قرار گرفته که مهم‌ترین آنها Tropos است. مدل‌سازی و تحلیل سیستم در این متدولوژی‌ها براساس مفاهیم و نمودارهای ارائه شده در i^* انجام می‌شود. در [۶۹]، شش روش که i^* را مبنای مدل‌سازی و تعریف نیازمندی‌های خود قرار داده‌اند انتخاب و براساس معیارهای مختلفی مقایسه شده‌اند. این معیارها عبارتند از: معیارهای فرآیند، میزان توضیحات و راهنماهای ارائه شده برای آنها، منابع معرفی شده مانند محصولات میانی و فرآورده‌های تولید شده، مباحث مطرح در i^* مانند Constructorهای مورد استفاده و میزان گسترش‌پذیری.

همچنین در تحقیق دیگری که در زمینه این چارچوب مدل‌سازی صورت گرفته است، مفهوم و ویژگی ارث‌بری در مدل‌های i^* معرفی شده است. براساس نتایج این تحقیق، سه ایده اصلی توسعه، اصلاح و تعریف مجدد معرفی و نحوه

نخستین تلاش‌های مستمر برای استفاده از زبان مدل‌سازی UML برای بازنمایی ویژگی‌های عامل توسط Odell و همکارانش صورت گرفته است [۵۸، ۵۹]. بنابر این پیشنهادها، توسعه نمودارهای ترتیب در UML جهت پشتیبانی قابلیت‌ها و مفاهیم مطرح در سیستم‌های مبتنی بر عامل در هنگام تعامل مانند نقش، رشته‌های تعاملی میان عامل‌ها، پروتکل‌های تعاملی تودرتو و توسعه مفاهیم پیام‌هایی که امکان تبادل آنها میان عامل‌ها وجود دارد، ارائه شده است [۶۰]. همچنین برای توسعه نمودارهای کلاس UML جهت پشتیبانی از ویژگی‌های خاص عامل، یک ساختار جدید برای کلاس‌های عامل پیشنهاد شده است. در این ساختار اقلام اطلاعاتی از قبیل نام عامل، وضعیت ذهنی که در آن ویژگی‌هایی از قبیل باورها، تمایلات و قصد عامل مشخص می‌شوند، عملیاتی که عامل می‌تواند انجام دهد، سرویس‌ها و توانایی‌هایی که عامل دارد، ساختار سازمانی عامل و رفتارهای عامل مشخص می‌شوند [۶۱].

در این زمینه همچنین برای مدل‌سازی ساختارهای سازمانی موجود میان عامل‌ها، توسعه‌هایی بر UML ارائه شده است. بدین ترتیب امکان بازنمایی ساختار اجتماعی در سیستم‌های مبتنی بر عامل فراهم شده است [۶۲].

بدین ترتیب توسعه‌هایی برای UML معرفی شد که منجر به معرفی Agent UML (AUML) گردید. هدف از این زبان، مدل‌سازی کلیه ویژگی‌های عامل با استفاده از UML است. بنابر این پیشنهادها، امکان توصیف نقش در قالب نمودارهای ترتیب، افزودن عامل به عنوان یک واسطه، توانایی حرکت میان سیستم‌های خودمختار مختلف و تغییر مدل‌های آرایش ساختاری سیستم، از جمله پیشنهادهای این تیم بوده است. مجموعه این پیشنهادها به تیم استانداردسازی UML ارائه شده است تا در UML2 به عنوان استاندارد جدید در نظر گرفته شود [۱۴].

نحوه استفاده و کاربرد نمودارهای مختلف ارائه شده در UML2 برای عامل و سیستم‌های مبتنی بر عامل در [۶۳] ارائه شده است. از میان نمودارهای ساختاری که جنبه‌های ایستای سیستم را مدل‌سازی می‌نمایند، نمودار کلاس یکی از پرکاربردترین نمودارها در مدل‌سازی سیستم‌های مبتنی بر عامل است. از این نمودار می‌توان برای مدل‌سازی عامل، ساختار سلسله مراتبی نقش‌ها، مدل سرویس عامل، ساختار سازمانی، واژگان شناختی، ساختار اجتماعی دانش عامل استفاده کرد. علاوه بر این نمودار شی را می‌توان برای مدل‌سازی عامل‌ها در زمان اجرا مورد استفاده قرار داد. نمودار ساختار ترکیبی در UML2 نشان می‌دهد که چگونه اجزای مختلف معماری با هم ترکیب می‌شوند. بدین ترتیب می‌توان از این نمودار برای نمایش ساختار اجتماعی استفاده کرد. از نمودار اجزای ارائه شده در این استاندارد می‌توان برای تعریف رفتارهای ورودی و خروجی هر وظیفه و برای نمایش شکستن معماری سیستم نیز استفاده نمود. برای تعیین توزیع عامل‌ها به صورت فیزیکی می‌توان از نمودار آرایش ساختاری استفاده کرد. جنبه‌های پویای سیستم با استفاده از نمودارهای رفتاری قابل مدل‌سازی است. با توجه به اصلاحاتی که در تعریف بازیگران سیستم در نمودار مورد کاربرد ارائه شده است، نسخه جدید این نمودار، برای مدل‌سازی سیستم‌های مبتنی بر عامل مناسب‌تر شده است به طوری که می‌توان بازیگران داخلی و خارجی سیستم را مشخص نمود. همچنین برای تعیین برنامه‌هایی که یک عامل از آن استفاده می‌کند، می‌توان از نمودار فعالیت استفاده نمود. ولی برای این کار، مدل‌های UML2 نیاز به توسعه دارند. همچنین برای اینکه امکان انتخاب میان برنامه‌های مختلف وجود داشته باشد، توسعه‌هایی برای این نمودار لازم است. نمودارهای ترتیب و حالت را می‌توان برای مدل‌سازی پروتکل‌های تعامل میان عامل‌ها مورد استفاده قرار داد. توسعه‌های ارائه شده در UML2 برای نمودارهای ترتیب، دربرگیرنده کلیه توسعه‌های پیشنهاد شده توسط FIPA است [۶۴]. اما در آن نقش به عنوان یک مفهوم در نظر گرفته نشده است. بنابراین برای مدل‌سازی سیستم‌های مبتنی بر عامل، در نظر

روش دیگری که از متدولوژی شی-فرآیند ایده گرفته است، امکان مدل‌سازی سیستم در سطوح مختلف تجرید را فراهم می‌کند و این روش در متدولوژی SODA اعمال شده است [۷۶].

در تحقیق دیگری که در آن ارتباط میان عامل‌ها از طریق تعریف نقش صورت می‌گیرد، چارچوبی با عنوان BRAIN (Behavioral Role for Agent Interactions) ارائه شده است. در این مدل لایه‌ای، مدل نقش، مدل‌سازی تعامل میان عامل‌های مختلف در سطح قابلیت‌های اصلی عامل، نمادهایی برای مدل‌سازی نقش براساس XML و زیرساخت تعامل ارائه شده است [۷۷].

مدل‌سازی دانش موردنیاز عامل در زمان اجرا موضوع تحقیقی است که در [۷۸] گزارش شده است. در این تحقیق، یک مدل سلسله‌مراتبی از دانش که لایه‌های پایین آن در داخل سیستم و لایه‌های بالاتر در اختیار عامل‌ها قرار می‌گیرد، ارائه شده است. بدین منظور دانش کسب و کار به سه دسته دانش فرآیند، دانش قوانین و دانش مفاهیم تقسیم شده است. بر این اساس، دو مدل مفهومی برای تعریف کلمات عامل و مدل واقعیت‌ها با استفاده از XML بازنمایی شده است.

برای بهبود مدل‌سازی سیستم‌های مبتنی بر عامل، چارچوب مدل‌سازی دیگری با عنوان Conceptual Graph Model (CG Model) پیشنهاد شده است. تمرکز این چارچوب بیشتر بر روی مرحله بدست آوردن نیازمندی‌های اولیه با استفاده از به کارگیری مفاهیم مدل ارائه شده می‌باشد [۷۹]. در فرامدل ارائه شده در این تحقیق، مدل مورد کاربرد، مدل تراکنش، مدل‌سازی سناریوهای توصیف سیستم، صحت‌سنجی و ارزیابی مدل موجود و انتقال به مدل طراحی مطرح شده است.

در تحقیق دیگری که برای مدل‌سازی عامل‌ها گزارش شده است، از I/O Automata که یک زبان رسمی برای توصیف سیستم‌های توزیع شده و واکنشی است، برای توصیف عامل‌ها استفاده شده است. بدین منظور یک معماری شامل حسگر، کنترل‌کننده و اثرگذار برای عامل در نظر گرفته شده است. همچنین قابلیت حرکت به عامل اضافه شده و بخش‌های مختلف معماری با استفاده از این زبان مدل‌سازی شده است [۸۰].

چارچوب Opera یک چارچوب برای مدل‌سازی ساختارهای سازمانی برای سیستم‌های چندعامله است که دیدگاه‌های مختلف از جامعه عامل‌ها ارائه می‌نماید. در این چارچوب، مدل سازمانی، مدل اجتماعی و مدل تعامل برای عامل‌ها در نظر گرفته شده است که براساس آن تعاملات، روابط اجتماعی، قوانین و قابلیت‌های مذاکره مورد نیاز عامل‌هایی که به یک سازمان ملحق می‌شوند، مورد بررسی قرار می‌گیرد [۸۱]. براساس این چارچوب، متدولوژی ارائه شده است که سناریوهای موردنیاز یک سازمان برای مدل‌سازی شرایط سازمانی (از قبیل اهداف و استراتژی سازمان و اهداف و نیازهای کاربران) را پوشش دهد [۸۲].

۵-۵- سایر تحقیقات مرتبط با مدل‌سازی

یکی از تحقیقاتی که در زمینه مدل‌سازی سیستم‌های مبتنی بر عامل، صورت گرفته است، به ترکیب روش‌های رسمی و غیررسمی توجه نموده است [۸۳]. در این تحقیق، روش غیررسمی i^* با روش رسمی Cognitive Agent Specification Language (CASL) ترکیب شده است تا امکان توصیف سیستم‌های مبتنی بر عامل فراهم شود. برای این کار مجموعه‌ای از نمادهای میانی در قالب نمودارهای Intentional Annotation Strategic Rational (iASR) پیشنهاد شده است که فاصله میان این دو روش توصیف رسمی و غیررسمی را بپوشاند. هدف از انجام این کار ارائه افزایش دقت توصیف نیازمندی‌ها و افزودن

اثرگذاری هر یک از آنها بر اجزای اصلی سیستم و مدل‌سازی مشخص شده است [۷۰].

۵-۳- AML

زبان مدل‌سازی عامل (AML) یک زبان گرافیکی برای توصیف، مدل‌سازی و مستندسازی سیستم‌هایی است که در آنها مفاهیم موردنیاز براساس تئوری سیستم‌های چندعامله در نظر گرفته شده است. این زبان توسعه‌ای بر UML2 منطبق بر چارچوب مدل‌سازی OMG است. AML مفاهیم موردنیاز برای مدل‌سازی ویژگی‌های مختلف سیستم‌های چندعامله مانند واژگان شناختی، موجودیت‌های سیستم چندعامله، ویژگی‌های اجتماعی، تجرید رفتار و تعامل میان عامل‌ها، جنبه‌ها و مفاهیم ذهنی برای مدل‌سازی ویژگی‌های ذهنی عامل را ارائه می‌نماید [۷۱].

برای مدل‌سازی ویژگی‌های اجتماعی سیستم‌های مبتنی بر عامل، مدل‌سازی ساختار اجتماعی با معرفی موجودیت‌های اجتماعی، نقش‌های موجودیت‌ها، روابط اجتماعی، ارتباطات و ویژگی‌های نقش صورت می‌گیرد. علاوه بر این رفتار اجتماعی با در نظر گرفتن جنبه‌های پویای اجتماعی بودن، تعاملات اجتماعی و فعالیت‌های اجتماعی مدل می‌شوند. بدین منظور توسعه‌هایی برای UML ارائه شده است [۷۲].

۵-۴- سایر روش‌ها

علاوه بر زبان‌های مدل‌سازی که در بخش‌های قبل عنوان گردید، روش‌های دیگری نیز در مدل‌سازی سیستم‌های مبتنی بر عامل وجود دارد. این روش‌ها با استفاده از ایده‌های موجود در شی‌گرایی یا با استفاده از زبان‌های خاص به ارائه چارچوب‌هایی پرداخته‌اند که براساس آنها امکان مدل‌سازی عامل و ویژگی‌های مبتنی بر عامل وجود دارد. نکته قابل توجه درباره کلیه این روش‌ها آن است که معمولاً کاربردهای خاص دارند و به اندازه زبان‌هایی که در بخش‌های قبل معرفی شده‌اند، عمومیت ندارند.

از آنجایی که ممکن است موجودیت‌هایی در سطوح مختلف تجرید در یک سیستم مورد استفاده قرار گیرند، یکی از مسایل موجود نحوه تعامل میان این موجودیت‌هاست. برای مدل‌سازی نحوه ارتباط عامل و شی در یک سیستم اطلاعاتی، روش مدل‌سازی AOR پیشنهاد شده است. هدف از این مدل‌سازی تسهیل انتقال اطلاعات میان موجودیت‌های مختلفی است که در طراحی سیستم‌های اطلاعاتی به کار گرفته می‌شوند. در این روش، عامل‌ها از دو دیدگاه داخلی و خارجی مدل می‌شوند. دیدگاه خارجی، دیدگاه یک ناظر خارجی است که در آن تعاملات سیستم میان موجودیت‌های مختلف سیستم مدل می‌شوند. در حالی که دیدگاه داخلی، دیدگاه یک عامل است که در آن محیط براساس وضعیت ذهنی عامل مدل می‌شود. برای هر یک از این دیدگاه‌ها مدل‌های قالب، ترتیب و الگو ارائه شده است [۷۳، ۷۴].

یکی دیگر از روش‌هایی که از روش‌های مدل‌سازی موجود در شی‌گرایی ایده گرفته است، OPM/MAS نامیده می‌شود. این روش توسعه‌ای بر متدولوژی شی-فرآیند است که برای سیستم‌های مبتنی بر عامل براساس مفاهیم معرفی شده در متدولوژی Gaia به روزرسانی شده است. ایده اصلی در این روش آن است که مدل‌سازی سیستم می‌تواند در یک نمودار واحد با در نظر گرفتن موجودیت‌ها و فرآیندهای موجود در سیستم صورت گیرد. موجودیت‌ها شامل سازمان، اجتماع، Platform، قانون، نقش، کاربر، پروتکل، باور، تمایل، واقعیت، هدف، قصد، و سرویس است. درحالی که فرآیند شامل عامل، وظیفه و پیام‌رسانی است [۷۵].

۲) ابزارهایی که به ابزارهای مهندسی نرم‌افزار معروفند و پشتیبانی تمام خودکار یا نیمه خودکار را برای فرآیند و متدولوژی‌های تولید نرم‌افزار مهیا می‌سازند. برای سیستم‌های مبتنی بر عامل، ابزارهای مهندسی نرم‌افزار یکی از زمینه‌هایی است که کمترین نتایج تحقیقات در آن گزارش شده است.

تحقیقات مرتبط با ابزارهای مهندسی نرم‌افزار مبتنی بر عامل معمولاً به صورت جانبی در کنار مفاهیم ارائه شده گزارش می‌شوند و تحقیقات مستقل در زمینه تولید ابزارهای یکپارچه‌ای که فرآیندها، متدولوژی‌ها و روش‌های تولید نرم‌افزار را پشتیبانی نماید، بسیار اندک است.

در این بخش در مجموع دو مقاله به صورت مستقل تحقیقات مرتبط با تولید ابزارهای مهندسی نرم‌افزار را گزارش نموده‌اند. در [۸۶] به تبدیل اتوماتیک نمودار برنامه‌ریزی متدولوژی Tropos به نمودارهای فعالیت در UML2 پرداخته شده است. در این تحقیق، نحوه استفاده از تکنیک‌های تبدیل برای نگاشت میان مدل‌ها تشریح شده است و ابزاری براساس این دیدگاه معرفی شده است.

تحقیق دیگری به موضوع افزایش قابلیت استفاده از سیستم‌های مبتنی بر عامل برای افراد خبره در دامنه مساله، پرداخته است. در این تحقیق امکان تغییر و اصلاح کاربران در سیستم مبتنی بر عامل با استفاده از یک چارچوب مبتنی بر اجزای نرم‌افزاری ارائه و ابزاری برای پشتیبانی آن معرفی شده است و کارایی ابزار در برآوردن نیازهای کاربران ارزیابی شده است [۸۷].

از میان متدولوژی‌های مهندسی نرم‌افزار معرفی شده در بخش ۴، متدولوژی MaSE با ابزار agentTool یکی از نخستین متدولوژی‌هایی بوده است که توسط ابزارهای یکپارچه در کلیه مراحل تحلیل و طراحی پشتیبانی می‌شوند. متدولوژی Tropos با ابزار TAOM4E پشتیبانی شده است. برای تحلیل هدف در این مرحله ابزار GR-Tool و برای ملاحظات امنیتی ابزار ST-Tool ارائه شده است.

متدولوژی Prometheus از دیگر متدولوژی‌هایی است که با ارائه ابزار PDT مراحل طراحی را پشتیبانی نموده است. متدولوژی PASSI متدولوژی دیگری است که با ارائه یک ابزار با عنوان PTK (Passi Toolkit) امکان پشتیبانی مراحل مختلف در آن فراهم شده است.

در تحقیقات مرتبط با نیازمندی‌های سیستم‌های مبتنی بر عامل، ابزارهای اتوماتیک برای نیازمندی‌ها در متدولوژی Tropos با عنوان OME ارائه شده است. علاوه بر اینکه GR-Tool نیز برای پشتیبانی تحلیل هدف در این مرحله می‌تواند مورد استفاده قرار گیرد. علاوه بر این از میان تحقیقات مرتبط با تحلیل نیازمندی‌ها، قابلیت استدلال عامل با ارائه یک ابزار که امکان بررسی ویژگی‌های پویای سیستم در برابر Trace های تعریف شده را به صورت اتوماتیک فراهم می‌کند، پشتیبانی شده است [۸۸]. همچنین مفاهیم مرتبط با تحمل‌پذیری خطا به عنوان یکی از ویژگی‌های کیفی سیستم مبتنی بر عامل با استفاده از ابزار PLFaultCAT فراهم شده است [۸۹].

در تحقیقات مرتبط با تست نرم‌افزار، ابزار تست اتوماتیک برای فراهم نمودن امکان تست در متدولوژی Tropos گزارش شده است. این ابزار که eCAT نامیده شده است، مخفف Continouse Agent Testing on Eclipse می‌باشد [۹۰]. همچنین توسعه ابزار JADE برای پشتیبانی تولید عامل‌های ساختگی که امکان تست واحد را فراهم می‌نمایند، صورت گرفته است [۹۱].

در تحقیقات دیگری که در [۹۲] گزارش شده، برای تست واحد با استفاده از چارچوب junit چارچوب جدیدی با عنوان Sunit ارائه شده است. همچنین برای فراهم نمودن امکان تست در متدولوژی PASSI چارچوب و ابزاری برای اجرای آن ارائه شده است [۹۳]. بدین ترتیب می‌توان نتیجه گرفت که در تحقیقات مرتبط با تست سیستم‌های مبتنی بر عامل، پشتیبانی با استفاده از ابزار بسیار حائز اهمیت و مورد توجه محققان می‌باشد.

امکان بررسی و استنتاج اتوماتیک درباره اهداف تعیین شده برای عامل و دانش تخصیص یافته به آن (به عنوان حالت‌های ذهنی عامل) بوده است.

یکی دیگر از تحقیقاتی که در زمینه مدل‌سازی سیستم‌های مبتنی بر عامل صورت گرفته است، بررسی ویژگی‌های عامل مانند یادگیری، قابلیت حرکت، امنیت با دیدگاه جنبه‌گرایی است [۸۴]. چون این ویژگی‌ها بر قابلیت‌های یک عامل از قبیل عملیات و اهداف آن بسیار تاثیرگذار است، در این تحقیق با معرفی مدل عامل، مدل جنبه و مدل ترکیب، یک چارچوب فرا-مدل برای مدل‌سازی جنبه‌های مختلف عامل ارائه نموده است.

در مدل عامل ویژگی‌های عامل از قبیل هدف و اعمال عامل مدل می‌شوند. مدل جنبه، مفهومی است که مفاهیم، ارتباطها و ویژگی‌های زبان‌های مدل‌سازی جنبه‌گرا را داراست. در مدل ترکیب، آنچه از جنبه‌های سیستم که ممکن است بر عامل، هدف یا اعمال اثرگذار باشد، مدل می‌شوند.

۵-۶- تحلیل تحقیقات انجام شده در زمینه مدل‌سازی

در زمینه مدل‌سازی سیستم‌های مبتنی بر عامل، می‌توان گفت که UML و توسعه‌هایی که برای آن ارائه شده‌اند، مبنای اصلی زبان‌های مدل‌سازی سیستم‌های مبتنی بر عامل هستند. به طوری که AUML و AML هر دو براساس این زبان بنا شده‌اند. در سایر روش‌ها و چارچوب‌های مدل‌سازی نیز معمولاً از UML و توسعه‌های آن استفاده شده است تا امکان توصیف سیستم براساس دیدگاه‌های موردنظر فراهم شود. تنها چارچوب مدل‌سازی i* دارای ساختاری منحصر به فرد است که براساس مهندسی نیازمندی‌ها با استفاده از دیدگاه عامل بنا شده است و نمودارهای وابستگی استراتژیک و عقلانیت استراتژیک را ارائه نموده است.

با بررسی سیرتکاملی مدل‌سازی نیز می‌توان دید که تقریباً توسعه‌های UML و استفاده از i* برای مدل‌سازی در سال‌های اولیه معرفی مهندسی نرم‌افزار مبتنی بر عامل مورد استفاده قرار گرفته‌اند اما AML زبان جدیدی است که در سال‌های اخیر معرفی شده است.

تحقیقات انجام شده در سال‌های اخیر و سال‌های گذشته دارای درصد نسبتاً یکسانی است. (در حدود ۴۵٪ از مجموعه مقالات جمع‌آوری شده در این زمینه مربوط به سال‌های قبل از ۲۰۰۵ و ۵۵٪ مربوط به سال‌های بعد از ۲۰۰۵) است. اگر چه بیشتر تحقیقات اولیه بر روی توسعه‌های UML متمرکز بوده است و در سال‌های اخیر معرفی زبان‌های مدل‌سازی خاص برای سیستم‌های مبتنی بر عامل بیشتر مورد توجه قرار گرفته است.

علاوه بر اینکه چارچوب‌های جدید با دیدگاه‌های خاص نیز در سال‌های اخیر موضوع تحقیقات بسیاری قرار گرفته است. به نظر می‌رسد با توجه به جدید بودن زبانی مانند AML، این زبان زمینه تحقیقاتی مناسبی برای سال‌های آینده باشد. هر چند که توسعه‌های UML در سال‌های اخیر نیز همچنان مورد توجه هستند و امکان تحقیق بر روی این زبان مدل‌سازی هنوز وجود دارد.

۶- ابزارهای مهندسی نرم‌افزار

ابزارهای تولید نرم‌افزارهای مبتنی بر عامل را می‌توان به دو دسته اصلی تقسیم کرد.

۱) ابزارهایی که برای پیاده‌سازی سیستم‌های مبتنی بر عامل به کار می‌روند. این ابزارها هم به صورت تجاری توسط گروه‌های صنعتی ارائه می‌شوند و هم به صورت نتایج فعالیت‌های تحقیقاتی توسط پژوهشگران معرفی می‌شوند. نمونه‌هایی از این ابزارها در [۸۵] معرفی شده‌اند.

۷- تحلیل فضای تحقیق و نتیجه‌گیری

با بررسی تحقیقات انجام شده در زمینه ابزارهای اتوماتیک و نیمه اتوماتیک مهندسی نرم‌افزار مبتنی بر عامل می‌توان نتیجه گرفت که:

- تحقیقات مستقل درباره ابزارهای تولید نرم‌افزارهای مبتنی بر عامل، سهم بسیار کمی در مقالات گزارش شده در این حوزه تحقیقاتی دارند.
- تقریباً تمامی متدولوژی‌های مهم در زمینه مهندسی نرم‌افزار مبتنی بر عامل، توسط یک ابزار اتوماتیک یا نیمه اتوماتیک پشتیبانی می‌شوند که فعالیت‌های مراحل مختلف پشتیبانی را پوشش می‌دهد.
- یکی از نقاط قوت تحقیقات مرتبط با متدولوژی‌ها، پشتیبانی توسعه‌های ارائه شده برای متدولوژی به کمک ابزار (در قالب توسعه ابزار) می‌باشد.
- تست سیستم‌های مبتنی بر عامل، یکی از حوزه‌های بسیار قابل توجه در پشتیبانی توسط ابزار است. به طوری که تحقیقات مرتبط با تست سیستم‌های مبتنی بر عامل، اغلب همراه با یک ابزار تست ارائه شده است.

مراجع

[1] N. R. Jennings, "On Agent-based Software Engineering," *J. of Artificial Intelligence*, vol. 117, no. 2, pp. 277-296, 2000.

[2] F. Zambonelli, and A. Omicini, "Challenges and Research Directions in Agent-Oriented Software Engineering," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 9, no. 3, pp. 253-283, 2004.

[3] A. Kay, "Computer Software," *J. of Scientific American*, vol. 251, no. 3, pp. 53-59, 1984.

[4] S. Russell, and P. Norving, *Artificial Intelligence: A Modern Approach*, Prentice Hall Series in Artificial Intelligence, Second Edition, 2003.

[5] H. V. D. Parunak, "A Practitioners' Review of Industrial Agent Applications," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 3, no. 4, pp. 389-407, 2000.

[6] R. Pressman, *Software Engineering, a Practitioner's Approach*, McGraw-Hill, Sixth Edition, 2005.

[7] I. Hadar, T. Kuflik, A. Perini, I. Reinhartz-Berger, F. Ricca, and A. Susi, "An empirical study of requirements model understanding: Use Case vs. Tropos models," In *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10)*. ACM, pp. 2324-2329, 2010.

[8] M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam, "ASPECS: an agent-oriented software process for engineering complex systems," *J. of Autonomous Agents and Multiagent Systems Systems*, vol. 20, no. 2, pp. 260-304, 2009.

[9] S. A. DeLoach, and J. C. García-Ojeda "O-MaSE: a customisable approach to designing and building complex, adaptive multi-agent systems," *Int. J. Agent-Oriented Software Engineering*, vol. 4, no. 3, pp. 244-280, 2010.

[10] N. Spanoudakis, and P. Moraitis, "Using ASEME methodology for model-driven agent systems development," *The 11th Int. Workshop on Agent Oriented Software Engineering (AOSE 2010)*, LNCS 6788, pp. 106-127, 2011.

با گذشت بیش از یک دهه از معرفی مهندسی نرم‌افزار برای سیستم‌های مبتنی بر عامل، در این مقاله تحقیقات مرتبط با مهندسی نرم‌افزار مبتنی بر عامل - به عنوان رویکرد بعدی تولید نرم‌افزار - معرفی شده است. بدین منظور روش تحقیق برای جمع‌آوری مجموعه تحقیقات انجام شده در این زمینه تشریح شده است. با پیروی از روش تحقیق بیان شده، مجموعه‌ای از مجلات و کنفرانس‌های معتبر در این زمینه (در سال‌های ۱۹۹۹-۲۰۰۸) انتخاب و مقالات ارائه شده در آنها به عنوان منابع تحقیق گردآوری شده است. با دستبندی تحقیقات در این زمینه، موضوعات مرتبط با متدولوژی‌ها و فرآیندهای مهندسی نرم‌افزار مبتنی بر عامل، مدل‌سازی و ابزارهای ارائه شده در این حوزه، در این مقاله مورد بررسی و تحلیل قرار گرفته است.

بر اساس بررسی‌های انجام شده می‌توان نتیجه گرفت که موضوعات مرتبط با متدولوژی‌های مهندسی نرم‌افزار، مدل‌سازی و معماری سیستم‌های مبتنی بر عامل در این حوزه بیشترین تحقیقات را به خود اختصاص داده‌اند. این در حالی است که موضوعات مرتبط با تست سیستم‌های مبتنی بر عامل که در سال‌های اخیر مورد توجه قرار گرفته است و ابزارهای پشتیبانی فعالیت‌های مهندسی نرم‌افزار به عنوان تحقیقات مستقل، کمترین سهم را از تحقیقات انجام شده داشته‌اند.

با بررسی تحقیقات انجام شده در زمینه متدولوژی‌ها و فرآیندهای مهندسی نرم‌افزار مبتنی بر عامل می‌توان نتیجه گرفت:

- تحقیقات جاری در این زمینه را می‌توان به دسته‌های زیر تقسیم کرد:
 - بهبود و توسعه متدولوژی‌های موجود مانند Gaia, MaSE, Tropos, PASSI
 - ارائه متدولوژی‌های خاص منظوره
 - ارزیابی متدولوژی‌های موجود
 - حرکت به سمت تلفیق و استفاده از متدولوژی‌ها به صورت شخصی با استفاده از روش‌های method Fragment
- تحقیقات از تنوع و تکرار روش‌ها در سال‌ها اولیه به سمت استانداردسازی و تلفیق متدولوژی‌ها در حرکت است.
- تحقیقات در زمینه فرآیندهای مهندسی نرم‌افزار مبتنی بر عامل به ندرت انجام شده است و نیاز به تحقیقات بیشتری در این زمینه وجود دارد.
- با بررسی تحقیقات انجام شده در زمینه مدل‌سازی برای سیستم‌های مبتنی بر عامل می‌توان نتیجه گرفت:
 - تحقیقات جاری در این زمینه را می‌توان به دسته‌های زیر تقسیم کرد:
 - استفاده از زبان مدل‌سازی UML و ارائه توسعه‌هایی بر آن جهت مدل‌سازی ویژگی‌های عامل و معرفی AUML
 - استفاده از زبان مدل‌سازی *i به عنوان یکی از روش‌های مدل‌سازی عامل‌ها و روابط اجتماعی آنها
 - AML به عنوان یک زبان مدل‌سازی بر مبنای UML
 - ارائه چارچوب‌های خاص برای مدل‌سازی عامل
 - UML و توسعه‌هایی که برای آن ارائه شده‌اند، مبنای اصلی زبان‌های مدل‌سازی سیستم‌های مبتنی بر عامل (مانند AUML و AML) هستند.
 - اگرچه بیشتر تحقیقات اولیه بر روی توسعه‌های UML متمرکز بوده است و در سال‌های اخیر معرفی زبان‌های مدل‌سازی خاص برای سیستم‌های مبتنی بر عامل بیشتر مورد توجه قرار گرفته است.
 - به نظر می‌رسد با توجه به جدید بودن زبانی مانند AML، این زبان زمینه تحقیقاتی مناسبی برای سال‌های آینده باشد.

- Software Engineering and Knowledge Engineering*, vol. 11 no. 3, 2001.
- [24] J. DiLeo, T. Jacobs, and S. A. DeLoach, "Integrating Ontologies into Multiagent Systems Engineering," 4th international bi-conference workshop on agent-oriented Information systems (AOIS 2002), Italy, 2002.
- [25] S. A. DeLoach, "Modeling Organizational Rules in the Multiagent Systems Engineering Methodology," Proceedings of the 15th Canadian Conference on Artificial Intelligence, USA, 2002.
- [26] A. Self, and S. A. DeLoach, "Designing and Specifying Mobility within the Multiagent Systems Engineering Methodology," Proceedings of the Eighteenth Annual ACM Symposium on Applied Computing, (2003).
- [27] S. Vafadar, A. Abdollahzadeh Barfouroush, and M. R. Ayatollahzadeh Shirazi, "Towards a More Expressive and refinable Multiagent System Engineering," 5th international bi-conference workshop on agent-oriented Information systems (AOIS 2003), LNCS 3030, pp. 126-141, 2003.
- [28] S. A. DeLoach, "Multiagent Systems Engineering of Organization-based Multiagent Systems," 4th Int. Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'05), Springer LNCS vol. 3914, pp. 109 – 125, 2006.
- [29] J. C. Garcia-Ojeda, S. A. DeLoach, Robby, W. H. Oyen, and J. Valenzuela, "O-MaSE: A Customizable Approach to Developing Multiagent Development Processes," 8th Int. Workshop Agent-Oriented Software Engineering (AOSE 2007), LNCS 4951, Springer-Verlag, pp. 1-15, 2008.
- [30] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "TROPOS: An Agent-Oriented Software Development Methodology," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 8, no.3, pp. 203-236, 2004.
- [31] P. Giorgini, J. Mylopoulos, and R. Sebastiani, "Goal-Oriented Requirements Analysis and Reasoning in the Tropos Methodology," *J. of Engineering Applications of Artificial Intelligence*, Elsevier, vol. 8, no.2, 2005.
- [32] J. Jureta, S. Faulkner, and P. Schobbens, "Allocating Goals to Agent Roles During MAS Requirements Engineering," 7th Int. Workshop Agent-Oriented Software Engineering (AOSE 2006), LNCS 4405, Springer-Verlag, pp. 19-34, 2007.
- [33] A. U. Mallya, and M. P. Singh, "Incorporating Commitment Protocols into Tropos," 6th Int. Workshop on Agent Oriented Software Engineering (AOSE 2005), LNCS 3950, pp. 69-80, 2005.
- [34] J. Lind, *Iterative Software Engineering for Multiagent Systems: The MASSIVE Method*, LNCS 1994, Springer-Verlag, 2001.
- [35] G. Caire, R. Evans, P. Massonet, W. Coulier, F. J. Garijo, J. Gomez, J. Pavón, F. Leal, P. Chainho, P. E. [11] T. Miller, S. Pedell, L. Sterling, and B. Lu, "Engaging Stakeholders with Agent-Oriented Requirements Modelling," The 11th Intl Workshop on Agent Oriented Software Engineering (AOSE 2010), LNCS 6788, pp. 62-78, 2011.
- [12] A. Sharpanskykh, and J. Treur, "Automated analysis of compositional multi-agent systems," *Int. Journal of Agent-Oriented Software Engineering (IJAOSE)*, vol.4, no.2, pp. 174 – 221, 2010.
- [13] J. C. Garcia-Ojeda, and S. A. DeLoach Robby, "AgentTool Process Editor: Supporting the Design of Tailored Agent-based Processes," In Proceedings of the 2009 ACM Symp on Applied Computing, ACM: New York, 2009.
- [14] A. Tveit, "A survey of Agent-Oriented Software Engineering," Norwegian University of Science and Technology, 2001.
- [15] M. Wooldridge, and P. Ciancarini, "Agent-Oriented Software Engineering: The State of the Art," Agent-Oriented Software Engineering Workshop (AOSE00). Springer-Verlag LNCS1957, pp. 55-82, 2001.
- [16] C. Bernon, M. Cossentino, and J. Pavon, "An Overview of Current Trends in European AOSE Research," *Informatica*, vol. 29, pp. 379-390, 2005.
- [17] S. Vafadar, and A. Abdollahzadeh Barfouroush, "Requirements Engineering for Agent Based Systems," Technical Report, CE/PR/88/01, Computer Engineering Faculty, Amirkabir University of Technology, 2008.
- [18] M. Wooldridge, *Reasoning about Rational Agents*, The MIT Press: Cambridge, MA, 2000.
- [19] F. Zambonelli, N. Jennings, A. Omicini, and M. Wooldridge, "Agent oriented Software Engineering for Internet Applications," *Coordination of Internet Agents: Models, Technologies, and Applications*, Chapter 13. Springer-Verlag.
- [20] T. Juan, A. Pearce, and L. Sterling, "ROADMAP: Extending the Gaia Methodology for Complex Open Systems," The First International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS 2002), ACM Press, pp. 3-10, 2002.
- [21] J. C. García-Ojeda, A. E. Arenas, and J. P. Alcázar, "Paving the Way for Implementing Multiagent Systems: Integrating Gaia with Agent-UML," 6th International Workshop Agent-Oriented Software Engineering (AOSE 2005), Lecture Notes in Computer Science 3950, Springer-Verlag, pp. 179-189, 2005.
- [22] J. Gonzalez-Palacios, and M. Luck, "Extending Gaia with Agent Design and Iterative Development," 8th International Workshop Agent-Oriented Software Engineering (AOSE 2007), LNCS 4951, Springer-Verlag. pp 16-30, 2007.
- [23] S. A. DeLoach, M. Wood, and C. Sparkman, "Multiagent Systems Engineering," *The Int. Journal of*

- [47] A. H. Elamy, and B. Far, "A Statistical Approach for Evaluating and Assembling Agent Oriented Software Engineering Methodologies," 8th international bi-conference workshop on agent-oriented Information systems (AOIS 2006), pp. 105-122, 2006.
- [48] L. Cernuzzi, M. Cossentino, and F. Zambonelli, "Process Models for Agent-based Development," *J. of Engineering Applications of Artificial Intelligence*, vol. 18, pp. 205-222, 2005.
- [49] R. Basseda, A. Moallem, T. Alinaghi, and F. Taghiyare, "Estimation of Complexity in Agent Oriented Methodologies via Evaluation of Models and Artifacts," 12th computer society of Iran computer conference (CSICC2007), 2007.
- [50] Q. N. Tran, B. Henderson-Sellers, and J. Debenham, "Incorporating the elements of the MASE methodology into Agent OPEN," In Proceedings of 6th Int. Conference on Enterprise Information Systems (ICEIS'2004), pp. 380-388, 2004.
- [51] B. Henderson-Sellers, J. Debenham, and Q. N. Tran, "Adding Agent-Oriented Concepts Derived from GAIA to Agent OPEN," 16th International Conference of Advanced Information Systems Engineering., CAiSE 2004, LNCS 3084, Springer-Verlag, Berlin, pp. 98-111, 2004.
- [52] B. Henderson-Sellers, Q. N. Tran, and J. Debenham, "Incorporating elements from the Prometheus agent oriented methodology in the OPEN Process Framework," Proceedings. Of AOIS@CAiSE04, pp. 370-385, 2004.
- [53] B. Henderson-Sellers, J. Debenham, Q. N. Tran, M. Cossentino, and G. Low, "Identification of Reusable Method Fragments from the PASSI Agent-Oriented Methodology," 7th international bi-conference workshop on agent-oriented Information systems (AOIS 2005), pp. 90-97, 2005.
- [54] B. Henderson-Sellers, "Creating a comprehensive agent-oriented methodology - using method engineering and the OPEN metamodel," in Agent-Oriented Methodologies (eds. B. Henderson-Sellers and P. Giorgini), Idea Group, pp. 368-397, 2005.
- [55] G. Beydoun, C. Gonzalez-Perez, G. Low, and B. Henderson-Sellers, "Synthesis of a Generic MAS metamodel," 4th International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'05), Springer LNCS, vol. 3914, pp. 1-5, 2005.
- [56] M. Cossentino, S. Gaglio, A. Garro, and V. Seidita, "Method fragments for agent design methodologies: from standardisation to research," *International Journal of Agent-Oriented Software Engineering*, vol. 1, no.1 pp. 91-121, 2007.
- [57] F. Bergenti, and A. Poggi, "Exploiting UML in the Design of Multi-Agent Systems," Proceedings of the ECOOP - Workshop on Engineering Societies in the Agents' World 2000 (ESAW'00), pp. 96-103, 2000.
- Kearney, and J. Stark, "Agent Oriented Analysis using MESSAGE/UML," LNCS 2222, Springer-Verlag, pp. 119-135, 2002.
- [36] J. Pavón, and J. Gómez-Sanz, "Agent-Oriented Software Engineering with INGENIAS," 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'03), LNCS 2691, Springer Verlag, pp. 394-403, 2003.
- [37] C. Iglesias, M. Garijo, J. Gonzales, and J. R. Velasco, "Analysis and Design of Multi-agent Systems using MAS-CommonKADS," Proceedings of the Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL'97), LNCS 1365, Springer-Verlag, pp. 313-326, 1998.
- [38] M. Cossentino, "From Requirements to Code with the PASSI Methodology," Agent-Oriented Methodologies, chapter IV, pp. 79-106, 2005.
- [39] M. Winikoff, and L. Padgham, "The Prometheus Methodology," Federico Bergenti, Marie-pierre Gleizes and Franco Zambonelli, editors, Methodologies and Software Engineering for Agent Systems, Chapter 11. Kluwer Academic Publishing (New York), 2004.
- [40] J. M. Serrano, and S. Ossowski, "On the Impact of Agent Communication Languages on the Implementation of Agent Systems," 8th International Workshop of Cooperative Information Agents (CIA 2004), LNCS 3191, Springer-Verlag, pp. 92-106, 2004.
- [41] M. Gervais, "ODAC: An Agent-Oriented Methodology based on ODP," *J. of Autonomous Agents and Multi-Agent Systems*, vol. 7, no.3, pp. 199-228, 2003.
- [42] ISO/IEC X.900 (1995). IS 10746-x ITU-T Rec. X90x, ODP Reference Model Part x.
- [43] G. B. Roest, and N. B. Szirbik, "Intervention and Escape Mode: Involving Stakeholders in the agent development process," 7th Int. Workshop Agent-Oriented Software Engineering (AOSE 2006), LNCS 4405, Springer-Verlag pp. 109-120, 2007.
- [44] M. Morandini, L. Penserini, A. Perini, and A. Susi, "Refining Goal Models by Evaluating System Behaviour," 8th Int. Workshop Agent-Oriented Software Engineering (AOSE 2007), LNCS 4951, Springer-Verlag, pp. 44-57, 2008.
- [45] A. Bogdanovych, M. Esteva, S. Simoff, C. Sierra, and H. Berger, "A Methodology for Developing Multiagent Systems as 3D Electronic Institutions," 8th International Workshop Agent-Oriented Software Engineering (AOSE 2007), LNCS 4951, Springer-Verlag, pp. 103-118, 2008.
- [46] G. Jayaputera, A. Zaslavsky, and S. Loke, "Approaches to Dynamically Generated User-Specified MAS," 6th Int. Workshop Agent-Oriented Software Engineering (AOSE 2005), LNCS 3950, Springer-Verlag, pp. 139-153, 2006.

- [70] R. Clote, X. Franch, L. López, J. Marco, N. Seyff, and P. Grünbacher, "On the Meaning of Inheritance in i*," In Proceedings of the 17th International Workshop on Agent-Oriented Information Systems (AOIS'07), published as part of the CAiSE'07 Proceedings of Workshops and Doctoral Symposium, vol. 2, pp. 651-665, 2007.
- [71] R. Cervenka, I. Trencansky, M. Calisti, and D. Greenwood, "AML: Agent Modeling Language. Toward Industry-Grade Agent-Based Modeling," In Odell, J., Giorgini, P., Muller, J., eds.: Agent-Oriented Software Engineering V: 5th International Workshop, AOSE 2004, Springer-Verlag, 2005.
- [72] R. Cervenka, I. Trencansky, and M. Calisti, "Modeling Social Aspects of Multi-Agent Systems -The AML Approach," International 6th International Workshop Agent-Oriented Software Engineering (AOSE 2005), Lecture Notes in Computer Science 3950, Springer-Verlag, pp. 40-53, 2006.
- [73] G. Wagner, "The Agent-Object- Relationship Metamodel: Towards a Unified View of State and Behavior," *J. of Information Systems* 28 (5), pp. 475-504, 2003.
- [74] G. Wagner, "Agent-Object-Relationship Modeling," Second International Symposium – from Agent Theory to Agent Implementation together with EMCRS, 2000.
- [75] A. Sturm, D. Dori, and O. Shehory, "Single-Model Method for Specifying Multi-Agent Systems," Second International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2003), ACM Press, pp. 121-128, 2003.
- [76] A. Omicini, "SODA: Societies and Infrastructures in the Analysis and Design of Agent-Based Systems," Agent-Oriented Software Engineering: First International Workshop, AOSE 2000. Lecture Notes in Computer Science 1957, Springer-Verlag, pp. 185-193, 2001.
- [77] G. Cabri, L. Ferrari, and L. Leonardi, "Supporting the Development of Multi-Agent Interactions via Roles," 6th International Workshop Agent-Oriented Software Engineering (AOSE 2005), Lecture Notes in Computer Science 3950, Springer-Verlag, pp. 40-53, 2006.
- [78] L. Xiao, and D. Greer, "A Hierarchical Agent-oriented Knowledge Model for Multi-Agent Systems," Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'06), 2006.
- [79] R. Hill, S. Polovina, and M. D. Beer, "Improving AOSE with an Enriched Modelling Framework," 6th International Workshop Agent-Oriented Software Engineering (AOSE 2005), Lecture Notes in Computer Science 3950, Springer-Verlag, pp. 94-108, 2006.
- [80] M. Bakhouya, and J. Gaber, "Autonomous Agent Modeling using I/O Automata," 7th International Workshop Agent-Oriented Software Engineering (AOSE 2006), Lecture Notes in Computer Science 4405, Springer-Verlag, pp. 157-168, 2007.
- [58] J. Odell, H. V. D. Parunak, and B. Bauer, "Extending UML for Agents," Agent-Oriented Information Systems (AOIS) Workshop at the 17th National conference on Artificial Intelligence (AAAI), 2000.
- [59] J. Odell, and C. Bock, OMG document ad/99-12-01, "Suggested UML Extensions for Agents," Submitted to the OMG's Analysis and Design Task Force (ADTF) in response to the Request of Information (RFI) entitled "UML2.0 RFI," 1999.
- [60] B. Bauer, J. Müller, and J. Odell, "Agent UML: A Formalism for Specifying Multiagent Interaction," The First International Agent-Oriented Software Engineering, AOSE 2000. Lecture Notes in Computer Science 1957, Springer-Verlag, pp. 91-103, 2001.
- [61] B. Bauer, "UML Class Diagrams Revisited in the Context of Agent-Based Systems," Second International Workshop of Agent-Oriented Software Engineering, AOSE 2001, Lecture Notes in Computer Science 2222, Springer-Verlag, pp. 101-118, 2002.
- [62] H. V. D. Parunak, and J. Odell, "Representing Social Structures in UML," In Proceedings of the fifth international conference on Autonomous agents, 2001.
- [63] B. Bauer, and J. Odell, "UML 2.0 and agents: how to build agent-based systems with the new UML standard," *J. of Engineering Applications of Artificial Intelligence* vol. 18, pp.141-157, 2005.
- [64] FIPA: Foundation for Intelligent Physical Agents, <http://www.fipa.org/specs>, 2005.
- [65] M. Kusek, and G. Jezic, "Extending UML Sequence Diagrams to Model Agent Mobility," 7th International Workshop Agent-Oriented Software Engineering (AOSE 2006), Lecture Notes in Computer Science 4405, Springer-Verlag, pp. 51-63, 2007.
- [66] V. Silva, R. Choren, and C. Lucena, "Using MAS-ML Dynamic Diagrams to Model MAS Behavioral Properties," 8th international bi-conference workshop on agent-oriented Information systems (AOIS 2006), pp. 9-16, 2006.
- [67] E. Yu, "Why Agent Oriented Requirements Engineering," Proceedings of 3rd International Workshop on Requirements Engineering: Foundations for Software Quality (June 16-17, 1997, Barcelona, Catalonia). E. Dubois, A.L. Opdahl, K. Pohl, eds. Presses Universitaires de Namur, 1997.
- [68] E. Yu, "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering (RE'97) pp. 226-235, 1997.
- [69] G. Grau, C. Cares, X. Franch, and F. J. Navarrete, "A Comparative Analysis of i* Agent-Oriented Modelling Techniques," Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'06), 2006.

[92] A. M. Tiryaki, S. Oztuna, O. Dikenelli, R. Erdur, "Sunit: A unit testing framework for test driven development of multi-agent systems," In 7th International Workshop on Agent Oriented Software Engineering, 2006.

[93] G. Caire, M. Cossentino, A. Negri, A. Poggi, and P. Turci, "Multi-agent systems implementation and testing," International symposium – From agent theory to agent implementation, AT2AI, 2004.



احمد عبدالله زاده بارفروش همکاری خود را با دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر در سال ۱۳۷۰ و پس از اخذ مدرک دکتری مهندسی کامپیوتر از دانشگاه بریستول انگلستان، آغاز نموده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می‌باشد. ایشان از سال ۱۳۷۹ الی ۱۳۸۱ به عنوان استاد مدعو در دانشگاه‌های Maryland College Park آمریکا و ORASY (Paris 11) فرانسه و در سال ۱۳۸۸ تا ۱۳۹۰ به عنوان استاد مهمان در دانشگاه‌های Trento ایتالیا و Loughborough انگلستان مشغول به کار بوده است. دکتر عبدالله زاده کتاب‌های "مقدمه‌ای بر هوش مصنوعی توزیع شده" و "کلیات متدولوژی تامین کیفیت" را نیز تالیف نمود. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: تکنیک‌های هوش مصنوعی، هوش مصنوعی توزیع شده، مذاکره خودکار، سیستم‌های خبره، پردازش زبان طبیعی، سیستم‌های تصمیم‌یار، هوش تجاری، پایگاه داده تحلیلی، داده‌کاوی، و مهندسی نرم افزار

آدرس پست‌الکترونیکی ایشان عبارت است از:

ahmadaku@aut.ac.ir



شیوا وفادار تحصیلات خود در رشته مهندسی کامپیوتر (نرم‌افزار) را در مقاطع کارشناسی و کارشناسی ارشد در دانشگاه‌های علم و صنعت ایران و صنعتی امیرکبیر به پایان رسانده و در حال حاضر دانشجوی دکتری این رشته در دانشگاه صنعتی امیرکبیر است. نامبرده در سال ۱۳۸۸ به عنوان دانشجوی مهمان در مرکز تحقیقات FBK دانشگاه Trento ایتالیا مشغول به فعالیت بوده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: مهندسی نرم‌افزار، مهندسی نرم‌افزار مبتنی بر عامل، مهندسی هوش برای سیستم‌های نرم‌افزاری، الگوهای تحلیل و نیازمندی در سیستم‌های نرم‌افزاری و سیستم‌های هوشمند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

vafadar@aut.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۸/۵/۳۱

تاریخ اصلاح: ۹۱/۳/۱۳

تاریخ قبول شدن: ۹۱/۸/۱۵

نویسنده مرتبط: دکتر احمد عبدالله زاده بارفروش، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران.

[81] V. Dignum, A Model for Organizational Interaction: based on Agents, founded in Logic. SIKS Dissertation Series 2004-1. PhD Thesis, Utrecht University, 2004.

[82] M. Mensonides, B. Huisman, and V. Dignum, "Towards Agent-Based Scenario Development for Strategic Decision Support," 8th international bi-conference workshop on agent-oriented Information systems (AOIS 2006), pp. 1-8, 2006.

[83] A. Lapouchnian, and Y. Lesperance, "Modelling Mental States in the Analysis of Multiagent Systems Requirements," 7th International Workshop Agent-Oriented Software Engineering (AOSE 2006), Lecture Notes in Computer Science 4405, Springer-Verlag, pp. 104-121, 2007.

[84] A. Garcia, C. Chavez, and R. Choren, "An Aspect Oriented Modelling Framework for Designing Multi-agent Systems," 7th International Workshop Agent-Oriented Software Engineering (AOSE 2006), Lecture Notes in Computer Science 4405, Springer-Verlag, pp. 35-50, 2007

[۸۵] ا. عبدالله زاده بارفروش، ب. معصومی و م. ر. آیت‌الله زاده شیرازی، مقدمه‌ای بر هوش مصنوعی توزیع شده (معرفی عامل و سیستم‌های چندعامله)، انتشارات جلوه، تهران، ۱۳۸۴.

[86] A. Perini, and A. Susi, "Automating Model Transformations in Agent-Oriented modeling," 6th International Workshop Agent-Oriented Software Engineering (AOSE 2005), Lecture Notes in Computer Science 3950, Springer-Verlag, pp. 167-178, 2006.

[87] G. B. Jayatilleke, L. Padgham, and M. Winikoff, "Model Driven Development Toolkit for Domain Experts to Modify Agent Based Systems," 7th International Workshop Agent-Oriented Software Engineering (AOSE 2006), Lecture Notes in Computer Science 4405, Springer-Verlag, pp. 25-36, 2007.

[88] T. Bosse, C. M. Jonker, and J. Treur, "Requirements Analysis of an Agent's Reasoning Capability," Proceeding of the 7th International Workshop on Agent-Oriented Information Systems, 7th international bi-conference workshop on agent-oriented Information systems (AOIS 2005), 2005.

[89] J. Dehlinger, and R. R. Lutz, "PLFaultCAT: A Product-Line Software Fault Tree Analysis Tool. Automated Software Engineerin," 13(1): 169-193, 2006.

[90] D. C. Nguyen, A. Perini, and P. Tonella, "A Goal-Oriented Software Testing Methodology," 8th International Workshop Agent-Oriented Software Engineering (AOSE 2007), Lecture Notes in Computer Science 4951, Springer-Verlag. pp 58-72, 2008

[91] R Coelho, U. Kulesza, A. von Staa, and C. Lucena "Unit testing in multi-agent systems using mock agents and aspects." Proceedings of the 2006, international workshop on Software engineering for large-scale multi-agent systems, pp. 83-90, ACM Press, 2006.

ارائه چارچوب آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و تحلیل نتایج آن با استفاده از معیارهای کمی

بهشید بهکمال محسن کاهانی سهیلا دهقانزاده

^۱دانشکده فنی و مهندسی، دانشگاه فردوسی مشهد، مشهد، ایران

چکیده

با توجه به رشد روزافزون کاربردهای مختلف نرم‌افزارهای مبتنی بر وب، موضوع کیفیت این سیستم‌ها از اهمیت بالایی برخوردار است و به دلیل تنوع و ناهمگونی در محیط اجرای آنها، روش‌های رایج برای آزمون نرم‌افزارهای متداول برای تست و ارزیابی سیستم‌های تحت وب کارآمد نمی‌باشند. هدف اصلی این مقاله ارائه چارچوبی برای آزمون است که با ویژگی‌های محیطی و اجرایی این سیستم‌ها سازش و تطابق داشته باشد. از این رو ابتدا با مطالعه سیستم‌های مبتنی بر وب و بررسی نیازمندی‌های آزمون این سیستم‌ها چارچوبی برای آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و تحلیل نتایج آن ارائه می‌شود سپس بر اساس معیارهای ارزیابی کیفیت نرم‌افزار، سه معیار کمی به منظور ارزیابی خروجی‌های چارچوب پیشنهاد خواهد شد. در پایان چارچوب پیشنهادی برای آزمون خودکار سه نمونه سیستم مبتنی بر وب با محیط‌های اجرای متفاوت بطور تجربی بکار گرفته شده و نتایج آزمون با استفاده از معیارهای کمی مورد ارزیابی قرار خواهد گرفت.

کلمات کلیدی: آزمون نرم‌افزار، معیارهای کمی، سیستم‌های مبتنی بر وب، خودکارسازی آزمون.

۱- مقدمه

پاسخگویی به کاربران مختلف را فراهم نمایند. همچنین، در سیستم‌های تحت وب، بین برنامه سمت سرور و برنامه سمت کلاینت، ارتباط تنگاتنگی وجود ندارد و اگر برنامه سمت سرور دچار خرابی شود و اجرای آن با مشکل مواجه شود، کاربر سیستم اطلاعات مناسبی برای تشخیص علت خرابی ندارد. این کمبود اطلاعات، عدم آگاهی از جزئیات دقیق وضعیت سیستم، و عدم امکان دستکاری و انجام آزمون و خطا بر روی آن، طبیعتاً بر عملیات تست سیستم تأثیر می‌گذارد. با توجه به ویژگی‌ها و پیچیدگی‌های خاصی که اشاره شد، روش‌ها، ابزارها و مدل‌های رایج برای آزمون نرم‌افزارهای متداول، معمولاً برای تست و ارزیابی سیستم‌های تحت وب، کارآمد نمی‌باشند. برخی از این روش‌ها نیازمند تغییر و تطبیق با محیط وب می‌باشند و برخی نیز بکلی قابل استفاده نمی‌باشند. همچنین برای برخی از موارد آزمون، روش‌ها و مدل‌های جدید که مخصوص سیستم‌های تحت باشند، مورد نیاز است [۲]، [۱] و خودکارسازی فرآیند آزمون راه حل مناسبی برای کاهش هزینه‌های این آزمون می‌باشد [۳].

در حال حاضر، تفاوت بین دو نرم‌افزار را توانایی نرم‌افزارها در برآورده ساختن ویژگی‌های کیفی مورد انتظار که همان کیفیت نرم‌افزار است، تعیین می‌کند. آزمون نرم‌افزار بعنوان یک راه حل مهم در زمینه سنجش کیفیت نرم‌افزار ارائه شده که باید در مراحل مختلف چرخه عمر نرم‌افزار انجام شود. نرم‌افزارهای مبتنی بر وب نیز از این قاعده مستثنی نیستند، ولی به دلیل آنکه مدل رایج برای توسعه این سیستم‌ها، روش توسعه تکاملی تدریجی است، این نرم‌افزارها همواره در حال تغییر و تکامل هستند و زمان بسیار کوتاه ارائه به بازار، موجب می‌شود تا توسعه‌دهندگان سیستم، فرصت کافی برای انجام آزمون‌های لازم را نداشته باشند. از سوی دیگر، محیط اجرای سیستم‌های تحت وب از پویایی و تنوع بسیار بالایی برخوردار است. در نتیجه طراحان سیستم‌های تحت وب، با حجم بالایی از ناهمگونی در محیط اجرای سیستم مواجه هستند که باید به شکلی با مدیریت این مساله، امکان

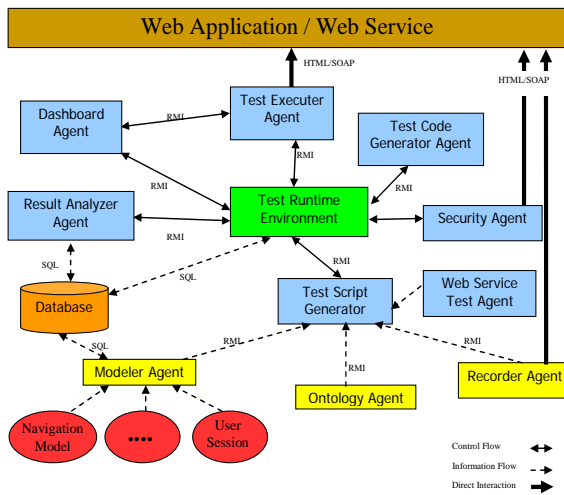
سیستم‌های مبتنی بر عامل در کاربردهای مختلفی که در یک بستر توزیع شده نظیر اینترنت اجرا می‌شوند، مورد استفاده قرار گرفته و با توجه به مزایایی که فراهم می‌کنند، همواره مورد توجه بوده‌اند.

برای پیاده‌سازی یک سیستم مبتنی بر عامل، مناسب‌ترین روش، استفاده از میان‌افزارهایی^۱ می‌باشد که مخصوص این کار طراحی شده‌اند. با استفاده از سرویس‌های پایه‌ای که این میان‌افزارها ارائه می‌کنند، پیاده‌سازی سیستم مورد نظر، از سرعت و کیفیت بالاتری برخوردار خواهد بود. از دیگر نکات مثبت استفاده از این میان‌افزارها، انطباق با استانداردهای مختلف نرم‌افزاری است که در این زمینه مطرح می‌باشد.

در این مقاله از یک میان‌افزار کد منبع باز مطرح در زمینه سیستم‌های مبتنی بر عامل که مبتنی بر زبان برنامه‌نویسی قدرتمند جاوا می‌باشد استفاده شده است.^۲

۲-۱- مدل کلی چارچوب

چارچوب بصورت یک سیستم چند عامله معرفی می‌شود که عامل‌ها در آن برای جداسازی فعالیت‌های مختلف و پویاسازی سیستم بکار می‌روند. عامل‌های مختلف در سرتاسر شبکه توزیع شده و می‌توانند در دسترس باشند یا شبکه را ترک کنند. فرایند آزمون به کمک این عامل‌ها انجام می‌شود و یک بخش مرکزی، مدیریت سیستم را بر عهده دارد. در شکل ۲ مدل کلی چارچوب ارائه شده است.



شکل ۲- مدل کلی چارچوب

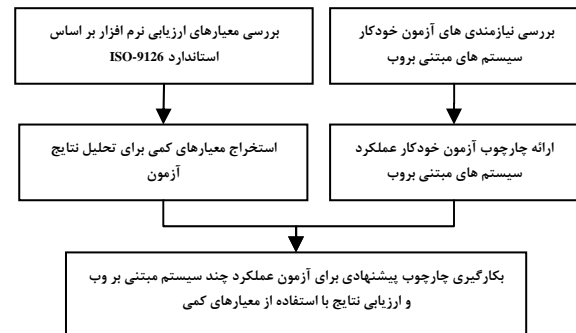
۲-۲- عامل‌های اصلی سیستم

محیط اجرای آزمون^۳، بخش مرکزی سیستم است که با دیگر بخش‌های سیستم ارتباط برقرار کرده تا بتواند وظیفه مدیریت تنظیم و اجرای فعالیت‌های مختلف آزمون را بدرستی انجام دهد. همچنین، این بخش وظیفه فراهم نمودن یک واسط کاربری مناسب برای ایجاد و اجرای آزمون‌های کاربران را برعهده دارد. در ابتدا کاربر سیستم، تمامی تنظیمات مربوط به تست را با استفاده از رابط کاربری عامل محیط اجرای آزمون انجام می‌دهد، سپس عامل محیط اجرای آزمون با توجه به میزبان‌های موجود در سیستم و تعداد عامل‌ها روی هر میزبان، اسکریپت تست مورد نیاز و تنظیمات لازم برای اجرای تست را به میزبان‌ها می‌فرستد.

عامل مولد کد آزمون^۴ اسکریپت آزمون را به عنوان ورودی گرفته و با کامپایل آن کد جاوای متناظر با آن را تولید می‌کند. باید توجه داشت که دو مفهوم مشابه

ساختار مقاله در شکل ۱ آورده شده است، در مقدمه این مقاله با مروری بر ویژگی‌های سیستم‌های مبتنی بر وب، ضرورت آزمون خودکار این سیستم‌ها بررسی شد. در بخش دوم، یک چارچوب چندعامله با استفاده از ابزارهای کد منبع باز معرفی خواهد شد که هدف آن، آزمون خودکار عملکرد سیستم‌های مبتنی بر وب می‌باشد. برای ارزیابی نتایج آزمون لازم است تا معیارهای کمی تعریف شود که در بخش سوم، با بررسی اجمالی معیارهای ارزیابی نرم‌افزار، معیارهای مناسب برای تحلیل نتایج آزمون استخراج خواهد شد.

در پایان، بمنظور بکارگیری عملی چارچوب پیشنهادی، نتایج آزمون خودکار عملکرد سه نمونه سیستم مبتنی بر وب ارائه شده و این نتایج با استفاده از معیارهای کمی استخراج شده، بصورت کمی مورد مقایسه و ارزیابی قرار خواهد گرفت.



شکل ۱- ساختار کلی مقاله

در بخش بعد چارچوب پیشنهادی برای آزمون خودکار عملکرد سیستم‌های مبتنی بر وب ارائه خواهد شد.

۲- ارائه چارچوب پیشنهادی

برای تست سیستم‌های مبتنی بر وب که ذاتاً سیستم‌هایی پیچیده، چند لایه، توزیع شده و پویا می‌باشند، به چارچوبی نیاز داریم که با ویژگی‌های محیطی و اجرایی این سیستم‌ها، سازش و تطابق داشته باشد. طبیعتاً برای این منظور، یک چارچوب توزیع شده، از توانایی‌های بالاتری برخوردار است. با توجه به پیچیدگی‌های قابل انتظار در طراحی چارچوب مورد نظر، استفاده از مدل طراحی مبتنی بر عامل، امکان توزیع و تقسیم کارها و وظایف مختلف چارچوب را فراهم کرده و مزایایی نظیر افزایش انعطاف‌پذیری، مقیاس‌پذیری، و همچنین پویایی سیستم را به دنبال خواهد داشت.

وظایف مختلف سیستم بین عامل‌های مختلف تقسیم شده و این عامل‌ها ضمن تعامل با یکدیگر، اهداف سیستم را برآورده می‌سازند. با طراحی مناسب نقاط تماس عامل‌ها، این امکان وجود دارد که در زمان اجرای چارچوب، عامل‌های جدیدی به آن اضافه شده و یا عامل‌هایی از سیستم حذف شوند. بطور خلاصه، اصولی که در طراحی چارچوب پیشنهادی مدنظر قرار گرفته، بشرح زیر می‌باشد:

- مبتنی بر عامل
- اجرای تست‌ها براساس روش جعبه سیاه
- تهیه موارد آزمون براساس روش‌های جعبه سفید
- استفاده از ابزارهای کد منبع باز موجود (تا حد ممکن)
- پشتیبانی از پروتکل‌ها و تکنیک‌های جدید
- مستقل از سکو بودن

- قابلیت فهم
- عدم همپوشانی بین سطوح مدل (دارای ساختار یک به چند)
- دسته‌بندی جامع و کامل ویژگی‌های کیفیت نرم‌افزار همراه با تعریف واضح و دقیق آنها
- داشتن معیارهای کمی برای اندازه‌گیری ویژگی‌های کیفی

به دلیل ویژگی‌های خاص سیستم‌های مبتنی بر وب که پیش تر به آنها اشاره شد، معمولاً معیارهای متداول برای آزمون سایر نرم‌افزارها، برای آزمون سیستم‌های تحت وب، کافی و کارآمد نمی‌باشند. برخی از این معیارها نیازمند تغییر و تطبیق با محیط وب می‌باشند و برخی نیز بکلی قابل استفاده نمی‌باشند. در این مقاله، به منظور انتخاب معیارهای کمی، فاکتورهای زیر مدنظر قرار گرفته است:

۱. معیارهای کمی مدل استاندارد بر دو دسته‌اند: برخی از آنها برای ارزیابی کیفیت در مراحل مختلف توسعه نرم‌افزار مورد استفاده قرار می‌گیرد و دسته دوم برای ارزیابی محصول نهایی. با توجه به اینکه هدف این چارچوب، آزمون محصول نرم‌افزاری است، از معیارهایی که برای آزمون محصول نهایی ارائه شده، استفاده شده است.
۲. معیارهای کمی ارائه شده برای استفاده فرد آزمون‌گر تعریف شده است و همه معیارها برای بکارگیری توسط عامل نرم‌افزاری قابل استفاده نیست. از آنجایی که قرار است آزمون نرم‌افزار بطور خودکار انجام شود، از معیارهایی که قابلیت ارزیابی توسط عامل نرم‌افزاری دارند، استفاده شده است.
۳. معیارهای ارائه شده در استاندارد برای استفاده ذینفعان مختلف نرم‌افزار طراحی شده است. این ذینفعان عبارتند از توسعه دهنده^۹، نگهدارنده^۸، آزمون‌گر^{۱۰}، طراح واسط کاربر^{۱۱}، تأییدکننده کیفیت^{۱۲}، تأمین کننده^{۱۳} و کاربر^{۱۴}. در این تحقیق از معیارهایی استفاده شده که توسط آزمون‌گر و تاحدی توسعه دهنده و کاربر بکار گرفته می‌شود.
۴. از آنجایی که در این چارچوب، نرم‌افزار از جنبه سطح کیفی عملکرد آن مورد آزمون و ارزیابی قرار می‌گیرد، لذا فقط از معیارهایی که برای اندازه‌گیری ویژگی کیفی عملیاتی بودن^{۱۵} در مدل استاندارد ارائه شده، استفاده شده است.
۵. معیارهای مدل استاندارد حالت کلی و عمومی دارد و نمی‌توان از آنها برای ارزیابی انواع نرم‌افزارها استفاده نمود. لذا از بین آنها، معیارهایی که پس از اصلاح و تغییر برای ارزیابی نرم‌افزارهای مبتنی بر وب مناسب هستند، استخراج شده است.

براساس عوامل پنج گانه فوق، از بین معیارهای مدل استاندارد سه معیار کمی کفایت عملکردی^{۱۶}، فرکانس خطا^{۱۷} و زمان پاسخگویی^{۱۸} که برای آزمون عملکرد نرم‌افزارهای مبتنی بر وب قابل استفاده هستند، استخراج شده است.

• معیار اول: کفایت عملکردی

این معیار برای ارزیابی کفایت عملکردی نرم‌افزار مورد آزمون است و براساس فرمول ۱ اندازه‌گیری می‌شود:

(۱)

$$1 - \frac{\text{number of failed steps executed by all threads on all hosts}}{\text{number of total steps executed by all threads on all hosts}}$$

روی میزبان‌های تعریف شده توسط کاربر آزمون، تعدادی عامل چندین نمونه تست را بصورت همزمان اجرا می‌کنند. هر سناریوی آزمون که شامل چندین مرحله است، در قالب یک نخ^{۱۹} به سرور وصل می‌شود. چنانچه هر مرحله تست از یک نخ عامل، روی یک میزبان شکست بخورد، نشان‌دهنده عدم کفایت عملکردی نرم‌افزار در پاسخگویی به درخواست‌های همزمان است. بنابراین معیار فوق هر چه

در چارچوب وجود دارد: اسکرپیت آزمون و کد آزمون. اسکرپیت آزمون یک نمایش متنی از فرایند آزمون است که با استفاده از سلنیوم و یا یکی از عامل‌های مدل کننده، عامل تست وب سرویس، عامل تست امنیت و یا عامل ایجاد کننده اسکرپیت تست ایجاد شده است، در حالی که کد آزمون کد جاوای اسکرپیت آزمون است که باید برای اجرا ابتدا توسط عامل زمان اجرا کامپایل و نسخه باینری آن به عامل‌های اجرای آزمون فرستاده می‌شود.

عامل اجرای آزمون^۵ وظیفه اجرای آزمون را روی برنامه‌ها یا سرویس‌های وب برعهده دارد. محیط اجرا نسخه اجرایی تولید شده توسط خود را برای اجرا به عامل اجرا تحویل می‌دهد. عامل اجرا آزمون را انجام داده و نتایج آزمون را در فایل ذخیره می‌کند. پس از خاتمه اجرا، عامل اجرا فایل نتیجه را برای محیط اجرا می‌فرستد تا در پایگاه داده سیستم ذخیره کند و در حین اجرای تست نیز هر چند وقت یکبار نتایج تست را برای عامل داشبورد می‌فرستد تا داشبورد را به هنگام کند. (فرکانس بهنگام‌سازی داشبورد ابتدا توسط کاربر تنظیم می‌شود).

عامل داشبورد^۶ در حین اجرای آزمون نتایج تست را به کاربر نشان می‌دهد. این نتایج را عامل اجراکننده تست هر چند وقت یکبار در حین اجرای آزمون برای عامل داشبورد می‌فرستد تا داشبورد یک نمایش بلادرنگ از وضعیت اجرای آزمون و نتایج آن ارائه دهد. این اطلاعات شامل مراحل می‌باشد که عامل اجرا کننده در حال اجرای آن است، اگر یک مرحله از تست شکست بخورد، نمودار میله‌ای عامل داشبورد برای مرحله جاری یک میله هاشور زده شده اضافه می‌کند و اگر این مرحله با موفقیت پاس شود، نمودار میله‌ای عامل داشبورد برای مرحله جاری یک میله آبی رنگ اضافه می‌کند.

محیط آزمون، نتایج آزمون را از عامل اجرا دریافت و به عامل تحلیل گر نتایج^۷ می‌فرستد تا تحلیلی براساس آنچه که کاربر مشخص کرده روی آنها انجام شود. عامل تحلیل گر می‌تواند نتایج آزمون را در قالب گزارشاتی با فرمت‌ها، گراف‌ها و جداول مختلف ارائه دهد. این عامل از معیارهای نرم‌افزاری برای ارزیابی استفاده می‌کند که در بخش بعدی به آن پرداخته خواهد شد.

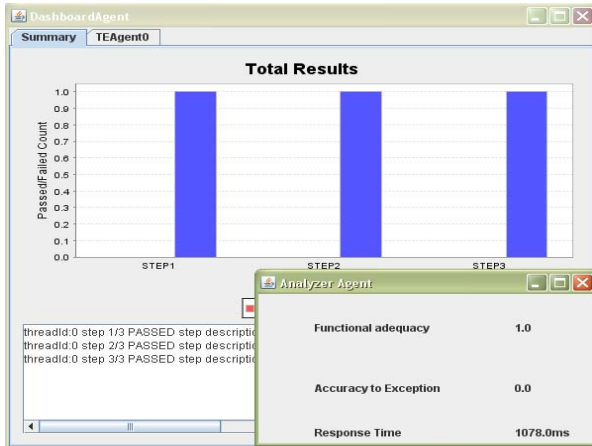
۳- استخراج معیارهای کمی برای تحلیل نتایج آزمون

معیارهای نرم‌افزار برای برای کنترل کیفیت نرم‌افزار، ارزیابی سودآوری سیستم، کنترل پروژه و تخمین نیازمندی‌ها مورد استفاده قرار می‌گیرند. طبقه‌بندی‌های متعددی برای معیارهای نرم‌افزار از دیدگاه افراد مختلف و با توجه به کاربرد آنها در حوزه‌های نرم‌افزارهای مختلف ارائه شده است [۴]، [۵]، [۶]، [۷]، [۸]، [۹] و [۱۰] که بطور خلاصه می‌توان به این موارد اشاره کرد: دسته‌بندی براساس کیفی یا کمی بودن، ایستا یا پویا بودن، تابعی یا شیئی گرابی بودن طراحی، داخلی یا خارجی بودن. از آنجایی که هدف چارچوب پیشنهادی، آزمون عملکرد نرم‌افزار می‌باشد، باید معیارهایی برای ارزیابی عملکرد ارائه شود.

استاندارد ISO/IEC 9126 که در سال ۲۰۰۱ توسط سازمان جهانی استاندارد برای ارزیابی محصول نرم‌افزار و ویژگی‌های کیفیت و راهنمایی‌های لازم برای استفاده از آنها ارائه شده است [۱۱]، یک مدل کیفیت را برای ارزیابی محصول نرم‌افزاری توصیف می‌کند. معیارهای تعریف شده در این مدل، هم برای ارزیابی نیازهای عملکردی و هم نیازهای غیر عملکردی کاربران و خریداران استفاده می‌شود.

با توجه به اینکه این استاندارد در مقایسه با سایر مدل‌های کیفیت نرم‌افزار [۱۲]، [۱۳] برای آزمون خودکار نرم‌افزار مناسب‌تر است، در این مقاله از معیارهای این مدل برای آزمون عملکرد نرم‌افزارهای مبتنی بر وب استفاده شده است. از مهمترین دلایل انتخاب این مدل می‌توان به موارد زیر اشاره کرد:

تمامی مراحل موفقیت‌آمیز باشد و تعداد مراحل شکست خورده برابر صفر و اندازه معیار کفایت عملکردی برابر ۱ و فرکانس رخداد خطا برابر صفر و زمان پاسخگویی نیز خیلی کم باشد که نتیجه آزمون، تایید کننده این انتظار است.



شکل ۳- نتایج آزمون عملکرد خودکار میزبان محلی

همانگونه که در شکل ۳ آمده است، معیار کفایت عملکردی برابر ۱ شده است یعنی در تمامی نمونه‌های اجرا شده تمامی مراحل با موفقیت اجرا شده و معیار فرکانس رخداد خطا نیز برابر ۰ است که نشان‌دهنده عدم رخداد خطا در حین آزمون وب سایت است. همچنین زمان پاسخگویی حدود ۱ ثانیه است.

۴-۲- ارزیابی نتایج آزمون عملکرد سایت دانشکده ریاضی

در یک نمونه تست روی سایت دانشکده ریاضی دانشگاه فردوسی مشهد، تعداد مراحل آزمون و مراحل شکست خورده و موفق در شکل ۴ نشان داده شده است. این آزمون نیز شامل ۹ مرحله بوده است که با توجه به نمودار میله‌ای زیر مراحل ۳ و ۵ و ۹ که بصورت هاشور زده نشان داده شده‌اند، شکست خورده و بقیه مراحل با موفقیت انجام شده است.

کفایت عملکردی نرم‌افزار در اینجا روی شبکه داخلی بررسی می‌شود و با توجه به کمبود ترافیک انتظار می‌رود تعداد مراحل شکست خورده به نسبت کم باشد و معیار کفایت عملکردی تا حد زیادی به واقع نشان‌دهنده کفایت نرم‌افزار است و اندازه آن نیز به یک نزدیک می‌باشد که نشان‌دهنده تعداد کم مراحل شکست خورده است و فرکانس رخداد خطا نیز به نسبت کم است و زمان پاسخگویی نیز تقریباً زیاد است. همانگونه که در شکل ۴ نشان داده شده است، معیار کفایت عملکردی این سایت، پس از ۱۰ بار اجرای آزمون برابر ۰.۶ و فرکانس رخداد خطا ۰.۳ و زمان پاسخگویی آن حدود ۷ ثانیه است.

۴-۳- ارزیابی نتایج آزمون عملکرد وب سایت Alibaba

به عنوان یک نمونه سایت خارجی نیز سناریوی ورود به سایت <http://www.alibaba.com/> با آزمون بار^{۲۱} برای ده کاربر همزمان اجرا شده و نتایج آن در شکل ۵ آمده است. از آنجایی که در این آزمون بستر ارتباطی، شبکه اینترنت بوده و اتصال از طریق پروکسی انجام می‌شود، بدلیل ترافیک شبکه، کفایت عملکردی به خوبی نشان‌دهنده کفایت واقعی نرم‌افزار نیست.

به یک نزدیک تر باشد نشان‌دهنده این است که سیستم از نظر عملکرد از کیفیت بالاتری برخوردار است.

• معیار دوم: فرکانس خطا

این معیار برای ارزیابی فرکانس رخداد خطا در سیستم است و براساس فرمول ۲ اندازه‌گیری می‌شود:

(۲)

$$\frac{\text{number of failed steps}}{\text{time of total testScript Execution}}$$

تعداد مراحل شکست خورده نسبت به زمان اجرای لازم برای اجرای کل مراحل، فرکانس رخداد خطا را نشان می‌دهد. هر چه فرکانس رخداد خطا بیشتر باشد، یعنی قابلیت سرور برای پاسخگویی کمتر است.

• معیار سوم: زمان پاسخگویی

این معیار زمان پاسخگویی متوسط مراحل را نشان می‌دهد و معیاری از زمان پاسخگویی سرور می‌باشد و با استفاده از فرمول ۳ محاسبه می‌شود:

(۳)

$$\frac{\sum_{\text{hosts}} \sum_{\text{thread}} \sum_{\text{steps}} (\text{finishtime} - \text{starttime})}{\text{hosts} * \text{thread} * \text{steps}}$$

زمان شروع و خاتمه هر مرحله آزمون در هر نخ عامل به تفکیک میزبان‌ها در پایگاه داده سیستم ذخیره می‌شود و برای محاسبه متوسط زمان پاسخگویی، تفاضل زمان خاتمه هر مرحله را از زمان شروع همان مرحله محاسبه کرده و سپس همه تفاضل‌ها را با هم جمع کرده و بر تعداد مراحل تقسیم می‌کنیم. بدین ترتیب زمان متوسط پاسخگویی متوسط سیستم بدست می‌آید.

۴-۴- ارزیابی خودکار سه سیستم مبتنی بر وب با استفاده از چارچوب پیشنهادی

در این بخش با استفاده از معیارهای پیشنهادی، نتایج آزمون عملکرد خودکار سه سیستم مبتنی بر وب تحلیل و ارزیابی خواهد شد. برای ارزیابی چارچوب، سه نوع محیط انتخاب شده است، سیستم اول میزبان محلی^{۲۰} است که آزمون در محیطی کاملاً محلی و فاقد هر گونه مانع شبکه و ترافیک انجام می‌شود. سیستم دوم سایت یکی از دانشکده‌های دانشگاه فردوسی است که نرم‌افزار روی بستر شبکه داخلی دانشگاه اجرا می‌شود، و سیستم سوم که یک وب سایت اینترنتی با ترافیک بالاست. با انتخاب این سه سیستم، چارچوب برای ارزیابی نرم‌افزارهای مبتنی بر وب روی بسترهای متفاوت میزبان محلی، شبکه داخلی و اینترنت مورد استفاده قرار می‌گیرد.

۴-۱- ارزیابی نتایج آزمون عملکرد میزبان محلی

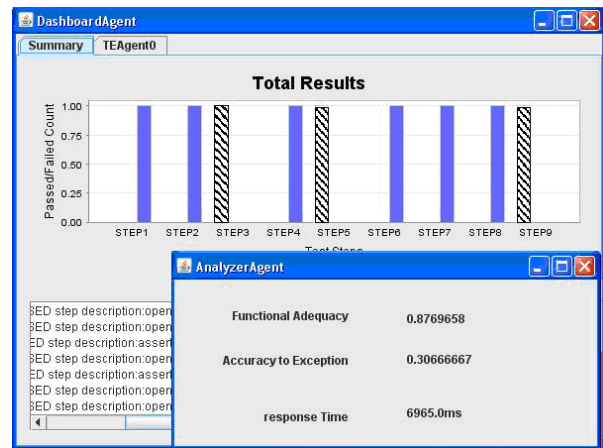
نمونه اول، یک میزبان محلی است که نتایج آزمون و مقادیر معیارهای حاصل از آزمون به همراه داشبورد سیستم، که نشان‌دهنده تعداد مراحل آزمون و مراحل دارای خطا و بدون خطاست، در شکل ۳ مشاهده می‌شود. این آزمون شامل ۳ مرحله بوده که با توجه به نمودار میله‌ای زیر، هر سه مرحله آزمون با موفقیت انجام شده است. بعبارت دیگر به دلیل نبود موانع شبکه و مسیریابی انتظار می‌رود،

سیاسگزاری

این پژوهش با حمایت پژوهشگاه تحقیقات ارتباطات و فناوری اطلاعات انجام گرفته است که بدین وسیله از این مرکز قدردانی می‌گردد. همچنین از اعضای محترم آزمایشگاه فناوری وب دانشگاه فردوسی مشهد^{۲۲} که در طراحی و پیاده‌سازی این چارچوب همکاری نموده‌اند، تشکر و قدردانی می‌گردد.

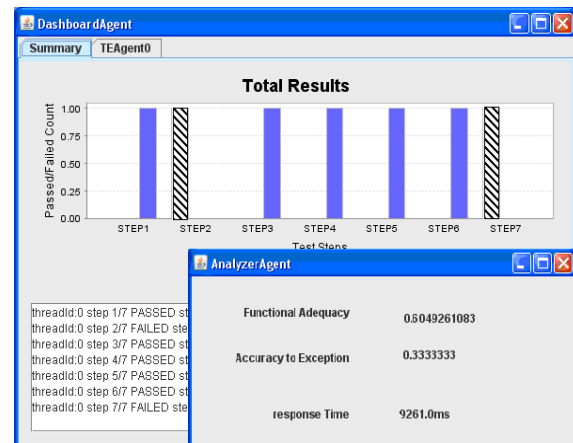
منابع

- [1] N. Mansour, and M. Hourri, "Testing Web Applications," *Information and Software Technology*, vol. 48, pp. 31-42, 2006.
- [2] A. Giuseppe, D. Lucca, and A. R. Fasolino, "Testing Web-based applications: The State of the Art and Future Trends," *Information and Software Technology*, vol. 48, pp. 1172-1186, 2006.
- [3] E. Hieatt, and R. Mee, "Going faster: Testing the Web Application," *IEEE Software*, vol. 19, pp. 60-65, 2003.
- [4] C. Calero, J. Ruiz, and M. Piattini, "Classifying web metrics using the web quality model," *Online Information Review Journal*, vol. 29, no. 3, pp. 645-661, 2005.
- [5] I. B. Hong, "A survey of web site success metrics used by Internet-dependent organizations in Korea," *Internet Research*, vol. 17, no. 3, pp. 272 - 290, 2007.
- [6] R. Pressman, *Software Engineering, a Practitioner's Approach*, McGrawHill, seventh Edition, 2010.
- [7] D. Dhyani, N. W. Keong, and S. Bhowmicka, "Survey of Web Metrics," *ACM Computing Surveys*, vol. 2, no. 3, pp. 1-42, 2002.
- [8] M. E. Wakil, A. Bastawisi, M. Boshra, and A. Fahmy, "Object-Oriented Design-Metrics - A taxonomy," *Third International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA'04)*, 2004.
- [9] M. Piattini, and M. Serrano, "Classifying Web Metrics," *First Intl Workshop on Software Audit and Metrics*, pp. 22-37, 2004.
- [10] E. Mendes, S. Counsell, and N. Mosley, "Towards a Taxonomy of Hypermedia and Web Application Size Metrics," *Proceedings of the ICWE'05 Conference*, Springer LNCS 3579, pp. 110-123, 2005.
- [11] ISO/IEC 9126-1, *Software engineering - Product quality-Part 1: Quality Model*, First edition, 2001.
- [12] A. Moraga, C.I Calero, and M. Piattini, "Comparing different quality models for portals," *Online Information Review Journal*, vol. 30, no. 5, 2006.



شکل ۴- نتایج آزمون عملکرد خودکار وب سایت دانشکده ریاضی دانشگاه فردوسی

بنابراین انتظار می‌رود این معیار بسیار کم باشد و بدلیل مشابه، فرکانس بالای رخداد خطا و زمان پاسخگویی زیاد پیش‌بینی می‌شود.



شکل ۵- نتایج آزمون عملکرد خودکار وب سایت Alibaba

نتایجی که در شکل ۵ آمده است، تاییدکننده پیش‌بینی است و نشان می‌دهد معیار کفایت عملکردی این وب سایت برابر ۰.۸ و فرکانس رخداد خطا برابر ۰.۳ می‌باشد. مدت زمان پاسخگویی وب سایت نیز حدود ۹ ثانیه اندازه‌گیری شده است. این آزمون نیز شامل ۷ مرحله بوه است که با توجه به نمودار میله‌ای فوق، مراحل ۲ و ۷ شکست خورده‌اند و ۵ مرحله دیگر همگی موفقیت‌آمیز بوده است.

۵- نتیجه‌گیری

هدف این مقاله، ارائه چارچوبی برای آزمون خودکار عملکرد سیستم‌های مبتنی بر وب و ارزیابی نتایج این آزمون با استفاده از معیارهای کمی نرم‌افزاری بوده است. برای این منظور، ابتدا با مروری بر ویژگی‌های سیستم‌های مبتنی بر وب، نیازمندی‌های آزمون خودکار این سیستم‌ها، بررسی و سپس چارچوبی برای آزمون خودکار عملکرد این سیستم‌ها ارائه شد. سپس، براساس معیارهای ارزیابی نرم‌افزار، معیارهای مناسب برای تحلیل نتایج آزمون عملکرد استخراج شده و در پایان، با استفاده از معیارهای پیشنهادی، نتایج آزمون عملکرد سه نمونه سیستم مبتنی بر وب مورد ارزیابی قرار گرفت.

- 12 Software Quality Assurance (SQA)
 13 Supplier
 14 User
 15 Functionality
 16 Functional Adequacy
 17 Accuracy to Expectation
 18 Response Time
 19 Thread
 20 Local host
 21 Load Test
 22 <http://wtlab.um.ac.ir>

[13] S. H. Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley Professional, 2009.



بهشید بهکمال دانشجوی دکتری مهندسی کامپیوتر دانشگاه فردوسی مشهد می‌باشد. وی دارای مدرک کارشناسی مهندسی کامپیوتر از دانشگاه فردوسی مشهد و مدرک کارشناسی ارشد مهندسی فناوری اطلاعات از دانشگاه صنعتی امیرکبیر می‌باشد. او از سال ۱۳۸۷ مشغول به تحصیل در مقطع دکتری است و در حال حاضر فعالیت پژوهشی وی متمرکز بر کیفیت داده، وب معنایی و داده‌های پیوندی می‌باشد. آدرس پست‌الکترونیکی ایشان عبارت است از:

behkamal@stu-mail.um.ac.ir



محسن کاهانی دانشیار گروه مهندسی کامپیوتر دانشگاه فردوسی مشهد می‌باشد. ایشان دکتری خود را در زمینه مهندسی کامپیوتر در سال ۱۳۷۷ از دانشگاه ولونگونگ استرالیا اخذ نمودند و از همان سال بعنوان عضو هیئت علمی در دانشگاه فردوسی مشهد مشغول به کار شدند. زمینه‌های تحقیقاتی ایشان مهندسی وب، وب معنایی، بازیابی اطلاعات و مهندسی نرم‌افزار می‌باشد. آدرس پست‌الکترونیکی ایشان عبارت است از:

kahani@um.ac.ir



سهیلا دهقانزاده مدرک کارشناسی و کارشناسی ارشد خود را در رشته مهندسی کامپیوتر از دانشگاه فردوسی مشهد به ترتیب در سال‌های ۱۳۸۷ و ۱۳۹۰ اخذ نموده است. او در حال حاضر دانشجوی دکتری در دانشگاه ملی ایرلند می‌باشد. نکته مرکزی در تحقیقات جاری او پیرامون کاوش وب، تحلیل شبکه‌های اجتماعی و بازیابی اطلاعات در زمینه وب معنایی است. آدرس پست‌الکترونیکی ایشان عبارت است از:

soheila.dehghanzadeh@stu-mail.um.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۸/۷/۷

تاریخ اصلاح: ۹۱/۶/۱۲

تاریخ قبول شدن: ۹۱/۸/۱۵

نویسنده مرتبط: بهشید بهکمال، دانشکده فنی و مهندسی، دانشگاه فردوسی مشهد، مشهد، ایران.

- 1 Middleware
 2 JADE
 3 Test Runtime Environment(TRE)
 4 Test Code Generator Agent (TCGA)
 5 Test Executer Agent (TEA)
 6 Dashboard Agent (DA)
 7 Result Analyzer Agent (RAA)
 8 Developer
 9 Maintainer
 10 Tester
 11 Human Interface Designer

ارائه الگوریتم مناسب برای بخش پردازش سیستم‌های شناسایی رادار ELINT و ESM

علی ناصری

دانشکده فناوری ارتباطات و اطلاعات، دانشگاه جامع امام حسین (ع)، تهران، ایران

چکیده

بخش پردازشگر سیستم‌های شناسایی رادار از جمله ESM، ELINT و RWR نقش کلیدی در کارایی این سیستم‌ها دارد. صورتیکه این موضوع نظر متخصصین زیادی را به خود جلب نموده و تا بحال الگوریتم‌های زیادی نیز ارائه گردیده است. در این مقاله الگوریتمی ارائه می‌شود که دارای دو مرحله آشکارسازی پالس‌های راداری و جداسازی و شناسایی رادارها می‌باشد. مرحله اول الگوریتم قادر به آشکارسازی پالس‌های منتشره از کلیه رادارهای پالسی (رادار با فاصله تکرار پالس ثابت، رادار با فاصله تکرار پالس stagger، رادار با فاصله تکرار پالس jitter، رادار با فاصله تکرار پالس sliding، رادار با فاصله تکرار پالس dwell and switch، رادار با فاصله تکرار پالس periodic، رادار با فاصله تکرار پالس scheduled، رادار با قابلیت ارسال pulse group، رادار با پرش فرکانسی و رادار با فرکانس حامل دو تایی) و رادارهای از نوع سیگنال موج پیوسته می‌باشد و مرحله دوم الگوریتم با استفاده از شبکه‌های عصبی عمل جداسازی و شناسایی رادارها را انجام می‌دهد. نتایج ارزیابی الگوریتم که از ۱۰۰۰ بار دریافت اطلاعات عملی رادارهای منطقه خلیج فارس بدست آمده نشان می‌دهد که این الگوریتم قادر است با دقتی معادل ۹۷/۰۲٪ رادارها را شناسایی نماید.

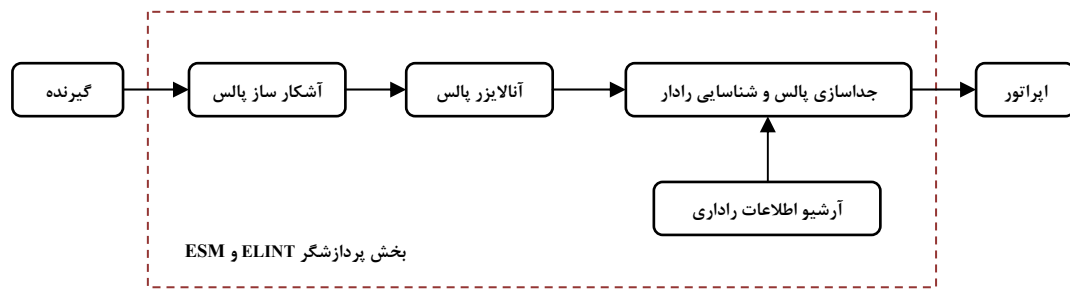
کلمات کلیدی: سطح آستانه، آشکارسازی پالس، جداسازی پالس، شناسایی رادار، هیستریزیس، مینیمم ماکزیمم، نرخ آلامر خطا، احتمال آشکارسازی، شبکه عصبی.

۱- مقدمه

سطح آستانه می‌باشد (تخمین نامناسب سطح آستانه منجر به افزایش نرخ آلامر خطا و یا کاهش دقت آشکارسازی پالس‌های متداخل دریافتی می‌شود). روش‌های وفقی مختلفی به منظور تعیین سطح آستانه و آشکارسازی پالس‌های راداری در سیستم‌های رادار ارائه شده است [۲-۸]. روش‌های مختلفی نیز برای جداسازی و آشکارسازی رشته پالس‌ها تاکنون ارائه گردیده که از مهمترین آنها می‌توان به الگوریتم‌های هیستوگرام Δ - τ ، هیستوگرام CDIF، هیستوگرام SDIF، جداسازی به روش تاکسونومی، جداسازی زاویه‌ای، جداسازی حول راهبر (Leader)، جداسازی با استفاده از انتخاب راهبر در هر منطقه و جداسازی مرحله‌ای (چند پارامتری) اشاره نمود [۱۰، ۱۴].

نتایج شبیه‌سازی‌های صورت گرفته روی روش‌های قبلی نشان داد روش‌های مذکور بدلیل تغییر تکنولوژی ساخت رادارها که منجر به تغییر پارامترهای رادار گشته هیچکدام دقتی بالاتر از ۸۵٪ ندارند. این موضوع در هنگام نبرد می‌تواند باعث اشتباه گردد که خسارت جبران‌ناپذیری را به بار خواهد آورد.

سیگنال‌های دریافتی توسط سیستم‌های شناسایی رادار شامل رشته پالس‌های متداخل منتشر شده توسط رادارهای فعال در محیط است (تقریباً دو میلیون پالس در ثانیه). در این سیستم‌ها ابتدا آشکارسازی پالس‌ها انجام می‌گیرد و سپس برای شناسایی رادارها عمل خوشه‌بندی انجام گرفته و مشخصات هر خوشه بدست می‌آید. در نهایت مشخصات هر خوشه با یک پایگاه اطلاعاتی از قبل تهیه شده از رادارهای منطقه مقایسه می‌شود تا رادار متناظر با رشته پالس بدست آمده، شناسایی گردد [۱]. بلوک دیاگرام کلی یک سیستم شناسایی رادار در شکل ۱ نشان داده شده است. دقت سیستم شناسایی رادار به ترتیب اولویت به دقت سه بخش آشکارساز پالس، جداسازی رشته پالس‌ها و شناسایی رشته پالس‌ها بستگی دارد. عامل مؤثر در تعیین دقت و کارایی بخش آشکارساز پالس تخمین مناسب



شکل ۱- بلوک دیاگرام کلی سیستم شناسایی رادار

ب) روش مینیمم ماکزیمم

روش مینیمم ماکزیمم مبتنی بر استفاده از یک سطح آستانه در طول هر بخش ۱ از سیگنال می‌باشد در این روش ابتدا سیگنال به بخش‌های مساوی تقسیم‌بندی شده و هر بخش به فریم‌هایی با طول N تقسیم می‌گردد. سپس حداکثر دامنه در هر فریم را بدست آورده و بین ماکزیمم‌های بدست آمده می‌نیمم‌گیری به عمل می‌آید. در واقع می‌نیمم بدست آمده سطح نویز را تخمین زده و سطح آستانه تطبیقی برابر M دی‌بی بالاتر از سطح مذکور انتخاب می‌گردد. الگوریتم روش مذکور در شکل ۲ نشان داده شده است. در این الگوریتم، طول زمانی فریم (N) باید از حداقل فاصله زمانی تکرار پالس‌ها کوچکتر باشد تا عمل یافتن می‌نیمم ماکزیمم‌ها در هر بخش زمانی بدرستی صورت گیرد، بنابراین:

$$T_s N \ll PRI_{\min} \Rightarrow \frac{N}{f_s} \ll \frac{1}{PRF_{\max}} \Rightarrow N \ll \frac{f_s}{PRF_{\max}} \quad (3)$$

که در آن f_s فرکانس نمونه‌برداری و $T_s = \frac{1}{f_s}$ می‌باشد. ضمناً بمنظور تخمین بهتر و دقیقتر توان نویز باید طول فریم‌ها حتی‌الامکان بزرگ در نظر گرفته شود. بنابراین باید بین دو شرط مذکور یک نوع مصالحه صورت گیرد.

۳- شبکه‌های عصبی SOFM و RBF

الف) شبکه عصبی SOFM

شبکه‌های عصبی SOFM شبکه‌هایی با ویژگی خودسازمانده هستند و معمولاً با نام‌های شبکه‌های خودسازمانده یا نگاشت‌های ویژگی خود سازمانده یا شبکه‌های عصبی کوهونن نیز بیان می‌شوند. آموزش شبکه‌های عصبی خود سازمانده می‌تواند با نظارت یا بدون نظارت باشد. برای جداسازی رشته پالس‌های متداخل چون تعداد خروجی‌ها معلوم نیست باید از آموزش بدون نظارت استفاده گردد. معماری شبکه‌های SOFM در شکل ۳ نشان داده شده است. این شبکه شبیه شبکه‌های رقابتی می‌باشد و تنها فرق آن در این است که در اینجا از بایاس استفاده نمی‌گردد.

این شبکه‌ها قادرند تا گروه‌هایی از بردارهای مشابه ورودی را تشخیص دهند. در این شبکه‌ها علاوه بر دسته‌بندی بردارهای ورودی نورون‌های همسایه نیز قسمت‌های مجاور از فضای ورودی را تشخیص می‌دهند. در شبکه SOFM در ابتدا وزن‌ها بصورت تصادفی انتخاب شده و سپس الگوهای ورودی به شبکه داده می‌شود. آنگاه شبکه باید خود را بگونه‌ای مرتب کند که وزن‌ها به مختصات مربوط

بنابراین لازم بود الگوریتمی بومی که دارای مزایای تکنیک‌های قبلی و تا حدودی رفع نقایص آنها باشد ارائه می‌گردد که در ادامه به آن پرداخته می‌شود. در این مقاله برای بخش پردازشگر سیستم‌های شناسایی رادار الگوریتمی مبتنی بر شبکه‌های عصبی و مباحث آماری ارائه می‌شود.

در ادامه در بخش ۲ به روش‌های هیستریزس و مینیمم ماکزیمم، بخش ۳ به شبکه‌های عصبی خود سازمانده (SOFM) و شبکه‌های عصبی با تابع پایه شعاعی (RBF)، بخش ۴ به الگوریتم پیشنهادی برای پردازشگر سیستم‌های شناسایی رادار، بخش ۵ به ارزیابی روش پیشنهادی و بخش ۶ به نتیجه‌گیری پرداخته می‌شود.

۲- روش هیستریزس و روش مینیمم ماکزیمم

الف) روش هیستریزس

این روش مبتنی بر استفاده از دو سطح آستانه بالا و پائین می‌باشد. اگر سیگنال دریافتی (سیگنال ویدئو) از سطح آستانه بالایی بیشتر شد نشانگر شروع پالس و هنگامی که از سطح آستانه پایینی کمتر شد نشانه انتهای پالس خواهد بود. برای فهم بهتر مسئله و البته نزدیک شدن به حالت واقعی‌تر خروجی ویدئوی سیستم بصورت نمونه‌برداری شده (توسط یک A/D) در نظر گرفته می‌شود. در این صورت سطح آستانه نمونه بعدی از دو قانون زیر تعیین می‌شود.

- اگر نمونه بعدی بزرگتر از سطح آستانه بالایی (T_H) بود، آنگاه سطح آستانه به مقدار پائینی‌اش (T_L) تغییر می‌یابد.
 - اگر نمونه بعدی کوچکتر از سطح آستانه پائینی (T_L) بود، آنگاه سطح آستانه به مقدار بالایی‌اش (T_H) تغییر می‌یابد.
- نکته‌ای که در دو قانون فوق وجود دارد نحوه تعیین T_L و T_H می‌باشد. بدین منظور باید سطح آستانه اولیه T_S معلوم باشد و T_H و T_L با توجه به تغییرات نویز تعیین شوند.

$$T_S = \frac{T_L + T_H}{2} \quad (1)$$

بعنوان مثال اگر سطح آستانه اولیه ۱۶ باشد و تغییرات نویز در حدود ۴ باشد، دو سطح سطح آستانه برابر خواهند بود با [۱۰]:

$$\begin{aligned} T_H &= 16 + 4 = 20 \\ T_L &= 16 - 4 = 12 \end{aligned} \quad (2)$$

$$x(t) = [x_1(t) \ x_2(t) \ \dots \ x_n(t)] \quad (5)$$

بهترین نرون (J) که کمترین فاصله اقلیدسی را دارد. طبق رابطه ۶ انتخاب می‌شود:

$$J = \arg \min \{ \|x(t) - W_j(t)\| \}, j = 1, 2, \dots, n \quad (6)$$

- بهینه‌سازی وزن‌ها

بعد از اینکه نرون J مرکز همسایگی و نرون‌هایی که تا فاصله N_c در همسایگی نرون مرکزی قرار دارند مشخص شدند. وزن‌های نرون J و نرون‌های همسایه با آن مطابق با رابطه ۷ تنظیم می‌گردد.

$$W_j(t+1) = W_j(t) + k[x(t) - W_j(t)] y(t) \quad (7)$$

که در آن $y(t)$ خروجی نرون در لحظه t می‌باشد. در این حالت وزن‌های سایر نرون‌ها تغییر نمی‌کنند. تصمیم‌گیری در مورد اندازه N_c و k مهم است و لازم است هر دو با زمان کاهش پیدا کنند.

- همگرایی نهایی

در این مرحله با کاهش N_c و k دقت وزن‌ها بهبود می‌یابد. در طی مرحله همگرایی نهایی k ممکن است از 0.1 تا 0 کاهش یابد در حالی که N_c در یک باقی بماند. این مرحله بسته به دقت موردنظر ممکن است چندین برابر بیشتر از مرحله قبل بطول انجامد.

ب) شبکه RBF

شبکه‌های عصبی مصنوعی RBF شبکه‌هایی دو لایه با توابع فعال‌سازی پایه شعاعی هستند، که توسط محققین برای کاربردهای متفاوت پردازش سیگنال پیشنهاد شده‌اند. تابع پایه شعاعی (RBF) یک تابع چند بعدی است که به فاصله بین بردار ورودی و بردار مرکزی بستگی دارد. در شبکه‌های RBF توابع پایه غیر خطی می‌توانند به شکل‌های مختلف، تابع گوسی، تابع چندضلعی، تابع چندضلعی معکوس و تابع کاوچی (Cauchy) باشند.

در کاربردهای عملی اغلب از تابع گوسی استفاده می‌شود که به شبکه‌های عصبی RBF گوسی (GRBF) مشهور است. ساختار کلی شبکه‌های عصبی مصنوعی RBF در شکل ۴ نشان داده شده است. این شبکه دارای سرعت آموزش بالا بوده و برای کاربردهای پردازش بلادرنگ مناسب است. دو نوع مختلف از شبکه‌های RBF شبکه‌های رگرسیون (GRNN) و شبکه‌های احتمالی (PNN) هستند. از شبکه‌های GRNN اغلب در تخمین توابع و شبکه‌های PNN در مسائل دسته‌بندی استفاده می‌شود.

در شبکه عصبی PNN زمانی که بردار ورودی به شبکه اعمال می‌شود، لایه اول فاصله بردار ورودی را از ورودی‌های آموزشی محاسبه می‌کند. و به این ترتیب برداری را فراهم می‌آورد که عناصر آن تعیین کننده میزان فاصله بین ورودی و ورودی آموزشی هستند. لایه دوم با استفاده از خروجی لایه اول برداری از احتمال‌ها را بعنوان خروجی شبکه تولید می‌کند. در نهایت تابع انتقال رقابتی موجود در لایه دوم حداکثر مقدار احتمال‌ها را از بردار احتمال‌ها انتخاب می‌کند، و به ازای آن خروجی یک و به ازای بقیه احتمال‌ها خروجی صفر تولید می‌نماید [۱۹، ۲۲].

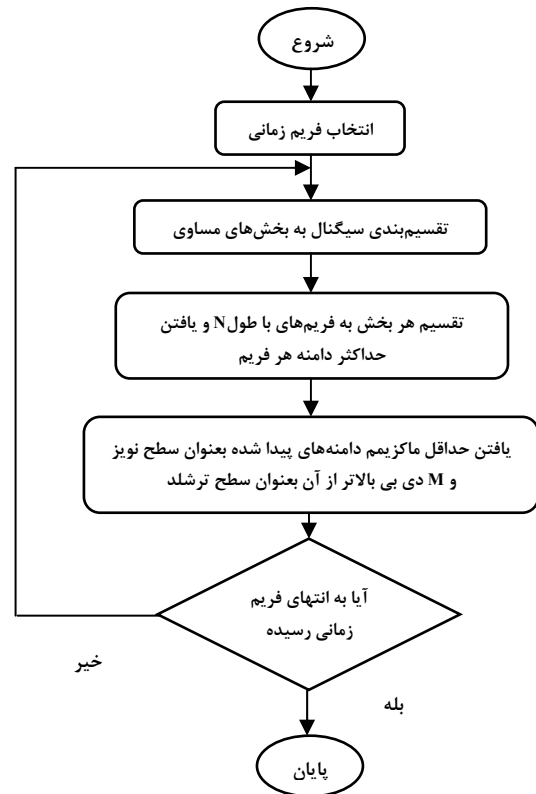
شوند و مکان عنصر متناظر با مکانش در سیستم مختصات باشد. در این شبکه فاصله بین نرون‌ها از روی مکان‌هایشان و با استفاده از تابع فاصله محاسبه می‌شود. برای این منظور چندین تابع فاصله قابل استفاده است اما اغلب از فاصله اقلیدسی استفاده می‌شود. الگوریتم SOFM بصورت زیر است [۱۵، ۱۸، ۲۲].

- مقداردهی اولیه وزن‌ها و پارامترها

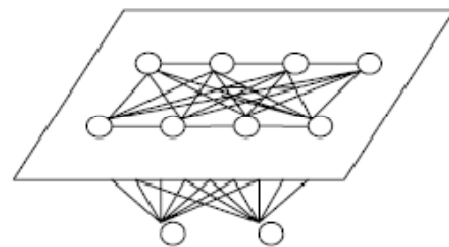
وزن‌ها برای همه نرون‌های ورودی با اعداد تصادفی کوچک بصورت رابطه ۴ مقداردهی اولیه می‌شوند.

$$W_j(0) = [W_{j1}(0) \ W_{j2}(0) \ \dots \ W_{jn}(0)] \quad (4)$$

نرخ یادگیری اولیه و شعاع همسایگی نیز قبل از آموزش تنظیم می‌شود.



شکل ۲- فلوچارت روش می‌نیم- ماکزیم گیری



شکل ۳- معماری شبکه عصبی SOFM

- انتخاب نرون نگاشت دهنده

در زمان t الگوی ورودی $x(t)$ با بردار اولیه نرون نگاشت دهنده $W_j(t)$ با استفاده از فاصله اقلیدسی $\|x(t) - W_j(t)\|$ مقایسه می‌شود که در آن $x(t)$ بصورت رابطه ۵ است.

ابتدا قدرت نویز سیگنال ویدئو دریافتی محاسبه شده و سپس مقدار آن بعلاوه M دی بی می‌شود). از آنجا که حداقل عرض پالس در رادارهای عملی $PW_{\min} = 50ns$ می‌باشد لذا طول زمانی نمونه‌های بالای سطح آستانه معادل $j_s \geq 50ns * f_s$ است. و از آنجا که حداقل فاصله دو پالس متوالی ۲ در رادارهای عملی برابر 500ns است لذا طول زمانی نمونه‌های زیر سطح آستانه معادل $j_n \geq (500ns) * f_s$ است.

در الگوریتم پیشنهادی شکل ۵ برای جداسازی، ابتدا رشته پالس متداخل دریافتی به شبکه عصبی SOFM اعمال می‌گردد. این شبکه مطابق پارامتر AOA اقدام به خوشه‌بندی می‌نماید. در شبکه عصبی SOFM فاصله اقلیدسی بین الگوی ورودی X_i و وزن‌های سلول w_k از رابطه ۸ بدست می‌آید.

$$d_{kj} = |x_i - w_k| \quad (8)$$

اگر این فاصله برای یک نرون از مقدار ζ بیشتر باشد آنگاه آن نرون به ورودی نزدیک نبوده و باید فاصله وزن‌های نرون بعدی از الگوی ورودی محاسبه گردد. نرونی که کمترین فاصله را با الگوی ورودی داشته باشد برنده می‌شود. برای پارامتر AOA فاصله اقلیدسی طبق رابطه ۹ می‌باشد.

$$d_{kj} = |AOA_i - w_k| \quad (9)$$

نتایج بررسی‌ها نشان داد که مقدار ζ در جداسازی پالس‌های متداخل براساس پارامتر AOA برابر ۲/۰۶ می‌باشد. طبق رابطه ۹ بعد از آموزش شبکه عصبی، آمین پالس دریافتی در k امین سلول قرار خواهد گرفت.

حال برای تشخیص رادارهای نزدیک بهم که ممکن است تغییرات AOA آنها به اندازه ای کوچک باشد که پالس‌های آنها در یک خوشه قرار گیرد، دوباره عمل جداسازی روی رشته پالس‌های خوشه‌های قبلی با توجه به پارامتر PA انجام می‌گیرد. در انتهای این بخش تعدادی خوشه از پالس‌ها استخراج می‌گردد که هر کدام از آنها معمولا مربوط به یک رادار خواهد بود. بعد از عمل خوشه‌بندی پالس‌ها به کمک پارامترهای پالس در هر خوشه PDW محاسبه می‌شود. برای شناسایی هر خوشه PDW به شبکه RBF اعمال می‌گردد و این شبکه بعد از آموزش اقدام به شناسایی رادار مربوط به هر خوشه می‌نماید [۲۱، ۱۸، ۱۵].

برای آموزش شبکه RBF می‌توان هر کدام از پارامترهای استخراج شده از خوشه‌ها را استفاده کرد. بعنوان نمونه بکمک سه پارامتر RF و AOA و PW با تعریف ماتریس ورودی بصورت رابطه ۱۰ می‌توان N رادار مختلف را به شبکه RBF آموزش داد.

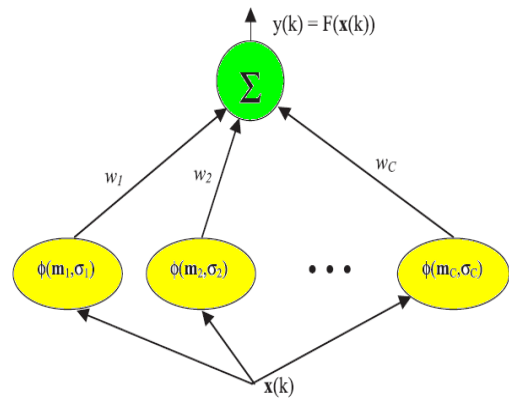
$$P = \begin{bmatrix} RF1 & RF2 & \dots & RFN \\ PW1 & PW2 & \dots & PWN \\ AOA1 & AOA2 & \dots & AOAN \end{bmatrix} \quad (10)$$

با توجه به اینکه اعداد مربوط به RF خیلی بزرگ و اعداد مربوط به PW خیلی کوچک هستند، لذا برای آموزش راحت‌تر شبکه باید آنها را بین صفر تا یک نرمالیزه نمود.

حال اگر N رادار متفاوت از همدیگر باشند آنگاه ماتریس هدف را نیز می‌توان به صورت رابطه ۱۱ نوشت:

$$T = [1 \ 2 \ 3 \ 4 \ \dots \ N] \quad (11)$$

با داشتن ماتریس ورودی و ماتریس هدف می‌توان شبکه را آموزش داده و نوع رادار تشعشع‌کننده را شناسایی نمود [۲۲، ۱۶].



شکل ۴- معماری شبکه عصبی RBF

۴- الگوریتم پیشنهادی برای پردازشگر سیستم‌های شناسایی رادار

در سیستم‌های شناسایی رادار به دلیل معلوم نبودن مشخصه‌های سیگنال ورودی دریافتی برای آشکارسازی پالس‌ها نمی‌توان از روش‌های وقتی استفاده نمود [۹]. شبیه‌سازی‌های انجام شده با نرم‌افزار مطلب برای دو روش مطرح در سیستم‌های شناسایی رادار (هیستریزس و مینیمم ماکزیمم) نشان می‌دهد که روش هیستریزس به دلیل وابستگی آن به انتخاب سطح نویز و روش مینیمم ماکزیمم به دلیل عدم عملکرد صحیح در محیط‌های با چگالی پالس بالا کارایی مناسبی ندارند. در الگوریتم پیشنهادی برای آشکارسازی پالس‌ها از روش ترکیبی هیستریزس و مینیمم ماکزیمم استفاده شده است. ضمناً برای جداسازی پالس‌ها و شناسایی رادارها از دو شبکه عصبی SOFM و RBF استفاده می‌گردد.

از بین انواع مختلف شبکه‌های عصبی، شبکه SOFM به دلیل داشتن دقت بالا برای مسئله جداسازی انتخاب شد و شبکه عصبی RBF بدلیل زمان طراحی کمتر، سرعت بالای آموزش، رفتار موثرتر و تولید مرزهای تصمیم بهتر برای شناسایی رادارها انتخاب گردید [۱۵، ۱۸، ۱۹، ۲۲]. از بین پارامترهای پالس دو پارامتر زاویه ورود (AOA) و دامنه پالس (PA) بدلیل تغییرات کم آنها بهترین پارامتر برای خوشه‌بندی می‌باشد. پارامتر AOA بدلیل سرعت تغییرات کم آن نسبت به سرعت پردازش و پارامتر PA نیز بعلت تغییرات ناچیز در پالس‌های مجاور برای این موضوع انتخاب گردید [۱۰، ۱۳، ۱۴].

شکل ۵ الگوریتم پیشنهادی برای بخش پردازش سیستم‌های شناسایی رادار را نشان می‌دهد. پارامترهای استفاده شده در الگوریتم مذکور در جدول ۱ بیان شده است.

همانطور که در شکل ۵ نشان داده شده است ابتدا فریم زمانی از سیگنال ویدئو انتخاب می‌گردد. مقدار این فریم زمانی توسط اپراتور و با توجه به محیط‌های راداری عملی انتخاب می‌شود. تخمین سطح آستانه نمونه از دو قانون زیر تبعیت می‌کند:

• اگر نمونه بزرگتر از سطح آستانه بالایی (T_H) بود، آنگاه سطح آستانه به مقدار پائینی‌اش (T_L) تغییر می‌یابد.

اگر نمونه کوچکتر از سطح آستانه پائینی (T_L) بود، آنگاه سطح آستانه به مقدار بالایی‌اش (T_H) تغییر می‌یابد.

تخمین دقیق مقادیر T_H ، T_L و T_s در بالا بردن دقت روش هیستریزس بسیار مهم است. این مقادیر وابسته به قدرت نویز است که در این الگوریتم این مقادیر با اعمال روش مینیمم ماکزیمم روی سیگنال ویدئو ورودی تخمین زده می‌شود (نحوه تخمین سطح آستانه اولیه T_s یا $Th(0)$ بدین صورت است که

جدول ۱- پارامترهای مورد استفاده در الگوریتم پیشنهادی

پارامتر	توصیف
Y_{in}	سیگنال خروجی ویدئو
$Y_{in}(i)$	i امین نمونه از سیگنال Y_{in} نمونه برداری شده
f_s	فرکانس نمونه برداری A/D
$Th(i)$	آستانه نمونه i ام
J_n	طول زمانی نمونه های زیر سطح آستانه
Sigma	جذر قدرت نویز ورودی (انحراف معیار استاندارد نویز)
J_s	طول زمانی نمونه های بالای سطح آستانه
AOA	زاویه ورود پالس
PA	دامنه پالس
M	تعداد خوشه ها براساس پارامتر AOA
M	تعداد خوشه ها براساس پارامتر PA در هر خوشه i
Aj	دامنه پالس های خوشه i
δ	دقت جداسازی براساس دامنه پالس
dkj	فاصله اقلیدسی بین زاویه ورود پالس i از وزن های نرون k (خوشه k)
PDW	کلمه توصیف کننده پالس
wk	وزن نرون k که نمایانگر خوشه k می باشد.

۵- ارزیابی الگوریتم پیشنهادی

الف) بخش آشکارسازی الگوریتم پیشنهادی

بخش آشکارسازی پالس های راداری در الگوریتم پیشنهادی با نرم افزار مطلب شبیه سازی شد. ضمناً یک سیمولاتور محیط راداری که قادر به تولید سیگنال ویدئو ۴۰ نوع رادار مختلف می باشد تهیه گردید که سیگنال های ورودی الگوریتم پیشنهادی را تولید می نماید. در شبیه سازی صورت گرفته مطابق مرجع [۱۱]، $M=14dB$ انتخاب گردیده است. همانطور که در الگوریتم پیشنهادی بیان شد تخمین سطح نویز یا انحراف معیار نویز (Sigma) با روش مینیمم ماکزیمم صورت می گیرد. الگوریتم پیشنهادی روی محیط های مختلف راداری اعمال گردید که به دلیل حجم زیاد آنها درج آنها در مقاله میسر نبوده و برای نشان دادن عملکرد الگوریتم به اعمال روی چهار محیط راداری زیرسند می شود. این چهار محیط عبارتند از:

محیط ۱: رادار از نوع فاصله تکرار پالس ثابت

محیط ۲: رادار از نوع فاصله تکرار پالس استگر

محیط ۳: رادار از نوع فاصله تکرار پالس ترکیبی

محیط ۴: رادار از نوع موج پیوسته

شکل ۶ سیگنال های ویدئو چهار نوع محیط راداری مذکور و مقادیر تخمین زده شده سطح آستانه با روش مینیمم ماکزیمم را نشان می دهد. همانطور که در

شکل دیده می شود طول پنجره زمانی ۳ میلی ثانیه انتخاب شده است. این فریم به سه بخش مساوی و هر بخش آن به ۴۰۰ فریم تقسیم می شود. تعداد نمونه های هر فریم ۱۰۰ انتخاب گردیده است.

شایان توجه است مقادیر مذکور با بررسی های آماری نتایج بدست آمده از اعمال الگوریتم پیشنهادی روی محیط های مختلف راداری حاصل گردیدند. همانطور که در شکل ۶ دیده می شود برای هر بخش یک سطح آستانه (مجموعاً ۳ سطح) با توجه به نمونه های فریم ها تخمین زده می شود. شکل ۷ نحوه آشکارسازی پالس های راداری چهار محیط شکل ۶ با الگوریتم پیشنهادی را نشان می دهد. همانطور که در این شکل دیده می شود الگوریتم پیشنهادی از سه سطح آستانه T_s ، T_H و T_L استفاده نموده است. لازم به توضیح است این روش در پنجره های 300µs ای اعمال شده است.

انتخاب عدد 300µs بدان علت است که محدوده تغییرات مشخصه های سیگنال های پالسی در رادارهای عملی بصورت زیر است:

$$PW : 50ns \sim 250ns,$$

$$PA : -60dBm \sim +5dBm,$$

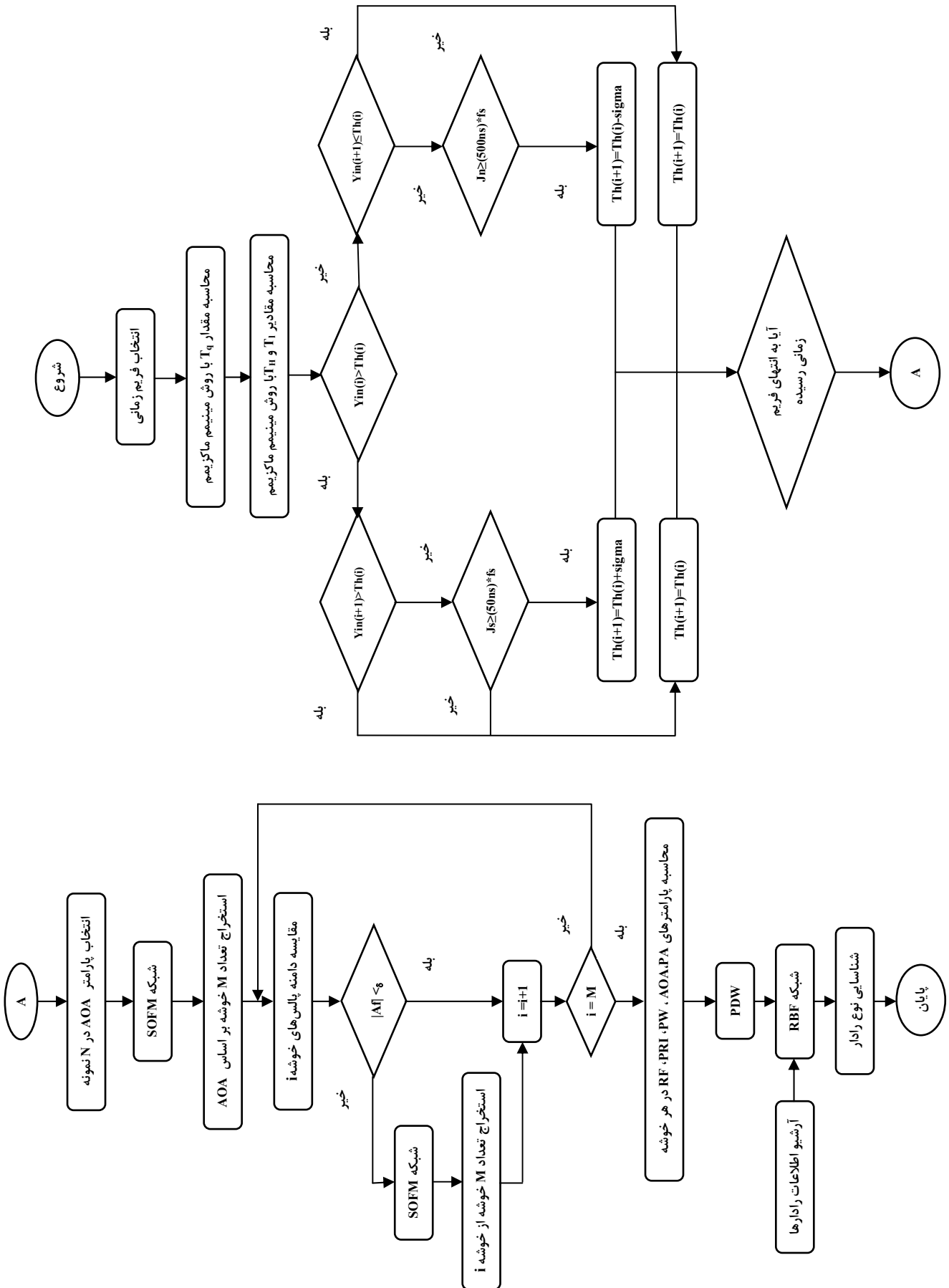
$$PRF : 100Hz \sim 1MHz$$

به عبارت دیگر $PW > 250 \mu s$ بعنوان سیگنال موج پیوسته شناخته شده و دیگر پالسی نیست. در نتیجه عرض پنجره اعمال سطح آستانه باید از 250 µs بزرگتر باشد که در اینجا این عرض 300 µs لحاظ شده است. با توجه به این موضوع همانطور که از شکل ۵ مشخص است بعد از اتمام هر پنجره بطور اختیاری سطح آستانه به مقدار اولیه بر می گردد. (البته این امکان نیز وجود دارد که مقدار ابتدایی سطح آستانه هر پنجره برابر مقدار انتهایی سطح آستانه پنجره قبل انتخاب شود ولی به هر حال در صورت انتخاب مناسب سطح آستانه اولیه تفاوت عملکردی بین این دو وجود ندارد.) با استفاده از این شبیه سازی ها عملکرد دو روش به ازای ورودی های ثابت، استگر، متداخل و موج پیوسته و تغییر مشخصه های هر ورودی (PRF، PW و PA) مورد ارزیابی قرار گرفت.

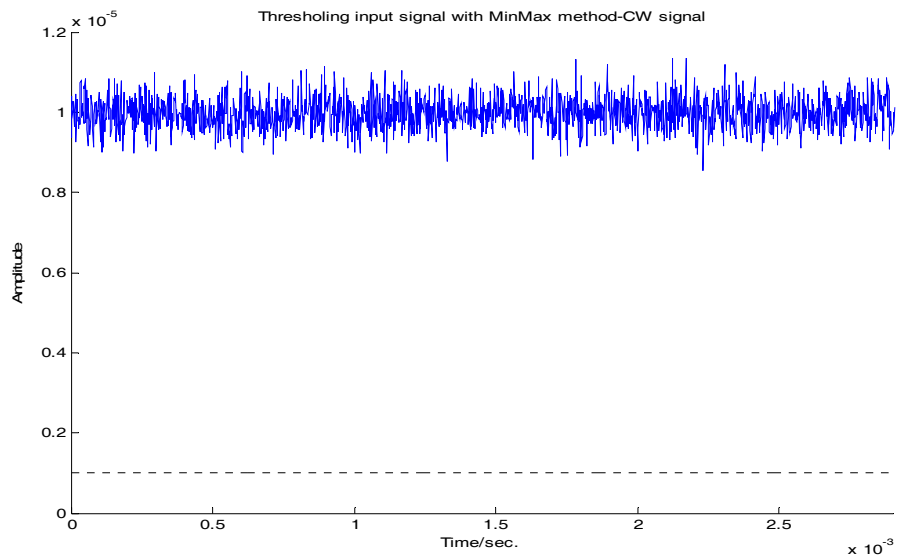
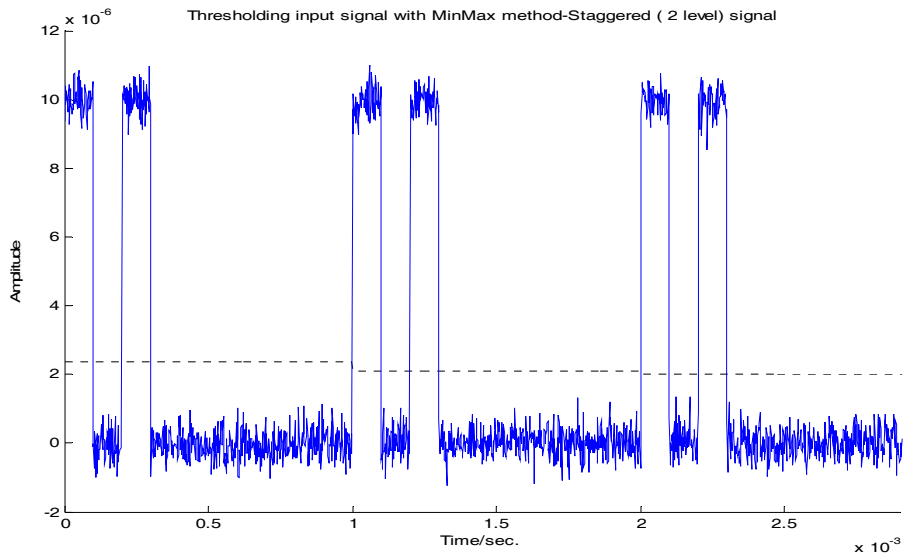
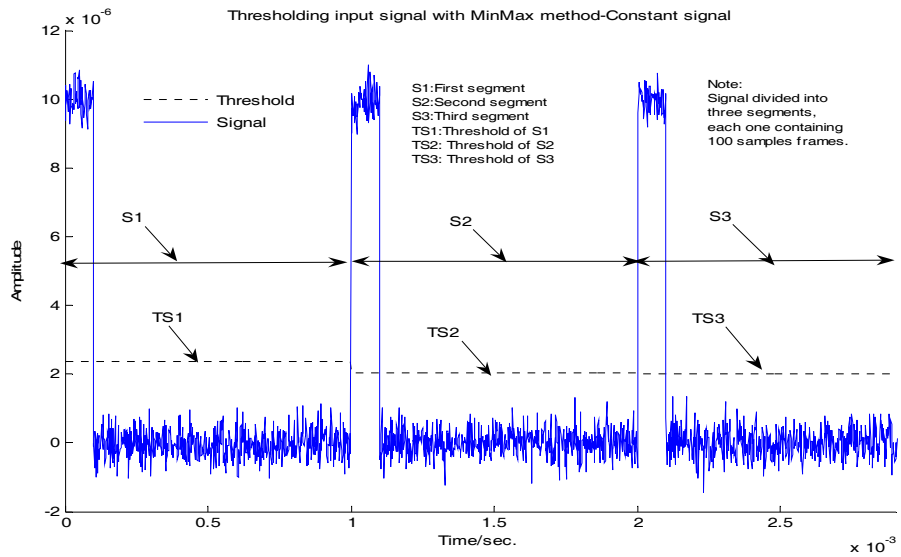
ملاک ارزیابی برآورد احتمال آشکارسازی و احتمال خطای سیگنال (که متناسب با FAR است) بوده است. برای مقایسه روش پیشنهادی با روش مینیمم ماکزیمم بررسی های آماری توسط سیستم شبیه ساز مطلب روی محیط های راداری مختلف انجام شد که نتایج حاصله در جدول ۲ آمده است. با توجه به بررسی های صورت گرفته روی رادارهای موجود عملی احتمال آشکارسازی ۹۹ درصد و احتمال خطا 10^{-6} مدنظر قرار گرفته است. لازم به ذکر است پیش فرض های مورد استفاده در شبیه سازی دو روش در جدول ۳ بیان گردیده است.

نتایج مذکور نشان می دهد که روش پیشنهادی به ازای ورودی های پالسی (ثابت، استگر و متداخل) و موج پیوسته و نیز تغییر وسیع مشخصه های هر ورودی (PRF، PW و PA) دارای عملکرد بسیار خوبی است بطوریکه در محدوده های ذکر شده در جدول ۲ احتمال آشکارسازی بیشتر از ۹۹ درصد و احتمال خطا در حدود 10^{-6} حاصل شده است. (البته محدوده های ذکر شده تقریبی است بطوریکه مثلاً در رشته پالس با فاصله تکرار پالس ثابت در $PA: -5dBm \sim +5dBm$ ، $PW=50ns$ ، $PRF=100Hz$ ؛ در رشته پالس با فاصله تکرار پالس ترکیبی در $PA: -5dBm \sim +5dBm$ ، $PW=100\mu s$ ، $PRF=100Hz$ می تواند باشد.)

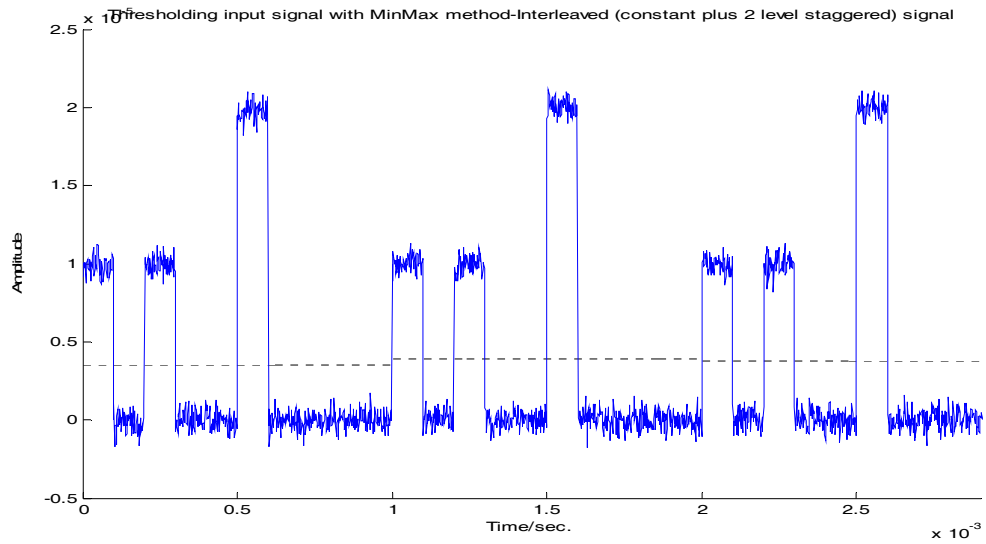
روش مینیمم ماکزیمم با استناد به نتایج جدول ۲ برای نرخ تکرار پالس بزرگتر از ۱۰۰ کیلو هرتز مناسب نیست که این مسئله پیاده سازی این روش را در سیستم های عملی مشکل ساز می نماید. ضمناً در روش مینیمم ماکزیمم انتخاب سطح آستانه اولیه بصورت دستی انجام می شود که عملیاتی بودن آن را ناممکن می سازد در صورتی که در روش پیشنهادی این مشکلات بر طرف شده است.



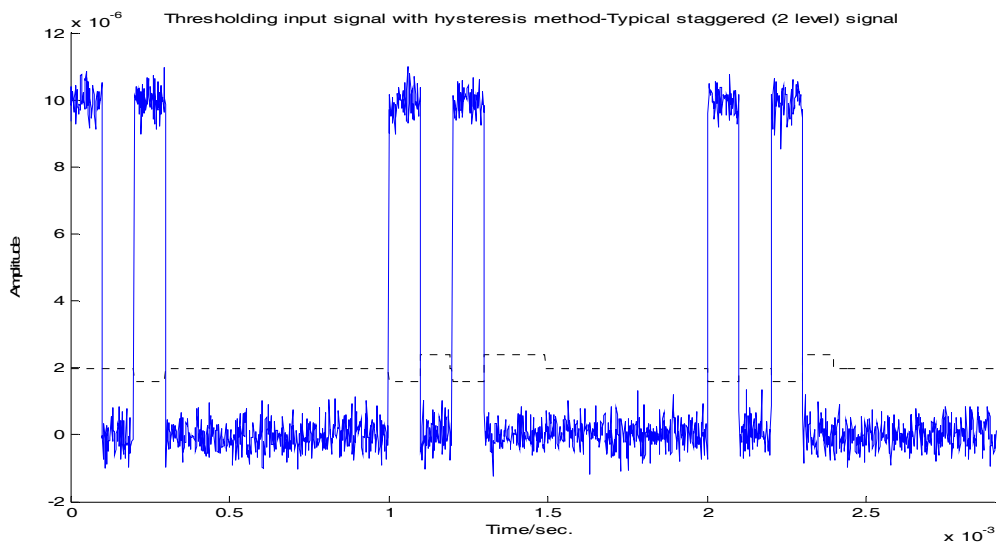
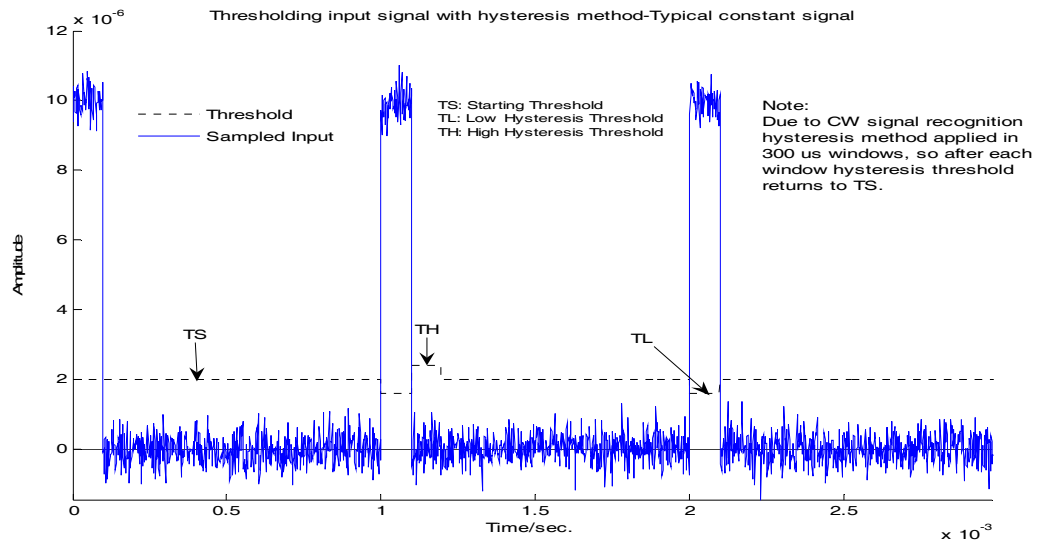
شکل ۵- فلوچارت روش پیشنهادی



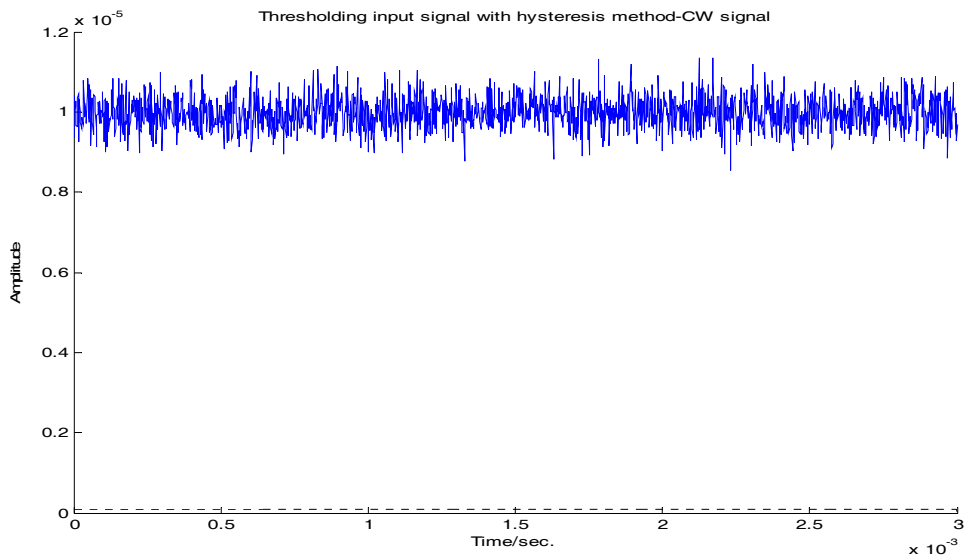
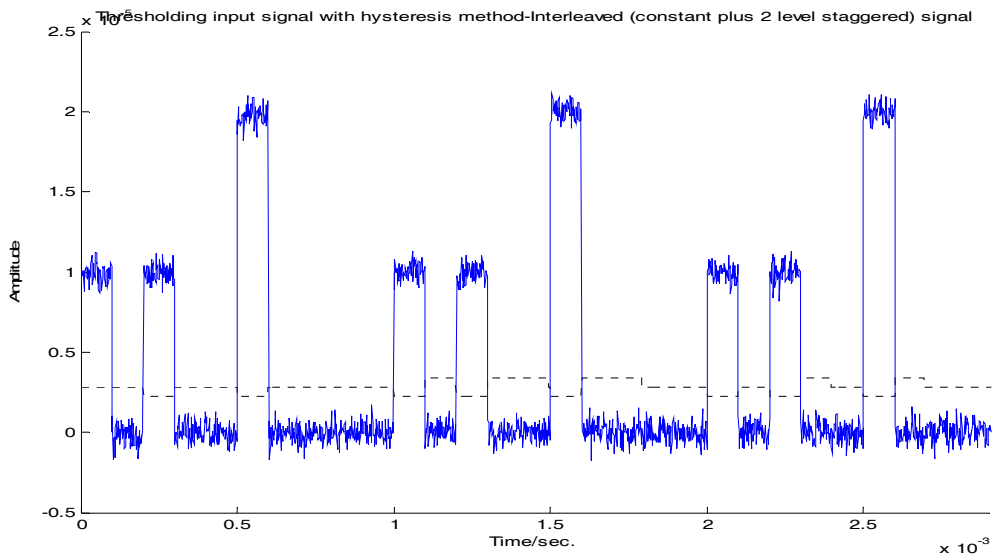
شکل ۶- نحوه عملکرد روش مینیمم ماکزیمم در تخمین سطوح آستانه چهار محیط مختلف راداری



شکل ۶- (ادامه)



شکل ۷- نحوه عملکرد (آشکارسازی پالس‌ها از سیگنال دریافتی) الگوریتم پیشنهادی برای چهار محیط مختلف راداری



شکل ۷- (ادامه)

جدول ۲- نتایج حاصله از روش‌های پیشنهادی و مینیمم ماکزیمم

Pd>0.99 & Pf ~ 10 ⁻⁶			
روش پیشنهادی		روش مینیمم ماکزیمم	
Pulse Signals	CW Signal	Pulse Signals	CW Signal
<ul style="list-style-type: none"> PRF:100Hz~1MHz • PW : 50ns ~ 250μs • A :-50dB ~ +5dBm • 	<ul style="list-style-type: none"> PA : - • 56dBm~+5dBm 	<ul style="list-style-type: none"> PRF:100Hz~100KHz • PW : 50ns ~ 250μs • PA :-50dBm ~ +5dBm • 	<ul style="list-style-type: none"> PA : - • 59dBm~+5dBm

جدول ۳- فرضیات مورد استفاده در شبیه‌سازی عملکرد روش‌های پیشنهادی و مینی‌م ماکزیمم

Hystersis	Min-Max
Signal Duration = 3ms Signal Frequency = 60MHz Gaussian Noise Level(GNL) = -68dBm Starting Threshold = 14dB above GNL Pulse Density = 1%	Signal Duration = 3ms Signal Frequency = 60MHz Gaussian Noise Level(GNL) = -68dBm Frame Length = 400 Samples Segment Length = 3ms Addaptive Threshold = 8dB above Min-Max of Segment Pulse Density = 1%

ب) بخش جداسازی و شناسایی الگوریتم پیشنهادی

آن بستگی دارد. نتایج بررسی‌های آماری صورت گرفته از ۵۰۰ بار تکرار روی تعداد رادارهای مختلف از ۲ تا ۲۰ نشان می‌دهد برای نویز و پالس گمشده ۵٪ این بخش دقتی معادل ۹۸٪ دارد.

در بخش شناسایی ابتدا با رادارهای آرشیو شبکه آموزش داده می‌شود که بعنوان مثال نتایج آموزش آن برای رادارهای جدول ۴ در شکل ۱۰ نشان داده شده است. سپس خروجی بخش جداسازی که شامل یکسری خوشه از پالس‌ها می‌باشد به شبکه RBF اعمال می‌گردد و نتایج آن مورد بررسی قرار می‌گیرد. بصورتیکه به هر خوشه‌ای که نزدیکتر باشد آن رادار بعنوان رادار مورد نظر اعلام می‌گردد.

برای نشان دادن این موضوع رشته پالس ۴ رادار از بین رادارهای جدول ۴ با تغییرات جزئی در پارامترهای RF و PW جدول ۵ به ورودی شبکه عصبی RBF اعمال شد. نتایج حاصله در شکل ۱۱ نشان داده شده است. نقاط با نام TR نقاط حاصل از ۴ رادار جدید است. همانطور که در شکل ۱۱ دیده می‌شود نقاط حاصل از تست (TR) نزدیک نقاط حاصل از رادارهای ۲، ۶، ۸ و ۱۸ می‌باشد پس این رادارها اعلام می‌گردند. نتایج بررسی‌های آماری صورت گرفته از ۱۰۰۰ بار اعمال تغییرات جزئی در پارامترهای مختلف رادارهای جدول ۴ نشان داد که این بخش از الگوریتم دقتی حدود ۹۹٪ دارد.

برای ارزیابی عملکرد این بخش یک سیمولاتور نرم‌افزاری تولید رشته پالس‌های متداخل راداری طراحی گردید بصورتیکه قادر است تعداد رادارها و مشخصات آنها را دریافت نموده و رشته پالس متداخل آنها را تولید نماید. با این سیمولاتور، رشته پالس متداخل ۲۰ نوع رادار عملی (رادارهای جدول ۴) را تولید نموده و سپس رشته پالس متداخل در یک فریم زمانی 200ms جهت ارزیابی این بخش از الگوریتم استفاده گردید. پترن رشته پالس متداخل تولید شده توسط سیمولاتور برای رادارهای جدول ۴ در شکل ۸ نشان داده شده است.

جدول ۴- مشخصات ۲۰ رادار عملی تولید شده توسط سیمولاتور

نوع رادار	Power (Kw)	PRF (Hz)	PW (μs)	RF (MHz)
رادار ۱	۲۱۰	۵۰۰	۲	۱۰۰۰
رادار ۲	۵۰۰	۳۰۰	۴	۱۵۰۰
رادار ۳	۲۲۰	۶۰۰	۳	۲۰۰۰
رادار ۴	۶۰۰	۱۵۰۰	۱۲	۳۰۰۰
رادار ۵	۲۰۰	۸۵۰	۲۰	۳۵۰۰
رادار ۶	۶۵۰	۷۰۰	۱	۳۰۰
رادار ۷	۱۵۰	۹۰۰	۱۰۰	۲۸۰۰
رادار ۸	۱۰	۲۵۰	۳۶	۴۰۰۰
رادار ۹	۸۰۰	۵۰۰	۳۳	۵۰۰۰
رادار ۱۰	۱۵۰۰	۲۸۸۰	۱۲	۵۱۵۰
رادار ۱۱	۸۰۰	۵۰۰	۳	۸۰۰۰
رادار ۱۲	۱۲	۱۰۰۰	۱۲	۱۰۱۰۰
رادار ۱۳	۱۰	۵	۰/۴	۲۰۰۰۰
رادار ۱۴	۴۰۰	۳۰۰	۰/۰۲	۲۰۰۰۰
رادار ۱۵	۷	۵۰۰۰	۱۰	۱۸۰۰۰
رادار ۱۶	۱۵	۲۴۰۰	۰/۱۴	۳۳۰۰
رادار ۱۷	۱۵	۶۰۰۰	۰/۲	۳۳۰۰
رادار ۱۸	۱۰۰	۳۰۰	۰/۰۲	۱۸۰۰۰
رادار ۱۹	۴۰۰	۳۰۰	۰/۰۲	۱۵۰۰
رادار ۲۰	۸۰۰	۳۶۵	۳	۴۵۰۰

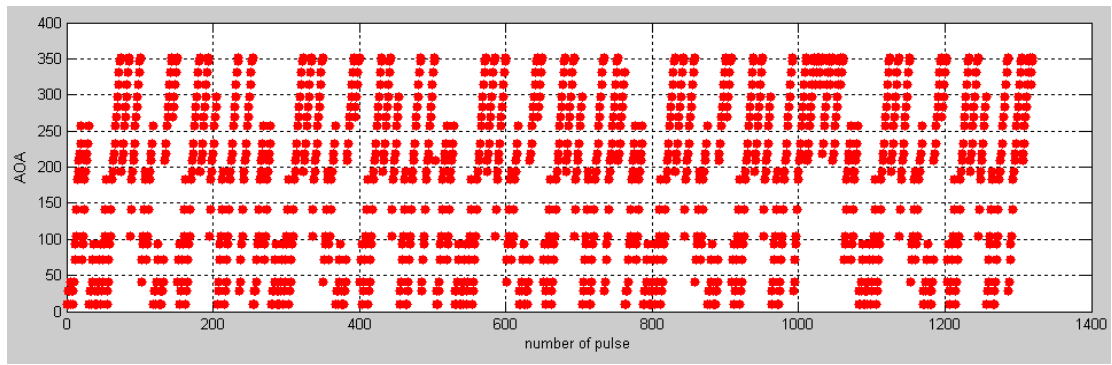
جدول ۵- تغییرات اعمالی برای ۴ رادار عملی

نوع رادار	PW(μs)	RF(MHZ)
رادار ۲	۳/۹۲	۱۴۹۰
رادار ۶	۱/۰۲	۳۰۱۰
رادار ۸	۳۵/۵	۳۹۹۲
رادار ۱۸	۳۰۵	۱۸۰۰۸

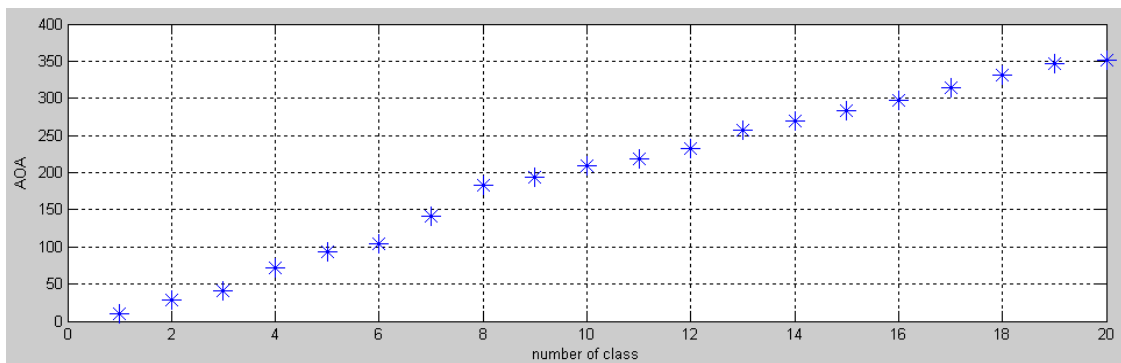
۶- نتیجه‌گیری

نتایج بررسی‌های صورت گرفته نشان می‌دهد روش پیشنهادی در محیط‌های با چگالی پالس بالا عملکرد مناسبی دارد. آشکارسازی پالس‌ها در روش پیشنهادی هیچگونه وابستگی به پارامترهای فاصله تکرار پالس، پهنای پالس، زاویه ورود پالس، زمان ورود پالس و فرکانس حامل پالس ندارد بنابراین قادر به آشکارسازی پالس‌های منتشره از رادارهای با تکنیک‌های مختلف (رادار با فاصله تکرار پالس ثابت، رادار با فاصله تکرار پالس stagger، رادار با فاصله تکرار پالس jitter، رادار با فاصله تکرار پالس sliding، رادار با فاصله تکرار پالس dwell and switch، رادار با فاصله تکرار پالس periodic، رادار با فاصله تکرار پالس scheduled، رادار با قابلیت ارسال pulse group، رادار با پرش فرکانسی و رادار با فرکانس حامل دو تایی) می‌باشد.

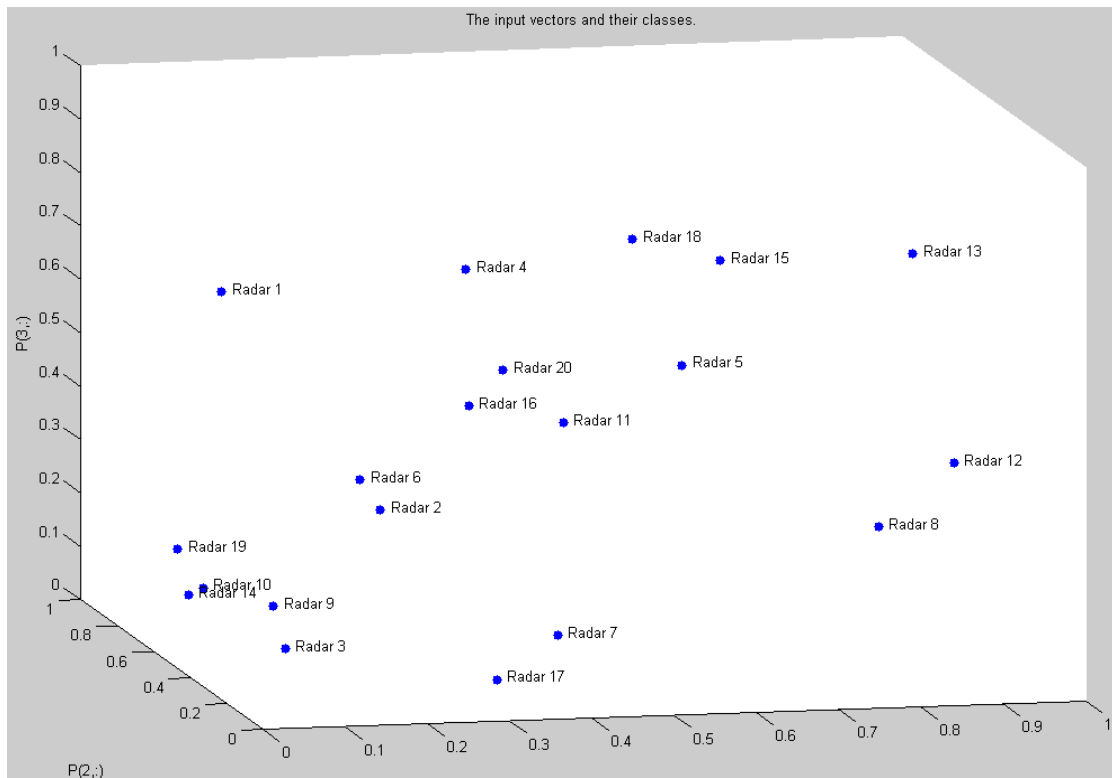
پس از اعمال رشته پالس متداخل به شبکه عصبی SOFM طبق الگوریتم پیشنهادی عمل جداسازی انجام می‌گیرد. نتیجه حاصله بخش جداسازی در شکل ۹ آمده است. همانطور که در این شکل دیده می‌شود تعداد خوشه‌های حاصله ۲۰ خوشه می‌باشد. دقت جداسازی خوشه‌ها توسط این شبکه به توپولوژی



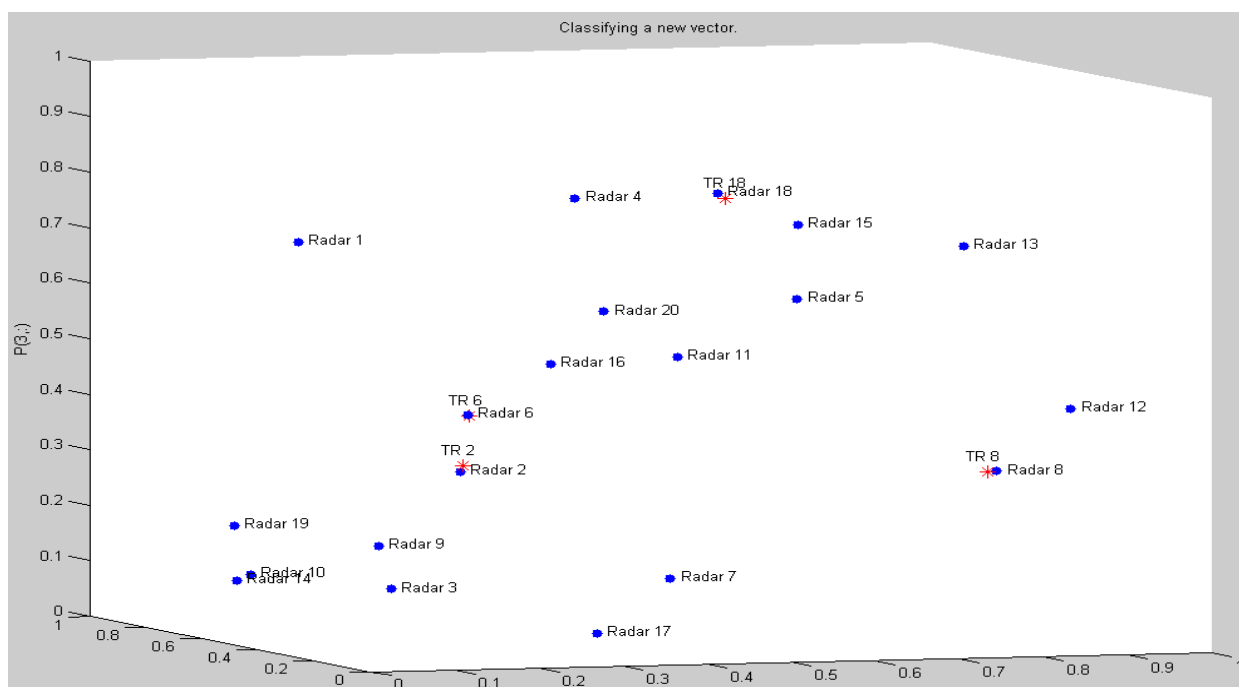
شکل ۸- خروجی سیمولاتور شامل رشته پالس متداخل ۲۰ رادار مختلف



شکل ۹- خروجی بخش جداسازی الگوریتم پیشنهادی برای رشته پالس متداخل ۲۰ رادار مختلف



شکل ۱۰- خروجی شبکه RBF بعد از آموزش ۲۰ رادار عملی (جدول ۴)



شکل ۱۱- خروجی بخش شناسایی الگوریتم پیشنهادی برای ۴ خوشه جداسازی شده

[4] C. J. Kim, and H. S. Lee, "Analysis of the Generalized Order Statistics Constant False Alarm Rate Detector," ETRI Journal, vol. 16, no. 1, Apr. 1994.

[5] M. I. Skolnik, "Radar Handbook," Second Edition, 1980.

[6] S. H. Moon, D. S. Han, M. J. Cho, and K.-H. Park, "Frequency Domain GSC with Low Computational Complexity," 1999 IEEE.

[7] B. Chen, P. K. Varshney, and J. H. Michels, "Adaptive CFAR Detection for Clutter-Edge Heterogeneity Using Bayesian Inference," IEEE Trans. On Aerospace and Electronic systems, vol. 39, no. 4, Oct. 2003.

[8] V. P. Behar, C. A. Kabakchiev, and L. A. Doukovska, "Adaptive CFAR PI Processor for Radar Target Detection in Pulse Jamming," Journal of VLSI Signal Processing, pp. 383-396, 2000.

[۹] م. ع. خلیقی، "طراحی یک گیرنده با نرخ هشدار کاذب ثابت به منظور کاربرد در جنگ الکترونیک،" پایان‌نامه کارشناسی ارشد، دانشکده برق دانشگاه صنعتی شریف، ۱۳۷۶.

[10] R. G. Wiley, "Electronic Intelligence: The Analysis of Radar Signals," 2nd Edition, Artech House, Inc., 1993.

[11] M. A. Khalighi, and M. M. Nayebi, "CFAR Adaptive Threshold for ESM Receiver with Logarithmic Amplification," Eurasip Signal Processing Journal, vol. 84, pp. 41-53, Jan. 2004.

این روش علاوه بر رادارهای پالسی قادر به تشخیص رادارهای با سیگنال موج پیوسته می‌باشد. بدین صورت که در این روش عرض پالس‌های بزرگتر از $250\mu s$ بعنوان سیگنال موج پیوسته شناخته می‌شود. تا بتوان سیگنال‌های موج پیوسته را از سیگنال‌های پالسی تفکیک کرد و برای آن سطح آستانه مجزایی (سطح آستانه سیگنال موج پیوسته) تعریف کرد. در روش مذکور سطح آستانه آغازین بصورت اتوماتیک از سیگنال ویدئو دریافتی تخمین زده می‌شود حال آنکه در روش‌های قبلی این مقدار بصورت دستی اعمال می‌گردید. نتایج بررسی‌های آماری صورت گرفته نشان می‌دهد بخش جداسازی دقتی معادل ۹۸٪ و بخش شناسایی دقتی معادل ۹۹٪ دارد و نتایج پیاده‌سازی آن و فعالیت سیستم در منطقه خلیج فارس نشان می‌دهد کل الگوریتم دقتی معادل ۹۷/۰۲٪ دارد که نسبت به سیستم‌های عملی مشابه خارجی (حداکثر دقت ۹۳/۵٪) بسیار بهتر می‌باشد. ضمناً روش پیشنهادی نسبت به دیگر روش‌ها دلیل استفاده از شبکه عصبی دارای پیچیدگی محاسباتی کمتری می‌باشد. ضمناً تحمل‌پذیری خطای بیشتری دارد

مراجع

[۱] م. ا. ضیاء، "کاربرد سیستم‌های هوشمند در تحلیل و تشخیص سیگنال‌های راداری،" پایان‌نامه کارشناسی ارشد، دانشکده برق دانشگاه علم و صنعت، خرداد ۱۳۷۴.

[2] M. Kiick, "Constant False Alarm Rate Detection of Radar Signals with Artificial Neural Networks," M.Sc.Thesis, University of Skovde Oct. 1996.

[3] S. Watts, "The Performance of Cell-Averaging CFAR System in Sea Clutter," Racial Defence Electronics.

- [12] S. E. Lipsky, "Microwave Passive Direction Finding," John Wiley & Sons, 1987.
- [13] S. George, "W: Introduction to Airborne Radar-Second Edition," 1998.
- [14] H. K. Mardia, "New Techniques for the Deinterleaving of Repetitive Sequences," IEEE 1989.
- [15] C. Zhao, Y. Zhao, and J. Lu, "Radar Signals Sorting with Kohonen Neural Network," IEEE 2006.
- [16] J. Perkins, "Pulse Train Deinterleaving via the Hough Transform," IEEE 1994.
- [17] N. J. Whittall, "Signal Sorting in ESM Systems," IEEE 1985.
- [18] E. Savaria, and P. Lavoie "A Comparison of Self-Organizing Neural Networks for Fast Clustering of Radar Pulses," 1998.
- [19] C. Dayid, W. A. Jams, and P. Thompson, "An Adaptive Data Sorter Based On Probabilistic Neural Networks," IEEE 1991.
- [20] E. Hossam, A. B. Hassan, F. Chan, and Y. T. Chan, "Queuing Analysis Of The Deinterleaving Of Radar Pulses In A Dense Emite Environment," IEEE 2003.
- [21] E. Hossam, A. B. Hassan, F. Chan, and Y. T. Chan, "Joint Deinterleaving And Recognition Of Radar Pulses," IEEE 2003.
- [22] J. A. Anderson, and M. T. ATELY, "Radar Signal Categorization Using A Neural Network," IEEE 1990.



علی ناصری استادیار دانشکده فناوری اطلاعات و ارتباطات و رییس گروه فناوری اطلاعات و ارتباطات دانشگاه جامع امام حسین (ع) است. وی بیش از بیست سال است در زمینه پردازش سیگنال‌های رادار مشغول تحقیق و پژوهش است. دکتر ناصری مشاور سازمان فناوری اطلاعات نیز می‌باشد.
آدرس پست‌الکترونیکی ایشان عبارت است از:

a.naseri@ict.gov.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۹/۱۱/۱۲

تاریخ اصلاح: ۹۱/۶/۱۱

تاریخ قبول شدن: ۹۱/۸/۱۶

نویسنده مرتبط: دکتر علی ناصری، دانشکده فناوری ارتباطات و اطلاعات، دانشگاه جامع امام حسین (ع)، تهران، ایران.

ارزیابی معیارهای کارآیی در مدیریت پردازش شبکه‌ای بوسیله اشیای سیار

محمود صادقی

محمد کاظم اکبری

سید محمد رضا میرزا بابائی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

چکیده

رویکرد تمام کاربردهای مدیریت شبکه به سمت استفاده از عامل‌های هوشمند سیار می‌باشد تا بتوانند معماری‌های سنتی را به سمت محیط پردازش فراگیر ارتقا دهند. خصوصیات نظیر مقیاس‌پذیری، قابلیت اطمینان، قابلیت دسترسی، امنیت، کارآیی و ... نقش مهمی در مدیریت شبکه خصوصاً زمانبندی پردازش شبکه‌ای دارند. مدیریت بسیاری از این خصوصیات را می‌توان با استفاده از عامل‌های سیار بهینه نمود. با توجه به فراگیری سیستم‌های مدیریت شبکه مبتنی بر دلایل‌های درخواست شیء (ORB) و نیز تجاری بودن این میان‌افزارها مانند CORBA، می‌توان از آنها برای بدست آوردن معیارهای کارآیی و سپس زمانبندی با استفاده از آن اقدام نمود. در این پروژه نمونه بسیار کوچکی از قابلیت‌های (قابلیت تحرک براساس استاندارد MASIF) عامل‌های سیار هوشمند به این سیستم اضافه شده است که امکان استفاده از سرویس‌دهنده‌های نامگذاری محلی را فراهم می‌نماید. نتایج مشاهده شده، حاکی از بهبود کارآیی سیستم مدیریت زمانبندی شبکه پیاده‌سازی شده می‌باشد. منظور از کارآیی در این تحقیق، مدت زمان پاسخ و ترافیک عبوری در شبکه است.

کلمات کلیدی: محیط فراگیر، کوربا CORBA، مدیریت شبکه توزیع شده، عامل‌های سیار، ارزیابی کارآیی.

۱- مقدمه

اینگونه سیستم‌های مدیریتی محدودیت‌های جدی را از نظر کارآیی، مقیاس‌پذیری و نیز انعطاف‌پذیری دارا می‌باشند. برای مدیریت ابزارها و خدمات شبکه‌ای در چنین محیط‌های با پیچیدگی بالا، سازمان‌ها و موسسات مختلفی سکوها مدیریتی گوناگونی (نظیر Internet Management Framework، OSI، Management Framework، TMN [۲]) را با استفاده از انواع مختلف پروتکل‌های ارتباطی مدیریتی (همچون SNMP [۳]، CMIP [۴]) توسعه داده و آماده ساخته‌اند.

در حال حاضر، بسیاری از شبکه‌های داده توسط SNMP و بسیاری از شبکه‌های مخابراتی توسط CMIP مدیریت می‌شوند که هر دو از انواع پیاده‌سازی‌های مشتری/خدمتگذار می‌باشند. علاوه بر این دو روش فوق، CORBA [۵] که یک معماری توسعه‌یافته توسط OMG می‌باشد، چارچوبی را برای تعاملات بین اشیاء در محیط‌های توزیع‌شده فراهم می‌سازد. این چارچوب نیز راه حل همه منظوره‌ای که مبتنی بر مدل مشتری/خدمتگذار می‌باشد را برای توسعه برنامه‌های کاربردی مدیریت شبکه فراهم می‌سازد که محدودیت خاصی را برای پروتکل ارتباطی تحمیل نمی‌کند. سیستم‌های مدیریت شبکه متمرکز^۱ به نوبه خود

اساساً مدیریت و زمانبندی پردازش در شبکه برای تحقق اهداف مدیریتی نیازمند نظارت و کنترل تجهیزات متصل به شبکه با استفاده از جمع‌آوری و تحلیل اطلاعات مختلف مدیریتی از تجهیزات شبکه می‌باشد [۱]. این ویژگی این امکان را برای مدیران شبکه فراهم می‌سازد تا بتوانند از یک نقطه‌ی شبکه، کل شبکه را تحت مدیریت خود داشته باشند. بسیاری از سیستم‌های مدیریتی از مدل مدیریت ایستا و متمرکز پیروی می‌کنند که موجب انتقال حجم عظیمی از اطلاعات خام از سیستم‌های راه دور به ایستگاه مرکزی مدیریت شبکه می‌گردد قبل از آنکه پردازش، خلاصه‌سازی و ساده‌سازی بر روی آنها انجام گیرد. در نتیجه این سیستم‌ها، موجب ازدحام در شبکه می‌شوند که منجر به مشکلاتی نظیر تأخیر پاسخ‌ها، کاهش کارآیی، عدم واکنش سریع و به موقع در هنگام رخداد آسیب‌های شبکه‌ای و همچنین پهنای باند بسیار پایینی می‌گردد.

بنابراین، با روند رشد نمایی که در اندازه، گستردگی و توزیع، پیچیدگی و نیز گوناگونی شبکه‌های ارتباطی و همچنین منابع و تجهیزات شبکه‌ای وجود دارد،

شیوه‌های مدیریت شبکه که در اینجا مورد بحث قرار می‌گیرد به چهار دسته تقسیم می‌گردد (مطابق شکل ۱):

۱) **مدل مشتری/خدمتگزار (CS)**، که همان شیوه سنتی مدیریت شبکه می‌باشد که در آن برای جمع‌آوری اطلاعات و کنترل تجهیزات، یک شی سرویس‌دهنده مقیم در عنصر مدیریت شونده تحت عنوان شی مدیریت‌شونده قرار دارد و به درخواست‌های مدیر پاسخ می‌دهد (شکل ۱-۱).

۲) **مدل مدیریت توزیعی شبکه - سلسله مراتبی ایستا (DNM-HS)**، که همانگونه که در شکل نیز مشاهده می‌شود، شامل سلسله‌مراتبی از مدیران است که زیرشبکه‌های خود را مدیریت می‌کنند و خود آنها نیز توسط مدیران سطح بالاتر مدیریت می‌گردند. هر کدام از زیرشبکه‌ها براساس مدل مشتری/خدمتگزار مدیریت می‌گردند (شکل ۱-۲).

۳) **مدل مدیریت توزیعی شبکه - سیار ضعیف (DNM-WM)**، که از روش توزیع کد و انتقال آن به هر عنصر شبکه و اجرای آن توسط عنصر استفاده می‌کند. در این روش کد مدیریتی از مدیر به عنصر شبکه انتقال پیدا کرده و توسط وی اجرا می‌گردد و نتیجه به مدیر برگردانده می‌شود. نهایتاً کد نیز در عنصر شبکه منقذ می‌شود. در این روش کد انتقال داده شده قابلیت مهاجرت به عناصر دیگر شبکه را ندارد (شکل ۱-۳).

۴) **مدل مدیریت توزیعی شبکه - سیار قوی (DNM-SM)**، که با استفاده از گسیل تعدادی عامل سیار توسط سیستم مدیریت به سطح شبکه، وظیفه مدیریتی خود را به انجام می‌رساند. در این روش وظیفه مدیریتی با همکاری تعدادی عامل سیار انجام می‌گیرد به این ترتیب که آنها با سفر به عناصر مختلف شبکه براساس برنامه سفر از پیش تعیین شده یا مسیریابی پویا این عملیات را کامل نموده و وظیفه مدیریت را به انجام می‌رسانند (شکل ۱-۴).

در کل، معیارهای کارایی عبارتند از: مدت زمان تکمیل یک کار و همچنین بهره‌وری پهنای باند شبکه. در این مقاله نیز همین معیارها برای ارزیابی کارایی مورد اندازه‌گیری قرار گرفته است.

سرویس‌گیرنده‌های عامل‌های مدیریتی هستند که در هریک از عناصر مدیریت‌شونده شبکه^۲، مقیم می‌باشند. با این وجود و با مطالعه نمونه‌های بسیاری از برنامه‌های کاربردی مدیریت شبکه که عملاً مورد استفاده قرار گرفته است، نقایص و محدودیت‌های مدل مشتری/خدمتگزار - به طور مثال، پردازش حجیم و محدودیت ترافیک در سیستم‌های مدیریت شبکه - سال‌هاست که مشخص و تعیین شده است.

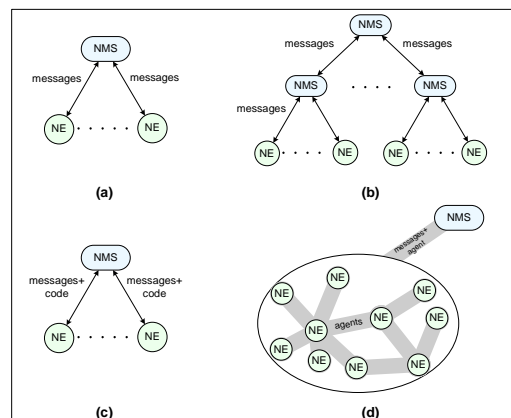
توزیع نمودن کارکردهای مدیریت شبکه ممکن است منافعی را به دنبال داشته باشد که از آن جمله می‌توان به موارد زیر اشاره نمود: تقلیل عمده حجم پردازش و ترافیک را در مدیریت شبکه به گونه‌ای که مقداری از پردازش بر عهده عناصر شبکه قرار بگیرد؛ افزایش قابلیت مقیاس‌پذیری در شبکه‌های بزرگ؛ استخراج اطلاعات در مکان‌هایی نزدیک به منابع داده جهت افزایش سرعت و کارایی؛ همچنین پایداری بالاتر به دلیل مدیریت توزیع‌شده شبکه عدم نیاز به اتصال دائمی و مداوم سیستم مدیریت شبکه به عناصر شبکه. لذا با قطع ارتباط نیز می‌توان برخی از کارکردهای مدیریتی را پیاده‌سازی نمود.

چارچوب کوربا بر مدل مشتری/خدمتگزار بنا شده است به طوری که در روش‌های سنتی کاربردهای مدیریت شبکه، اشیاء در عناصر شبکه به صورت دائمی مقیم هستند. کاربری و نیز مفید بودن استفاده از فناوری‌های عامل‌های سیار^۳ در مدیریت سیستم‌ها و شبکه‌های توزیع‌شده چندین سال است که به اثبات رسیده است [۶]. آنچه که در این مقاله به آن پرداخته شده است، افزودن ویژگی سیار بودن عامل‌های سیار، به اشیاء ایستای کوربا در محیط‌هایی است که این چارچوب به عنوان زیرساخت مدیریتی استفاده می‌گردد، تا بتوان به کارایی بهتری دست یافت. برای این منظور، یک محیط آزمایشی بر اساس چارچوب ارتباطی MAF برای اجرای اشیاء سیار آماده گردید و اندازه‌گیری‌های لازم صورت پذیرفت. ادامه مقاله از بخش‌های زیر تشکیل یافته است: بخش ۲ یک مرور کلی است بر مدیریت توزیعی شبکه، چارچوب کوربا و قابلیت سیار بودن اشیاء. بخش ۳ نمونه پیاده‌سازی شده را توضیح می‌دهد. بخش ۴ شبیه‌سازی و نتایج حاصله را بیان می‌دارد. و در نهایت نکات نتیجه‌گیری ارائه شده است.

۲- پیش‌زمینه

۲-۱- مروری بر مدیریت توزیعی شبکه

مدیریت متمرکز شبکه، مدیریتی با کارایی بسیار پایین می‌باشد چراکه تمامی اعمال مدیریتی به یک مدیریت یکپارچه وابسته است. به وضوح می‌توان دریافت که توزیع کارکردهای مدیریت شبکه به سلسله‌مراتبی از مدیران میانی که مسئول مدیریت بخشی از شبکه می‌باشند، باعث افزایش کارایی می‌گردد [۷].



شکل ۱- شیوه‌های مختلف مدیریت شبکه

۲-۲- CORBA و قابلیت سیار بودن

چارچوب کاری کوربا این امکان را برای برنامه‌نویسان سیستم‌های توزیعی فراهم می‌نماید تا آنها بتوانند بدون در نظر گرفتن زبان برنامه‌سازی و همچنین بدون درگیر شدن با لایه‌های پایین شبکه و نیز تنوع سیستم عامل، به فراخوانی متدهای اشیاء چارچوب به شیوه‌ای شی‌گرا بپردازند. هسته این چارچوب یک دلال درخواست است که به عنوان یک میان‌افزار توزیع اطلاعات، ارتباط و همکاری بین اشیاء را به صورتی یکپارچه فراهم می‌سازد. ارتباطات اشیاء کوربا از مدل مشتری/خدمتگزار پیروی می‌کند که مبتنی بر مکانیزم درخواست-پاسخ^۴ می‌باشد. برای آنکه یک عملیات مشخص صورت گیرد، سرویس‌گیرنده باید یک متد مشخص شی راه دور را فراخوانی نموده و ارتباط خود را با سرویس‌دهنده حفظ نماید تا انتقال اطلاعات صورت گیرد.

چارچوب کوربا مستقل از سکو است و لذا می‌تواند انواع مختلف شبکه‌ها، ماشین‌ها و سیستم‌عامل‌ها را پوشش دهد و این خود منجر به تولید سیستم‌ها و برنامه‌های کاربردی مبتنی بر اشیاء توزیع‌شده با امنیت بالاتر و همچنین قابلیت استفاده دوباره نرم‌افزار می‌گردد. در چارچوب کوربا می‌توان سرویس‌های مختلفی را یافت که برای پیاده‌سازی سیستم‌های توزیعی مورد نیاز است؛ سرویس‌هایی برای محلی‌سازی نظیر: سرویس نامگذاری و Trader و ارتباطات رخدادگرا نظیر سرویس رخداد از آن جمله‌اند. در این معماری پایداری، تراکنش‌ها و امنیت نیز تبیین شده است. تمامی این مباحث مورد علاقه سیستم‌های مبتنی بر عامل‌های سیار نیز

را دریافت نمود؛ همچنین کلاس‌های آنها را انتقال داد. واسط MAFinder نیز برای ثبت نام و بازیابی عامل‌ها، مکان‌ها و سیستم‌های عامل اختصاص یافته است.

۳- پیاده‌سازی

بر اساس مطالب بیان شده در بخش‌های قبلی، در این پروژه، قابلیت حرکت عامل‌های سیار به اشیاء به چارچوب کوربا اضافه شده و شی سیار کوربا (MCO) با استفاده از MASIF پیاده‌سازی شده است.

۳-۱- آژانس‌ها

آژانس، یک محیط اجرایی برای عامل‌های سیار فراهم می‌کند که در واژگان MASIF تحت عنوان رابط سیستم عامل شناخته می‌شود. هر آژانس به یک ناحیه تعلق دارد و نام آن در آن ناحیه منحصر به فرد است؛ همچنین این آژانس متعلق به یک مالک^۵ می‌باشد. در این سیستم، هر آژانس عبارت است از یک شی کوربا در نقش سرویس دهنده، که واسط MFAgentSystem را پیاده‌سازی کرده است. عامل‌ها را می‌توان از طریق واسط MFAgentSystem و نیز متدهای کلاس Agency مدیریت نمود. عملیاتی نظیر: ایجاد و خاتمه یک عامل، تعلیق و از سرگیری فعالیت یک عامل، انتقال عامل، لیست‌گیری اسامی عامل‌های مقیم در یک رابط سیستم عامل، و دریافت اطلاعات عامل‌های محلی بر عهده یک آژانس می‌باشد.

۳-۲- اشیاء (عامل‌های) سیار

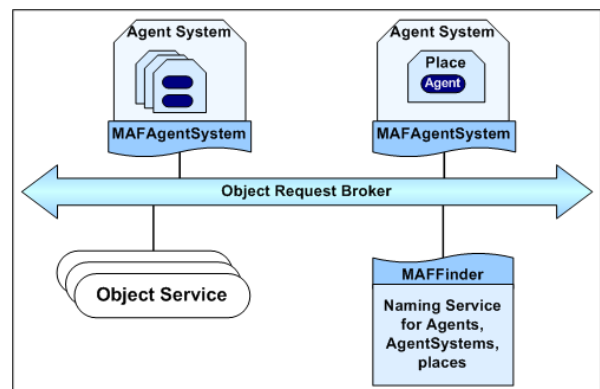
آژانس‌ها دارای توابعی برای ایجاد و مدیریت اشیاء کوربا هستند. این اشیاء، واسط MobileObject را که به صورت واسط کوربا یا IDL نوشته شده است، پیاده‌سازی کرده‌اند. این واسط دارای متدهایی برای مدیریت دوره زندگی عامل‌های سیار می‌باشد. در شکل زیر بخشی از این واسط دیده می‌شود.

طراحی واسط MobileObject بسیار آسان بوده و از تحلیل چارچوب MASIF به دست می‌آید. بر اساس این استاندارد عامل‌ها می‌توانند ایجاد شوند، حرکت کنند، معلق شوند، ادامه فعالیت دهند و یا خاتمه پیدا کنند. پس یک آژانس باید تمامی این اعمال را پشتیبانی نماید. از طرف دیگر، MCO باید از رخدادهایی که برای آن اتفاق می‌افتد باخبر شود تا بتواند واکنش مناسب نسبت به آن رخداد را از خود نشان دهد. علاوه بر آن یک MCO باید قابلیت پذیرش درخواست‌های انتقال و خاتمه از راه دور را داشته باشد؛ با اینکه این اعمال در اصل توسط آژانس نگهدارنده MCO انجام می‌گیرد ولی MCO باید متدهای لازم برای ارتباط کلاینت‌ها را برای انجام چنین اعمالی پشتیبانی کند. MCO باید از رخدادهای مختلف دوره زندگی خود از شروع تا موفقیت یا شکست باخبر شود تا نه تنها بتواند به مناسبترین صورت عمل نماید بلکه، قادر باشد مجوزهای مختلف چنین عملی را بیازماید (جدول زیر را مشاهده نمایید).

پیاده‌سازی‌های انجام شده برای MCO، علاوه بر واسط MobileObject باید قابلیت رشته‌سازی^۶ را نیز پشتیبانی کنند. رشته‌سازی این امکان را فراهم می‌سازد تا بتوان حالت شی سیار را به صورت یک رشته تولید کرد. هدف این تحقیق، پیاده‌سازی یک چارچوب با قابلیت تحرک در جهت افزایش کارایی عملیات مدیریتی مبتنی بر اشیاء می‌باشد. بنابراین به جای استفاده از سرویس برونی‌سازی کوربا که فایل‌های stub زیادی تولید می‌کند و نیز دارای سرعت کمتری است از امکان سریال‌سازی خود جاوا استفاده شده است. لذا MCOها باید واسط java.io.Serializable را نیز پیاده‌سازی کنند.

می‌باشد، و بدون آنکه آنها را درگیر زبان‌های برنامه‌نویسی بکند (چرا که ارتباطات اشیاء توسط زبان تعریف واسط یا همان IDL به زبان خاصی نگاشت می‌گردد)، در محیط‌های گسترده و بر اساس استانداردهای فراگیر قابل اجرا می‌سازد. پیاده‌سازی‌های انجام شده در کوربا، سیار بودن اشیاء را پشتیبانی نمی‌کنند ولی می‌توانند در سیستم‌های مبتنی بر عامل‌های سیار برای موارد خاصی استفاده شوند نظیر: محیط‌های اجرایی میزبان عامل‌ها، زیرساخت ارتباطی عامل‌ها، سرویس فهرست‌گیری و ...

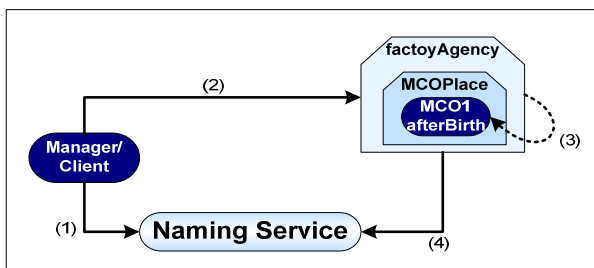
در این مقاله، ما از ویژگی‌ها و خصوصیات گفته شده در MASIF [۸] برای افزودن قابلیت سیار بودن عامل‌ها بهره گرفته‌ایم. این ویژگی‌ها که توسط OMG ارائه شده است، برای ایجاد قابلیت تعامل میان سکوها عامل‌ها و خود عامل‌ها می‌باشد. چارچوب MASIF بر اساس مفاهیم زیر می‌باشد: مالکیت یک عامل، شخص یا سازمانی را که آن عامل از طرف وی عمل می‌کند، مشخص می‌کند و این مالکیت در صورت حرکت عامل باید مورد تایید قرار بگیرد؛ هنگامی که عاملی خود را انتقال می‌دهد، این عامل، بین محیط‌های اجرایی‌ای که مکان‌ها خوانده می‌شوند، سفر می‌کند. مکان، زمینه‌ای در سیستم عاملی می‌باشد که عامل در آن اجرا می‌شود. این زمینه می‌تواند توابعی مانند کنترل دسترسی، را فراهم نماید. مکان مبدا و مقصد، می‌تواند در یک سیستم عامل یا در دو سیستم عاملی که شناسنامه عامل یکسانی را پشتیبانی می‌کنند، وجود داشته باشد. عامل‌های سیار این توانایی را دارند که می‌توانند از یک مکان به مکانی دیگر، بین سیستم‌های عامل حرکت نمایند، و این در صورتی است که، نوع سیستم عامل آنها توسط سیستم عامل مقصد قابل شناسایی باشد. سیستم‌های عامل می‌توانند به یک مالک خاص محدود شوند و در این صورت در یک ناحیه قرار می‌گیرند. بر این اساس، عامل‌ها می‌توانند یک نام منحصر به فرد عمومی داشته باشند که از سه تایی (مالک، شناسه عامل، نوع سیستم عاملی) تشکیل می‌گردد. سیستم عامل خود سکویی است که می‌تواند عامل را ایجاد، اجرا، منتقل و ختم کند. سیستم عامل نیز، مانند عامل به مالکی که فرد یا سازمانی که برای آن کار می‌کند را مشخص می‌کند منصوب می‌شود. در اینجا سیستم عامل با نام و آدرسش بصورت یکتا مشخص می‌شود. ممکن است در یک میزبان، یک یا چند سیستم عامل داشته باشیم. نوع سیستم عامل، شناسنامه عامل را توضیح می‌دهد. برای مثال، اگر نوع سیستم عاملی Aplet باشد، سیستم عاملی، توسط IBM پیاده‌سازی شده، زبان Java را بعنوان زبان عامل پشتیبانی می‌کند، از Itinerary برای حرکت استفاده می‌کند و برای سریال‌سازی شی جاوا را بکار می‌برد.



شکل ۲- مدل مفهومی MASIF

این چارچوب توسط دو واسط کوربا مدیریت می‌گردد؛ واسط MFAgentSystem باید توسط سیستم‌های عامل پیاده‌سازی شود تا بتوان عامل‌ها را مدیریت کرد (ایجاد، تعلیق، از سرگیری و ختم) یا عامل‌های سیار مهاجر

۴) Bind: شی ایجاد شده جدید، زمینه اسمی جدیدی در Naming Service ایجاد کرده و نام خود را به آن منسوب می‌کند و سپس کلاینت‌ها می‌توانند با یافتن مولفه نام، کنش‌های لازم را با آن انجام دهند.

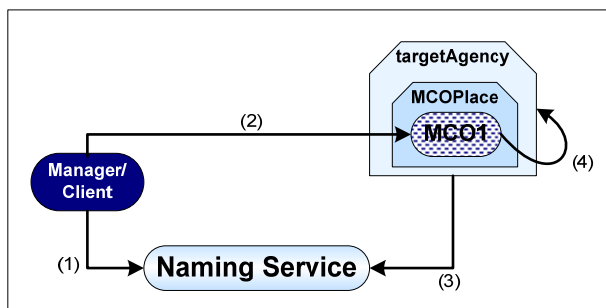


شکل ۴- ایجاد یک شی سیار

۳-۳-۲- حذف شی سیار

سناریوی کنش:

- ۱) Resolve: کلاینت با اتصال به Naming Service، شی‌ی که باید حذف شود را جستجو می‌کند تا بتواند به شی مورد نظر دسترسی داشته باشد.
- ۲) حذف: کلاینت متد terminateME مربوط به MCO را صدا می‌زند.
- ۳) Unbind: MCO با فراخوانی تابع terminateMCO آژانسی که در آن مقیم است، عملیات حذف را می‌آغازد. در ابتدا، آژانس، مدخل نام MCO را از انتساب Naming Service خارج می‌کند.
- ۴) حذف نهایی: پس از حذف انتساب نام MCO از سرویس نامگذاری، آژانس مذکور خود شی MCO را حذف می‌کند.



شکل ۵- حذف شی سیار

۳-۳-۳- انتقال یک شی سیار

سناریوی کنش:

- ۱) Resolve: کلاینت، نام MCO موردنظر را به سرویس نامگذاری ارسال کرده و اشاره‌گر MCO را برمی‌گرداند.
- ۲) آغاز انتقال: کلاینت متد moveMe شی MCO را فراخوانی می‌کند تا انتقال صورت گیرد. برای مثال در شکل فوق انتقال MCO ایجاد شده در factoryAgency را به آژانس مقصد (NEAgency) بیان می‌کند.
- ۳) حال MCO متد moveMCO مربوط به آژانس دربرگیرنده خود را فراخوانی می‌کند تا انتقال به آژانس مقصد را آغاز کند.
- ۴) Resolve: آژانس مبدأ، اشاره‌گر آژانس مقصد را با ارسال درخواست به سرویس نامگذاری دریافت می‌کند.

```

module mco{
//....
interface MobileObject {

void afterBirth(in AgentSystem agnsys, in AgentInfo
info, in Object arguments)
raises(OperationException);

void moveMe(in Location dstLocation, in Place
dstPlace) raises(OperationException);

void beforeMove(in Location dstLocation, in Place
dstPlace) raises(OperationException);

void afterMove(in AgentSystem agnsys, in Location
dstLocation, in Place dstPlace)
raises(OperationException);

void afterMoveFailed(in Location dstLocation, in
Place dstPlace, in Info message);

void beforeDeath();

void beforeSuspend() raises(OperationException);

void beforeResume() raises(OperationException);

void terminateMe() raises(OperationException);

void beforeShutdown();
//....
};
}
    
```

شکل ۳- واسط MobileObject به زبان IDL

جدول ۱- مدیریت دوره زندگی MCO و متدهای واسط MobileObject

Agency operations	involved MCO's call-backs
createMCO	afterbirth
resumeMCO	Resume
suspendMCO	Suspend
moveMCO	moveMe beforeMove afterMove afterMoveFailed
terminateMCO	terminateMe beforeDeath

۳-۳-۳- سناریوهای کنش MCO در محیط

توجه داشته باشید که در سناریوهای نشان داده شده، کنش‌های MAFFinder ارائه نشده است.

۳-۳-۱- ایجاد یک شی سیار

سناریوی کنش:

- ۱) Resolve: کلاینت شی factoryAgency را با استفاده از سرویس نامگذاری resolve می‌کند. این شی آژانس تولیدکننده MCOهای خاصی می‌باشد که کلاینت، مد نظر دارد.
- ۲) ایجاد: کلاینت متد createMCO مربوط به Factory مناسب را صدا می‌زند. این فراخوانی از راه دور بوده و بر اساس پروتکل IIOP صورت می‌گیرد.
- ۳) آژانس تولیدکننده، MCO درخواست شده را از روی کلاس‌های جاوایی که در اختیار دارد، در سیستم عاملی خود و در مکان تخصیص داده شده ایجاد می‌کند.

برای مثال مدیر می‌تواند، برای جمع‌آوری اطلاعات از یک (یا چند) شی، در خواست ساخت یک MCO جمع‌کننده داده را به factoryAgency ارسال کند و سپس آن را با دادن برنامه سفر به سوی اشیای مختلف ارسال کند تا، به جمع‌آوری اطلاعات بپردازد.

همانگونه که در شکل فوق دیده می‌شود، مدیر درخواست تولید شی MCO1 را با فراخوانی راه دور createMCO از آژانس مولد تقاضا می‌کند. در این عملیات، می‌توان پارامترهای لازم برای ایجاد این شی را به آژانس مولد داد (مانند اینجا که برنامه سفر به او داده می‌شود). شی MCO1 ایجاد شده، بلافاصله شروع به حرکت بر روی شبکه کرده و به آژانس نزدیک شی مدیریت‌شونده گسیل می‌شود. حال بر اساس کد نوشته شده در پیاده‌سازی آن، شروع به انجام فراخوانی‌های محلی کرده و اطلاعات لازم را از شی مدیریت‌شونده دریافت می‌کند.

منظور از فراخوانی‌های محلی به معنای فراخوانی‌های با تاخیر کم می‌باشد چرا که، آژانس عنصر شبکه و شی مدیریت‌شونده هر دو در یک شبکه محلی و یا حتی بر روی یک میزبان می‌توانند وجود داشته باشند. شی MCO1 پس از انجام جمع‌آوری داده، به سوی مقصد حرکت می‌کند که می‌تواند همان مدیر آغازکننده انتقال باشد و یا اینکه مدیر دیگری که وظیفه بررسی اطلاعات را دارد، به عنوان مقصد نهایی این MCO مشخص شده است.

۴- ارزیابی

در این بخش به بررسی مدل‌ها و معیارهای مورد ارزیابی پرداخته می‌شود. این مدل‌ها و معیارها بر اساس استفاده از چارچوب کوربا به عنوان زیرساخت ارتباطی مدیریت شبکه بوده و با یکدیگر مقایسه می‌شود. مدل‌ها به دو بخش ایستا و سیار تقسیم‌بندی می‌شود. البته این مقایسه در پارامترهای کارایی انجام می‌گیرد که عمده‌ترین این پارامترها را می‌توان مدت زمان پاسخ به اعمال مدیریتی و نیز ترافیک ایجاد شده بر روی شبکه در نظر گرفت.

در اینجا ترافیک شبکه، هم برای مدیر و هم برای سرویس‌دهنده نامگذاری اندازه‌گیری و مقایسه شده است. آنچه که واضح است، نقش غیر قابل انکار سرویس دهنده‌های نامگذاری^۷ می‌باشد که می‌تواند در برخی مواقع گلوگاه سیستم شود و لذا، از اهمیت بالایی برخوردار است. پس ترافیک اندازه‌گیری شده هم برای مدیر و هم برای سرویس‌دهنده نامگذاری می‌باشد.

برای این ارزیابی‌ها، آژانس و شی بر روی یک ماشین قرار داده شده‌اند. همچنین، اندازه‌گیری‌های انجام شده برای هر مورد، ۱۰۰ بار تکرار شده و میانگین آنها ذکر گردیده است. اجرای این پروژه بر روی ماشین‌های مشابه با مشخصات زیر بوده است:

پردازش‌گر: AMD Sempron 2.4

حافظه: ۵۱۲ مگابایت

سیستم عامل: Microsoft windows xp SP2

ORB: JacORB 2.2.3

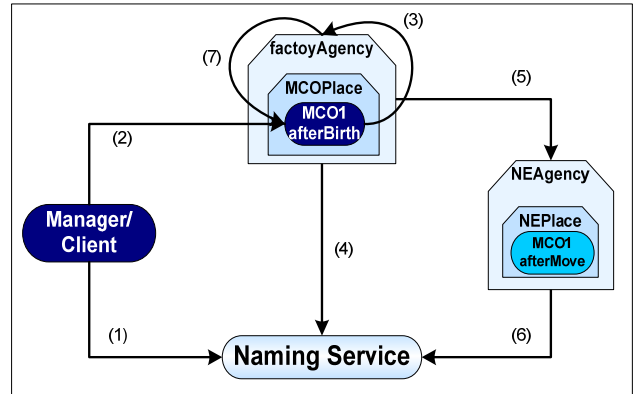
۴-۱- مدل‌های مورد ارزیابی

مدل‌هایی که مورد ارزیابی قرار گرفته است، مبتنی بر ویژگی قابلیت تحرک و همچنین نحوه استفاده از سرویس نامگذاری می‌باشند. بدین ترتیب، یک عملیات مدیریتی می‌تواند به طرق مختلف پیاده‌سازی و اعمال شود. مدل‌ها را می‌توان به سه بخش زیر تقسیم نمود:

(۵) انتقال: آژانس مبدا MCO موردنظر را به آژانس مقصد انتقال می‌دهد که این عمل شامل سریال‌سازی شی MCO و ارسال آن به آژانس مقصد می‌باشد. البته، قابل ذکر است که در اینجا، انتقال حالت اجرایی MCO مدنظر نبوده و فقط حالت داده آن برونی شده و ارسال می‌شود.

(۶) Rebind: آژانس مقصد فرم برونی شده MCO را دریافت کرده و بر اساس کلاس‌هایی که به آنها دسترسی دارد، شی مورد نظر را در زمینه خود ساخت و با استفاده از سرویس نامگذاری، نام آن را Rebind می‌کند.

(۷) حذف: پس از آنکه شی MCO با موفقیت و بدون خطا به آژانس مقصد انتقال یافت، آژانس مبدا MCO موجود در محیط اجرایی خود را حذف می‌کند.

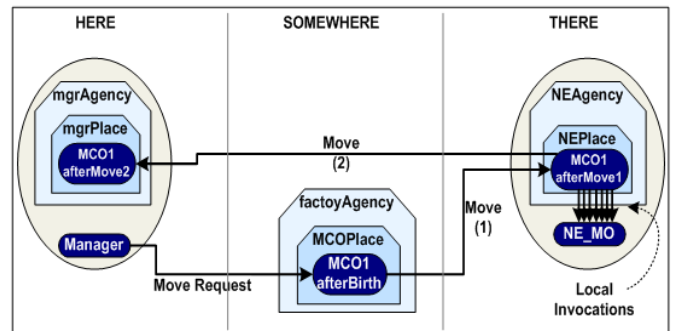


شکل ۶- انتقال یک شی سیار

در اینجا نحوه استفاده از این چارچوب برای اهداف مدیریت شبکه بیان می‌شود. سه نوع آژانس در شبکه موجود است:

- (۱) آژانس مدیر، که در نزدیکی اشیای مدیر ایجاد می‌شود.
- (۲) آژانس مولد، که در هر جایی از شبکه می‌تواند قرار گیرد.
- (۳) آژانس شی مدیریت‌شونده، که در نزدیکی شی مدیریت‌شونده استقرار دارد.

مدیر شبکه به عنوان یک شی کوربا، در نزدیکی آژانس مدیر است. و شی مدیریت‌شونده نیز خود یک شی است که در نزدیکی آژانس عنصر شبکه قرار دارد. شی MCO هم توسط آژانس مولد ایجاد می‌شود. منظور از عبارت "نزدیک"، حضور در یک شبکه محلی یا روی یک میزبان می‌باشد. مدیر می‌تواند با درخواست از آژانس مولد، شی MCO خاص موردنظر خود را ایجاد کرده و آن را به نزدیک شی مدیریت‌شونده ارسال نموده و سپس آن را به مقصد دیگر بفرستد و یا اینکه برنامه سفر آن را از ابتدا مشخص نماید.



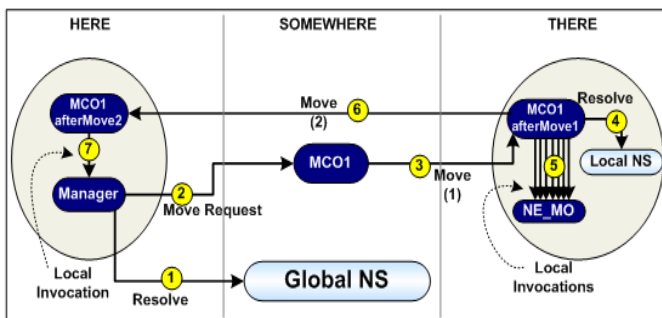
شکل ۷- الگوی مدیریت شبکه با استفاده از MCO

۴-۱-۱- مدل ایستا

این شبکه خواستار دسترسی به آنها می‌باشند، بایستی به این سرویس‌دهنده دسترسی داشته باشند. فرض کنید مدیر شبکه در یک سرویس‌دهنده نام عمومی خود را ثبت کرده باشد و اشیای مدیریت شونده نیز در سرویس‌دهنده های محلی خود ثبت شده باشند. در این صورت مدیر نمی‌تواند از طریق سرویس‌دهنده عمومی به آنها دسترسی پیدا کند. برای رفع این مشکل دو راه حل وجود دارد: یکی اینکه، مدیر باید نگاهی از اشیای مدیریت‌شونده و سرویس‌دهنده های آنها داشته باشد تا بتواند با درخواست از سرویس‌دهنده مناسب، اشاره‌گر شی را به دست آورد که این خود باعث افزایش بار پردازشی مدیر شده و در بسیاری از موارد که سرویس‌دهنده‌های محلی دور از مدیر هستند، باعث افزایش تاخیر می‌شود. همچنین بار ترافیکی مدیر را افزایش می‌دهد. روش دیگر این است که شی مدیریت‌شونده علاوه بر ثبت در سرویس‌دهنده های محلی در سرویس‌دهنده های عمومی نیز ثبت گردد که این روش مشکل خاصی را حل نموده و همان مدل اول را تداعی می‌کند.

تمامی این مشکلات به دلیل عدم توانایی در انتقال اشیا به حوزه‌های نامگذاری مختلف است. مدلی که پیشنهادی ما برای سیستم مدیریت شبکه طرح به این ترتیب است که هر آژانس یک جفت سرویس‌دهنده محلی و عمومی دارد. اسامی آژانس‌ها و MCOها در سرویس‌دهنده نام عمومی ثبت می‌شود و از سرویس‌دهنده‌های محلی برای ثبت اسامی اشیای مدیریت‌شونده و مدیران استفاده می‌شود.

روال کار به این ترتیب است که: مدیر، از سرویس‌دهنده عمومی درخواست یافتن شی MCO مورد نظر خود را می‌کند و سپس آژانس دربرگیرنده این شی نیز از NS عمومی درخواست اشاره‌گر آژانس مقصد نموده و MCO به مقصد ارسال می‌شود. در آژانس مقصد، MCO، نام شی مدیریت‌شونده را از سرویس‌دهنده محلی درخواست نموده و سپس فراخوانی‌ها را انجام داده و با ارسال درخواست به سرویس‌دهنده عمومی برای یافتن آژانس شی مدیر، به آژانس مبدا برمی‌گردد (همانگونه که در شکل زیر توضیح داده شده است).



شکل ۸- مدیریت شبکه با استفاده از MCO و سرویس‌دهنده محلی

این مدل، همان مدل عمومی و سنتی استفاده از کوربا برای پردازش توزیعی توسط اشیای توزیع‌شده می‌باشد که در آن اشیا به صورت توزیع‌شده بر روی میزبان‌های مختلف در شبکه قرار گرفته و با فراخوانی‌های از راه دور رویه‌های یکدیگر به کنش متقابل می‌پردازند. در این مدل، اشیا به هیچ وجه دارای قابلیت تحرک نبوده و بر روی میزبانی که اجرا شده و مقیم شده‌اند باقی می‌مانند. مدیر شبکه به عنوان یک شی ایستا بر روی یک میزبان خاص قرار داشته و اشیای مربوط به تجهیزات مدیریت‌شونده در میزبان‌های دیگر استقرار یافته‌اند. مدیر برای انجام اعمال مدیریتی، درخواست‌های فراخوانی را بر روی شبکه و به صورت مستقیم به شی مدیریت‌شونده ارسال می‌دارد. در این مدل، نقش سرویس‌دهنده بسیار کلیدی است چرا که باید تمامی اشیای مدیریت‌شونده در آن ثبت شوند تا مدیر بتواند مستقیماً متدهای آن را فراخوانی نماید. با افزایش تعداد اشیای مدیریت‌شونده، سرویس نامگذاری ممکن است دچار اختلال شود.

۴-۱-۲- مدل سیار

در این مدل، اشیا می‌توانند بر روی میزبان‌های مختلف حرکت نمایند. این مدل، از نظر برنامه‌نویسی و پیاده‌سازی سیاست مدیریتی، شبیه مدل قبل می‌باشد و مکان اشیا نیز کاملاً مشخص می‌باشد. بر اساس مکانیزم گفته شده در بخش‌های قبل، اشیا می‌توانند در شبکه حرکت نموده و دوباره خود را در سرویس نامگذاری ثبت نمایند، لذا این امر تقریباً تاثیری در پیاده‌سازی‌ها ندارد. در این مدل، مدیر برای انجام وظایف مدیریتی از اشیای سیار بهره می‌گیرد. اشیا سیار می‌توانند ساختار بسیار ساده و یا پیچیده داشته باشند و این بستگی به میزان هوشمندی و منطقی دارد که مدیر لازم می‌داند تا در آنها پیاده‌سازی شود. در مدلهایی که ارزیابی شده است، از اشیا سیار برای جمع‌آوری اطلاعات استفاده شده است به این ترتیب که آنها بتوانند به نزدیک اشیای مدیریت‌شونده مهاجرت نموده و فراخوانی‌های راه دور را از نزدیک انجام داده و نتایج را در خود نگهداشته و به مدیر انتقال دهند. پر واضح است که این اشیا می‌توانند اعمال مختلفی نظیر خلاصه‌سازی، فشرده‌سازی و یا یک سری پیش‌پردازش‌های دیگر را در میزبان انجام دهند.

در این مدل نیز همانند مدل پیشین، تنها یک سرویس نامگذاری موجود بوده و تمامی آژانس‌ها، اشیای سیار، اشیای مدیریت‌شونده و مدیر در آن ثبت شده‌اند لذا، در این طرح نیز نقش سرویس‌دهنده نام بسیار مهم بوده و ممکن است سیستم مدیریت شبکه را دچار اختلال نماید.

۴-۱-۳- مدل سیار با سرویس‌دهنده محلی نامگذاری

یکی از مشکلات اساسی که در دو مدل قبل دیده می‌شود، استفاده از "یک" سرویس‌دهنده نام برای تمامی اشیای موجود در سیستم مدیریت شبکه می‌باشد. یکی از مشکلات اساسی مدل ایستا، اجبار در ثبت اشیا در یک سرویس‌دهنده قابل دسترسی توسط مدیر و اشیای مدیریت‌شونده می‌باشد. در مدل اول برای رفع مشکل گلوگاه بودن سرویس‌دهنده نام، از خوشه‌بندی سلسله‌مراتبی استفاده می‌کنند. به این ترتیب که اشیایی را که در حوزه دسترسی یکدیگر قرار دارند، بر روی یک سرویس‌دهنده نام ثبت می‌کنند و این اشیا با اشیای حوزه سرویس‌دهنده نامگذاری دیگر نمی‌توانند ارتباط برقرار کنند. به عنوان مثال فرض می‌کنیم برای هر شبکه محلی از اشیا، یک سرویس‌دهنده نام در نظر گرفته می‌شود و اشیای ثبت شده در آن فقط می‌توانند به همدیگر دسترسی پیدا کنند و اشیایی که از بیرون از

۴-۲- معیارهای مورد ارزیابی

اندازه‌گیری‌های انجام شده در پارامترهای کارایی می‌باشد که مهم‌ترین این معیارها را می‌توان مدت زمان پاسخ به اعمال مدیریتی و نیز ترافیک ایجاد شده بر روی شبکه در نظر گرفت. در ادامه به بررسی بیشتر آنها و نتایج مربوطه پرداخته می‌شود.

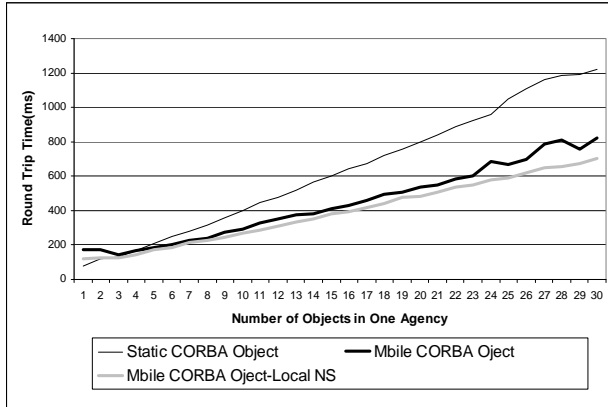
۴-۲-۱- اندازه‌گیری زمان پاسخ

زمان پاسخ برای هر یک از مدل‌های ایستا و سیار چنین تعریف می‌شود:

- ۳) تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
- ۴) تعداد اشیای مدیریت‌شونده برای هر آژانس

• مورد ۳

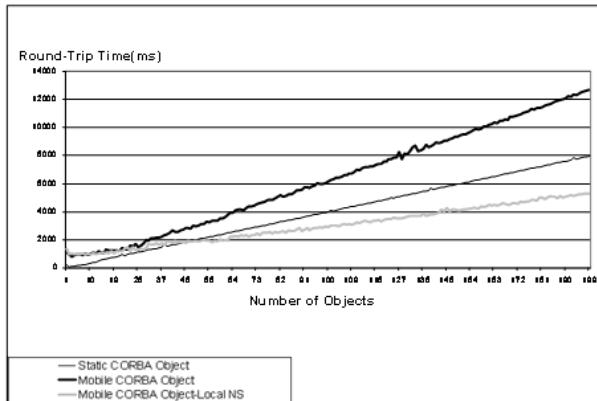
10Mbps Ethernet	پهنای بند
50Bytes	اندازه پاسخ برگشتی برای هر ویژگی
30	تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
Variable	تعداد اشیای مدیریت‌شونده برای هر آژانس



شکل ۱۱- نمودار زمان پاسخ بر حسب تعداد اشیا

• مورد ۴

64Kbps	پهنای بند
100Bytes	اندازه پاسخ برگشتی برای هر ویژگی
Variable	تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
1	تعداد اشیای مدیریت‌شونده برای هر آژانس



شکل ۱۲- نمودار زمان پاسخ بر حسب تعداد ویژگی با پهنای بند کم

۴-۲-۲- تفسیر نتایج اندازه‌گیری زمان پاسخ

همانگونه که در اکثر نمودارها دیده می‌شود، زمان پاسخ شی بسیار با سرویس محلی بهتر از مدل‌های دیگر است.

موارد ۱ نتیجه تاثیر اندازه پاسخ برگشتی بر روی زمان پاسخ را نشان می‌دهد. در این موارد تعداد ویژگی‌ها، متغیر است. به علت اینکه پاسخ برگشتی برای هر فراخوانی، اندازه کمی دارد لذا زمان ارسال و دریافت پاسخ به همراه سرآیند، در برابر مدل بسیار قابل ملاحظه می‌باشد. در مورد ۲ تاثیر تعداد فراخوانی‌ها با حجم

• زمان پاسخ برای مدل ایستا عبارت است از مدت زمان میان ارسال درخواست به سرویس‌دهنده نام برای یافتن شی مدیریت‌شونده و دریافت کل نتایج فراخوانی‌ها.

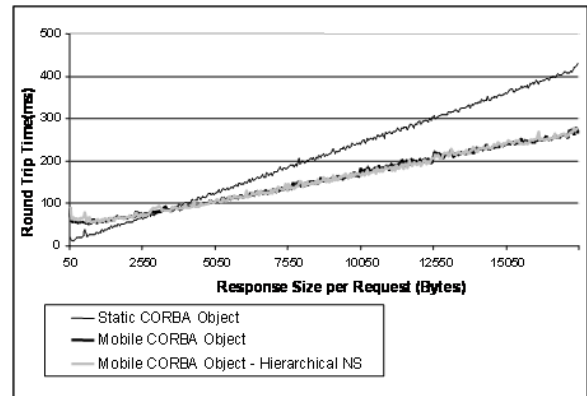
• زمان پاسخ برای مدل سیار عبارت است از مدت زمان میان ارسال درخواست به سرویس‌دهنده نام برای یافتن شی MCO و ارسال آن به سمت اشیا مدیریت‌شونده و بازگرداندن نتایج فراخوانی‌ها به مدیر. اندازه‌گیری‌های انجام شده برای حالت‌های مختلف در موارد زیر آورده شده است. در هر یک از آنها شرایط آزمایش بیان شده است.

این شرایط عبارتند از:

- ۱) پهنای بند شبکه
- ۲) اندازه پاسخ برگشتی برای هر ویژگی (که معادل با یک فراخوانی راه دور می‌باشد)

• مورد ۱

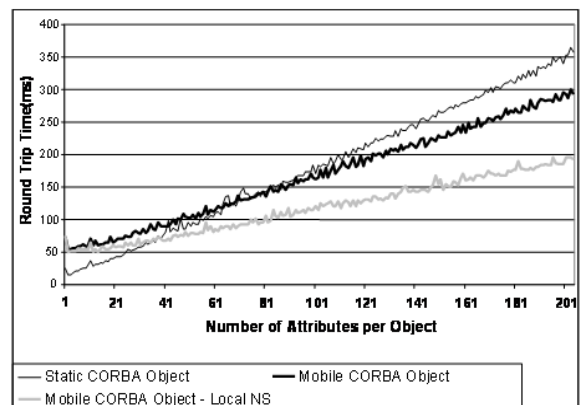
10Mbps Ethernet	پهنای بند
Variable	اندازه پاسخ برگشتی برای هر ویژگی
50	تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
1	تعداد اشیای مدیریت‌شونده برای هر آژانس



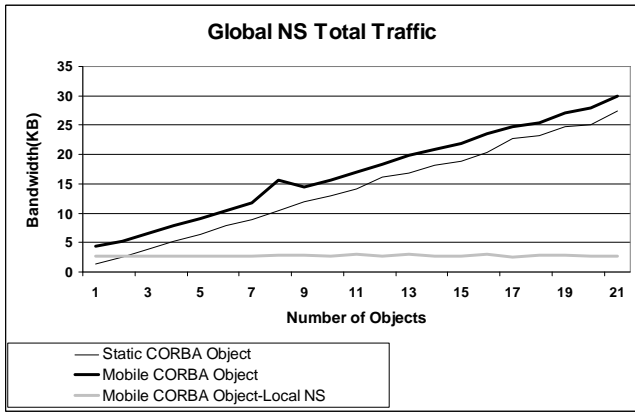
شکل ۹- نمودار زمان پاسخ بر حسب اندازه پاسخ

• مورد ۲

10Mbps Ethernet	پهنای بند
100Bytes	اندازه پاسخ برگشتی برای هر ویژگی
Variable	تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
1	تعداد اشیای مدیریت‌شونده برای هر آژانس



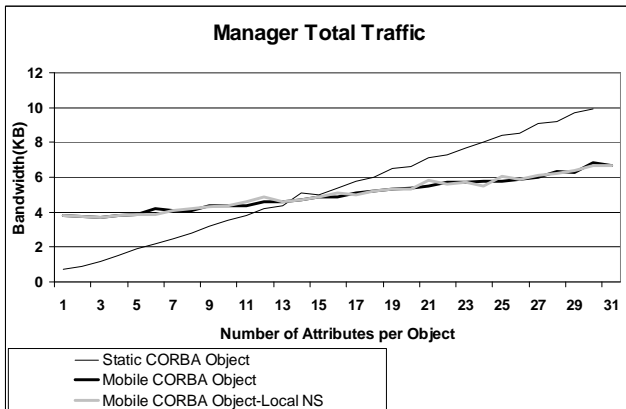
شکل ۱۰- نمودار زمان پاسخ بر حسب تعداد ویژگی‌ها



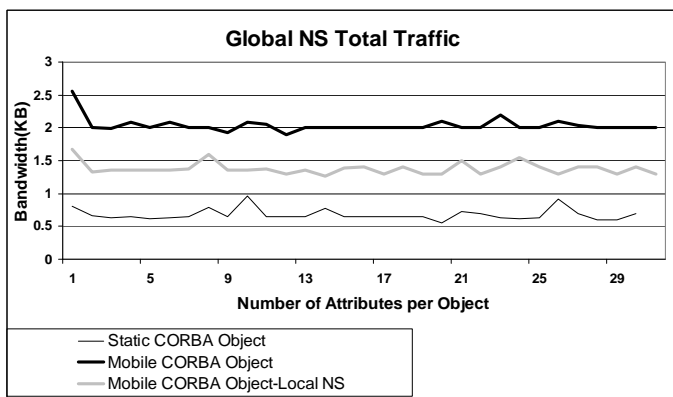
شکل ۱۴- نمودار ترافیک سرویس‌دهنده نام بر حسب تعداد اشیاء

• مورد ۲

100 Bytes	اندازه پاسخ برگشتی برای هر ویژگی
Variable	تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
1	تعداد اشیای مدیریت‌شونده برای هر آژانس



شکل ۱۵- نمودار ترافیک مدیر بر حسب تعداد ویژگی‌ها



شکل ۱۶- نمودار ترافیک سرویس‌دهنده نام بر حسب تعداد ویژگی‌ها

۵- نتیجه‌گیری

رویکرد کاربردهای مدیریت شبکه به سمت استفاده از عامل‌های هوشمند سیار می‌باشد تا بتوانند مشکلات استفاده از معماری‌های سنتی و چارچوب‌های قدیمی را

داده‌های متفاوت بر روی زمان پاسخ نشان داده می‌شود. در این حالت نیز، معمولاً نتایج مدل‌های سیار بهتر از ایستا می‌باشد. در موارد ۳ و ۴ نیز تاثیر تعداد اشیاء بر روی زمان پاسخ بررسی شده است و در آنها هم تعداد ویژگی و هم پهنای باند شبکه را تغییر داده و نتایج ثبت شده است. نتایج نشان می‌دهد که در تمامی موارد، مدت زمان پاسخ مدل‌های سیار بهتر از مدل ایستا می‌باشد.

در کل باید این موضوع در نظر گرفته می‌شود که مدت زمان پاسخ به پارامترهای زیادی وابسته است و در مواردی که در بالا ذکر شده است، سعی شده است شرایط یکی در نظر گرفته شود ولی آنچه که واضح است این است که در واقعیت، شبکه می‌تواند بسیار متفاوت بوده و تاخیرهای متفاوت داشته باشد همان طور که قدرت پردازشی میزبان‌ها نیز متغیر است که در آن حالات، مدل‌های سیار از پاسخ بهتری برخوردار است.

همانگونه که در نمودارهای فوق نمایش داده می‌شود، تفاوت میان مدل‌های سیار با سرویس‌دهنده عمومی و محلی هنگامی اتفاق می‌افتد که میزان رجوع به سرویس‌دهنده قابل ملاحظه می‌باشد. همچنین تاخیر مدت زمان ارسال و دریافت داده در مقابل تاخیر درخواست از سرویس‌دهنده نباید قابل توجه باشد.

۴-۲-۳- اندازه‌گیری ترافیک

ترافیک اندازه‌گیری شده هم برای حالت ایستا و هم برای سیار عبارت است از ترافیک ارسال، ترافیک دریافت و ترافیک مجموع برای مدیر و سرویس‌دهنده نام عمومی.

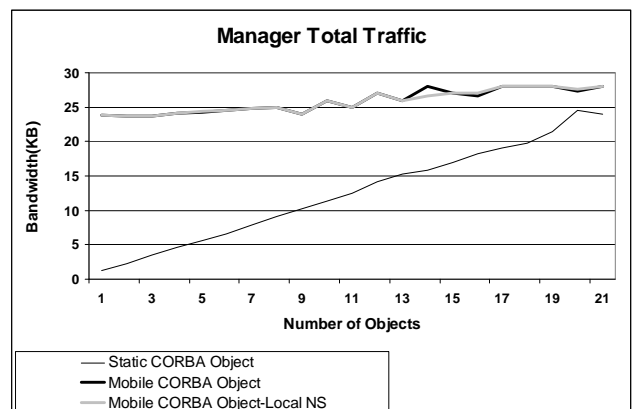
اندازه‌گیری ترافیک، برای سه حالت انجام شده است.

- یکی آنکه پاسخ فراخوانی‌ها متغیر باشد.
- دیگری برای حالتی که تعداد اشیای مدیریت‌شونده داخل آژانس‌ها متغیر باشد.
- و در پایان برای حالتی که تعداد آژانس‌ها متغیر باشد.

همانگونه که در شکل‌ها نیز مشاهده می‌شود، در کل استفاده از MCO باعث کاهش ترافیک بر روی مدیر و نیز سرویس‌دهنده عمومی می‌شود. علاوه بر آن باعث از میان رفتن خاصیت گلوگاهی آنها نیز می‌شود.

• مورد ۱

100 Bytes	اندازه پاسخ برگشتی برای هر ویژگی
1	تعداد ویژگی‌ها برای هر شی مدیریت‌شونده
Variable	تعداد اشیای مدیریت‌شونده برای هر آژانس



شکل ۱۳- نمودار ترافیک مدیر بر حسب تعداد اشیاء

[7] T. M. Chen, S. S. Liu, "A Model and Evaluation of Distributed Network Management Approaches," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 850-857, May 2002.

[8] IBM, *Mobile Agent System Interoperability Facilities Specification*. Joint submission: GMD Fokus & IBM Corp., supported by Crystaliz Inc., General Magic Inc., The Open Group. OMG TC document orbos/97-10-05 (1997).



سیده محمد رضا میرزابابائی فارغ‌التحصیل کارشناسی مهندسی کامپیوتر از دانشگاه صنعتی امیرکبیر در سال ۱۳۷۴ و کارشناسی ارشد هوش ماشین و ریاتیک از دانشکده فنی دانشگاه تهران در سال ۱۳۷۷ و هم‌اکنون دانشجوی دکترای رشته معماری کامپیوتر در دانشگاه صنعتی امیرکبیر. علاقمند به موضوعات هوش مصنوعی و پردازش موازی. آدرس پست‌الکترونیکی ایشان عبارت است از:

mirzababaei@aut.ac.ir



محمد کاظم اکبری فارغ‌التحصیل کارشناسی دانشگاه شهید بهشتی در سال ۱۳۶۲ و کارشناسی ارشد و دکترا در مهندسی کامپیوتر از دانشگاه کیس وسترن رزرو در سال‌های ۱۹۹۱ و ۱۹۹۵. علاقمند به موضوعات شبکه‌های عصبی، طراحی سیستمی و پردازش موازی، پردازش توری و معماری کامپیوتر. آدرس پست‌الکترونیکی ایشان عبارت است از:

akbari@aut.ac.ir

محمود صادقی فارغ‌التحصیل کارشناسی و کارشناسی ارشد رشته معماری کامپیوتر دانشگاه صنعتی امیرکبیر در سال‌های ۱۳۸۷ و ۱۳۹۰. علاقمند به موضوعات پردازش متحرک، پردازش ابری و موازی. آدرس پست‌الکترونیکی ایشان عبارت است از:

sadeghi@aut.ac.ir

رفع کند. مشکلاتی نظیر مقیاس‌پذیری، قابلیت اطمینان، قابلیت دسترسی، امنیت، کارایی و ... همچنان در کاربردهای مدیریت شبکه مطرح هستند. این مسایل را تا حد زیادی می‌توان با استفاده از عامل‌های سیار و یا ترکیب آنها با سیستم‌های مدیریت شبکه موجود حل نمود. این تحقیق برای ارزیابی و سنجش پویای معیارهای کارایی انجام شده است. با یافتن ضرائب مناسب برای معیارهای مختلف قابل اندازه‌گیری می‌توان مدیریت بسیار بهتری بر انجام پردازش در شبکه خصوصاً شبکه‌های پردازش ابری و نسل آینده محیط فراگیر انجام داد. در این پروژه نمونه بسیار کوچکی از قابلیت‌های (قابلیت تحرک) عامل‌های سیار هوشمند به کوربا اضافه شده است. نتایج مشاهده شده، حاکی از بهبود کارایی سیستم مدیریت شبکه پیاده‌سازی شده است.

تشکر و قدردانی

این پروژه با حمایت مالی پژوهشگاه تحقیقات ارتباطات و فناوری اطلاعات انجام گرفته است.

اختصارات

CMIP	Common Management Information protocol
CORBA	Common Object Request Broker Architecture
DNM	Distributed Network Management
HS	Hierarchical static
IIOIP	Internet Inter-ORB Protocol
LAN	Local Area Network
MA	Mobile Agent
MASIF	Mobile Agent System Interpretability Facility
MCO	Mobile CORBA Object
NE	Network Element
NMS	Network Management System
NS	Name Server
OMG	Object Management Group
ORB	Object Request Broker
SM	Strong Mobility
SNMP	Simple Network Management Protocol
WM	Weak Mobility

مراجع

- [1] W. Stallings, *SNMP, SNMPv2 and RMON: Practical Network Management*. Addison-Wesley 1999.
- [2] ITU-T Recommendation M.3010, *Principles for a Telecommunication Management Network*, May 1996.
- [3] J. Case, M. Fedor, M. Schoffstall, and C. Davin, "The Simple Network Management Protocol (SNMP)," *RFC 1157*, May 1990.
- [4] ITU-T Recommendation X.711, *Common Management Information Protocol (CMIP)-Part 1: Specification*, 1991.
- [5] OMG, *The Common Object Request Broker: Architecture and Specification*, Revision 2.4, Oct 2000.
- [6] E. Reuter, and F. Baude, "System and Network Management Itineraries for Mobile Agents," MATA, 2002.

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۵/۱۱/۳

تاریخ اصلاح: ۹۱/۵/۲

تاریخ قبول شدن: ۹۱/۸/۱۶

نویسنده مرتبط: دکتر محمد کاظم اکبری، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران.

¹ Centralized NMSs
² Network Element (NE)
³ Mobile Agents
⁴ Request-Reply
⁵ Authority
⁶ Streamable
⁷ Name Server

نسخه نهائی مقالات ارسالی برای چاپ در نشریه علوم و مهندسی کامپیوتر باید بر طبق اصول مطرح شده در این راهنما تهیه شده باشد. رعایت این اصول در نسخه اولیه نیز قویاً توصیه می‌شود. مقالات می‌تواند به یکی از دو زبان فارسی یا انگلیسی ارسال گردد.

۱- ساختار مقاله

- عنوان: کوتاه و معرف محتوای مقاله باشد و از ۱۵ کلمه تجاوز نکند.
- نام نویسندگان و مؤسسه محل اشتغال آنان: از ذکر عناوین خودداری شود.
- چکیده فارسی: حاوی تعریف مسأله، روش حل، و نتایج مهم باشد و از ۱۵۰ کلمه تجاوز نکند.
- واژه‌های کلیدی: حداکثر ۱۰ کلمه
- بدنه اصلی مقاله: بدنه اصلی با "مقدمه" شروع و با "نتیجه‌گیری" خاتمه می‌یابد. بخش‌ها و زیربخش‌های بدنه اصلی باید شماره‌گذاری شوند.
- شماره "مقدمه" یک خواهد بود.
- تشکر و قدردانی (در صورت نیاز).
- مراجع: مراجع به ترتیبی که در متن به آنها رجوع می‌شود آورده شوند. نام مؤلفان مراجع در صورت لزوم در متن بصورت فارسی آورده شود. رجوع به مراجع با ذکر شماره آنها در داخل کروشه ([]) انجام پذیرد.
- پیوست‌ها (در صورت نیاز)
- واژه‌نامه (در صورت نیاز)
- برای مقالات فارسی، عنوان مقاله، نام نویسندگان، مؤسسه محل اشتغال، چکیده، و واژه‌های کلیدی به زبان انگلیسی نیز در صفحه‌ای جداگانه داده شود. برای مقالات انگلیسی این اطلاعات به زبان فارسی نیز در صفحه‌ای جداگانه داده شود.

۲- معادله‌ها، شکل‌ها، جدول‌ها، و عکس‌ها

- معادله‌ها باید با فاصله کافی از بالا و پائین تایپ و به صورت متوالی شماره‌گذاری شوند. شماره معادله در پراکنش در انتهای سمت راست سطر حاوی معادله قرار داده شود. معادلات دستنویس به هیچ شکل قابل قبول نیستند.
- شکل‌ها و جدول‌ها باید دارای شماره و عنوان باشند. در شکل‌ها شماره و عنوان در زیر شکل و در جدول‌ها در بالای شکل قرار می‌گیرد. اعداد و متون روی شکل‌ها و جدول‌ها باید دارای اندازه مناسب و کاملاً خوانا باشند.
- اعداد و کلمات روی شکل‌ها و جدول‌ها در مقالات فارسی به زبان فارسی باشند.
- عکس‌ها سیاه و سفید، برقی، و با کیفیت عالی باشند.

۳- نحوه نگارش مراجع

در لیست مراجع انواع مختلف مرجع‌ها به شکل زیر نوشته شوند:

[۱] ب. مقدم، ا. تقوی، و ن. طاهری، آشنائی با شبکه‌های کامپیوتری، چاپ دوم، انتشارات نصر، تهران، ۱۳۷۵.

[۲] ی. براون، مقدمه‌ای بر شبکه‌های عصبی، ترجمه م. ع. آرام، انتشارات فجر، مشهد، ۱۳۷۰.

[۳] راهنمای کاربران حسابرس، شرکت پردازش رایانه‌ای ایران، تهران، ۱۳۶۵.

[۴] ج. عارف، استنتاج فازی بوسیله شبکه‌های عصبی، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شهر، ۱۳۷۴.

[۵] ج. حسینی، و ج. ربانی، "تشخیص چهره انسان در تصویر"، نشریه امیرکبیر، سال هشتم، شماره ۴۲، ص ۱۲۵-۱۴۷، ۱۳۷۷.

[۶] ج. حسینی، و ج. ربانی، "تشخیص چهره انسان در تصویر"، در مجموعه مقالات هفتمین کنفرانس سالانه انجمن کامپیوتر ایران، ص ۲۲۴-۲۳۲، ۱۳۸۰.

[7] M. A. Ahmadi, and M. H. Rahimi, *Fuzzy Set Theory*, New Jersey: Prentice-Hall, 1995.

[8] M. A. Ahmadi, M. H. Rahimi, and A. Fatemi, "Evidence-Based Recognition of 3D Objects," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 12, no. 10, pp. 811-835, 1994.

[9] A. Taheri, "On-Line Fingerprint Verification," *Proc. IEEE Intl Conf. Pattern Recognition*, pp. 752-758, 1992.

[10] M. A. Ahmadi, *On-line Fingerprint Verification*, Ph. D. Dissertation, MIT, Cambridge, MA, 1982.

[11] A. J. Washington, "The Fingerprint of MalcomX," <http://www.dermatoglyphics.com>, June 2003.

[12] International Biometrics Group, <http://www.biometricgroup.com>, May 2003.

۴- واژه‌نامه

هر واژه خارجی در واژه‌نامه در انتهای مقاله با شماره‌ای مشخص شود و شماره در معادل فارسی آن واژه در متن، بصورت بالانویس آورده شود.

۵- آماده‌سازی مقاله

- مقاله را با نرم افزار WORD XP تحت ویندوز XP تایپ و با چاپگر لیزری روی کاغذ A4 چاپ نمایید.
- متن چکیده به صورت تک ستونی با طول سطر ۱۸ سانتیمتر و متن مقاله به صورت دو ستونی با طول هر ستون ۸۷ میلیمتر و فاصله دو ستون ۶ میلیمتر تایپ شود. حاشیه‌ها از بالا و پائین برابر ۲۰ میلیمتر و از طرفین برابر ۱۵ میلیمتر اختیار شود.
- فاصله عنوان مقاله در صفحه اول از بالای صفحه برابر ۸۵ میلیمتر باشد و عنوان وسط چین شود.
- کلیه عناوین بصورت پررنگ با قلم "Nazanin یا B Nazanin" تایپ شوند، اندازه قلم عنوان مقاله ۱۸، عناوین سطح اول ۱۴، و عناوین سطح دوم و سوم ۱۲ انتخاب شوند.
- متن چکیده‌ها با قلم "Nazanin یا B Nazanin" اندازه ۹، متن مقاله با قلم "Nazanin یا B Nazanin" اندازه ۱۰، و کلمات و متن انگلیسی با قلم Times New Roman اندازه ۹ تایپ شوند.
- تمام متن بصورت تک فاصله تایپ شود. اسامی نویسندگان از عنوان مقاله و اسامی نویسندگان از عناوین محل اشتغال نویسندگان دو خط فاصله داشته باشد. بالای هر عنوان یک سطر فاصله قرار داده شود.
- سعی شود تعداد صفحات مقاله از ۳۰ صفحه بیشتر نباشد.

۶- نحوه ارسال مقاله

- ارسال مقاله فقط از طریق تارجهان گستر مجله (www.jcse.ir) انجام شود.
- مقاله ارسالی برای نشریه علوم و مهندسی کامپیوتر نباید در جای دیگری به چاپ رسیده باشد و یا در زمان بررسی توسط نشریه برای چاپ به نشریه دیگری ارسال گردد. نویسندگان باید به همراه مقاله نامه‌ای دال بر این مطلب نیز ارسال دارند.
- پس از قبول مقاله، نسخه نهائی تصحیح شده مقاله باید هم بصورت چاپ شده و هم در فایل کامپیوتری در قالب‌های DOC و PDF روی یک دیسکت به نشریه ارسال گردد. همچنین فرم "اختصاص حق انتشار" باید توسط نویسندگان تکمیل، امضاء و ارسال شود.
- در نسخه نهائی باید بیوگرافی و عکس کلیه مؤلفین نیز در انتهای مقاله قرار داده شود.

[4] M. A. Ahadi, *On-line Fingerprint verification*, Ph.D. Dissertation, University of Tehran, Tehran, Iran, 1994.

[5] M. A. Washington, "The Fingerprint of Malcom X," <http://www.dermatologyphicisc.com>, June 2004.

[6] International Biometrics Group, <http://www.Biometrics.com>, May 2003.

4. Manuscript preparation

Typing and Printout: The manuscript should be prepared using Microsoft Word and should be printed on A4 size paper using a laser printer.

Fonts: Use Times New Roman font type. The font size must be 9 for the Abstract and 10 for the main text. All the titles should be made bold. The paper title must have a font size of 18, first-level sub-titles a size of 14, second-and third-level sub-titles a size of 12.

Layout: The paper should be typed in 2-column single space format. The top margin for the first page should be equal to 85 mm, each column length should be 87mm, and the spacing between columns 6mm. Only the abstract is to be typed in a single column format. Two space line must be allowed between the paper title and authors' names, and one space line should be allowed above each section title or sub-title.

5. Paper Submission

Exclusive Submission: Submission of a paper to JCSE implies that it has not been published previously, that it is not under consideration for publication elsewhere, and that if accepted, it will not be published elsewhere in any language. Explicit announcement of this matter must be made in a letter to the editor.

Paper Length: The paper is expected to be no longer than 30 pages.

Submission Process: Authors are requested to submit their papers electronically in PDF format at the journal homepage (www.jcse.ir). All relevant correspondence should be addressed to csi-jcse@ipm.ir.

Review Process: Each manuscript will be reviewed by experts and their constructive criticism will be forwarded to the authors.

Final Manuscript: After the acceptance of a paper, the authors should provide the final manuscript in PDF and Word formats. In addition, the authors should sign and submit the form for the transfer of copyright to the Computer Society of Iran.

INFORMATION FOR AUTHORS

The final manuscript of the English papers accepted for publication in the CSI JCSE should be prepared in accordance with this guide. It is, however, strongly recommended that preparation of the initial manuscript also follows the instructions given here. Manuscript may be in English or Farsi.

1. Paper Organization

- Title: The title should be short (at most 15 words) and indicative of the paper contents.
- Authors: the authors' names (initial and last name) and their affiliations should appear next.
- Abstract: the abstract should include the problem explanation, methods used for solution, and the significant results; and should not be longer than 150 words.
- Keywords: The keywords should be relatively independent and together optimally characterize the paper. Include 5 to 10 keywords.
- Text: the main body of the paper should start with Introduction and end with concluding remarks. All sections and sub-sections should be numbered. The number for introduction is 1.
- Acknowledgement: If required, Acknowledgements appear after the concluding remarks.
- References: All publications cited in the text should be presented in the order they are cited in a numbered list of References following Acknowledgements. In the text refer to references by the reference number in square brackets on the line.
- Appendices: If appendices are necessary, they are placed after the list of References.
- Farsi Section: the title, authors' names and affiliations, abstract, and keywords should also be given in Farsi a separate sheets (for papers written in English). For the non-Farsi speaking authors, the journal will supply this section.

2. Figures, Tables, Photographs and Equations

- Figures and Tables: Each figure or table must have a number and a caption. In figures, the number and the caption appear under the figure while in tables, they appear over it. The size of text and numbers in tables and figures must be suitable to allow high legibility. Do not use any type of shading in computer generated illustrations.
- Photographs: High quality glossy black and white photographs must be supplied as they are to be reproduced.
- Equations: Equations are to be numbered consecutively. The number of each equation should appear in parentheses in the right-most end of the equation line. Sufficient space should be allowed above and below each equation.

3. References

The format for various types of references should be as follows:

- [1] M. A. Ahadi, and M. H. Rahimi, *Fuzzy Set Theory*, New Jersey, Prentice-Hall, 1995.
- [2] M. A. Ahadi, M. H. Rahimi, and A. Fatemi, "Evidence-Based Recognition of 3D Objects," *IEEE Trans. Pattern Analysis and Mach. Intell.*, vol. 12, no. 10, pp. 18-25, 1994.
- [3] A. Taheri, "On-line Fingerprint verification," *Proc, IEEE Int'l Conf. Pattern Recognition*, pp. 752-759, 1992.

Performance Criteria Evaluation for Network Computation Management by Mobile Agents

**Sayyed MohammadReza Mirzababaei Mohammad Kazem Akbari
Mahmoud Sadeghi**

Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

ABSTRACT

Intelligent mobile agents are the aim of any network management applications paradigm to upgrade the regular architectures to the Pervasive environment scenario. The scalability, security, availability, durability and performance have important roles in network management, exclusively its scheduling. This management and its properties could be optimized by mobile agents. The Object Request Brokers (ORB)-based management systems could measure the performance criteria and schedule the resources. In this study we enriched the CORBA with the movability according to the MASIF standard to engage the local Name Servers. The results show some improvements in the network resource scheduling capabilities considering the average response time and the networks throughputs.

Keywords: Pervasive Environment, CORBA, Mobile Agents, Distributed Network Management, Performance Evaluation.

Proposing An Algorithm for the Process Section of ELINT and ESM Radar Identification Systems

Ali Naseri

Department of Information and Communications Technology, Imam Hossein University, Tehran, Iran

ABSTRACT

Processing unit of radar detection systems such as ESM, ELINT, and RWR has a key role in the performance of these systems. Many researches have been done in this area during recent years and many algorithms have been proposed for solving its problems. In this paper, an algorithm is proposed which has two stages, pulse detection and isolation - identification of radars. In first stage, the algorithm can detect the pulses emitted by all radars which are pulsed and continuous wave radar. Pulse radars includes fixed radar pulse repetition interval, stagger radar pulse repetition interval, jitter radar pulse repetition interval, sliding radar pulse repetition interval, dwell and switch radar pulse repetition interval, periodic radar pulse repetition interval, scheduled radar pulse repetition interval, radar with capability publication pulse group, jump frequency radar, and radar with dual-frequency carrier. The second stage of the algorithm uses neural networks and separates and identifies these radars. Simulation results on 1000 sets of received signals in Persian Gulf region show that this algorithm is able to detect different radars with an accuracy of 97.02%.

Keywords: Threshold, Detection Pulse, Pulse Separation, Radar Detection, Hysteresis, False Alarm Rate, Probability of Detection, Neural Network.

Proposing A Framework for Automatic Testing of Web-Based Applications and Evaluating the Results Using Quantitative Metrics

Behshid Behkamal

Mohsen Kahani

Soheila Dehghanzadeh

Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ABSTRACT

According to the rapid growth of web-based applications, the quality of these systems needs more consideration. Because of the diversity and complexity of the Web, current methods of testing software systems are not efficient. The aim of this paper is proposing a compatible framework to the properties of web-based applications for testing them. At first, studying web-based applications and their requirements, a framework is proposed for automatic testing of web-based applications. Then based on software quality metrics, some metrics will be offered to analyze the results. Finally, proposed framework will be applied for automatic testing of three sample web-based applications and the results are evaluated by offered metrics.

Keywords: Software Testing, Metrics, Web-Based Applications, Automating Testing.

Research Areas in Agent Oriented Software Engineering (AOSE)

Ahmad Abdollahzadeh Barfourosh

Shiva Vafadar

Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

ABSTRACT

Agent Oriented Software Engineering (AOSE) is considered as the next paradigm of software engineering (after Object Oriented). In this paper, we review and analyze AOSE research area in its first decade. In this regard, we follow a systematic review approach and present various research areas in AOSE field, namely Process and Methodologies, Modeling and Tools. Via introducing briefly the different works in each area, we analyze the trends in the field and introduce the open problems that can be the subject of further research.

Keywords: Software Engineering, Agent, Agent Oriented Software Engineering (AOSE).

Proposing A New Model of Fuzzy Particle Swarm Optimization (FPSO) with Increasing the Accuracy and Veracity of Convergence for Functions Optimization

Mohammad Fiuzy¹ Javad Haddadnia¹
Seyed Kamaledin Mousavi Mashhadi² Mohammad Alipour³

¹Department of Electrical and Computer Engineering, Hakim Sabzevari University, Sabzevar, Iran

²Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran

³Department of Biomedical Engineering, Chalmers University of Technology, Chalmers, Sweden

ABSTRACT

Particle Swarm Optimization algorithm (PSO¹), searches in the problem space with path planning of particle's population based on best previous performance information of any particle (P_best) and best previous performance information of the neighbors for each particle (g_best, N_best). In this way, each particle tries to be convergent to previous global best or previous personal best that on those area, offers some point (P_best or g_best), due to this process the optimal point for objective function determined. Falls in the local optimum trap is the main problem of standard PSO algorithm. Set and control the parameters for each PSO have effective and important role in convergence and reaching answer in many problems by wide domain or variables. In this paper by using fuzzy logic, we presented a new method for control and determine the correct PSO parameter such as W, C₁ and C₂. Each particle in any iteration, used W, C₁ and C₂ values which determined by fuzzy controller based the inputs such as normalized the iterative process, difference between global and local optimal points per iteration and NCBPE parameter to obtain the next position for each particle. The proposed algorithm for solves a few issue, compared with PSO, CPSO, CFPSO then results presented. The results of experiments, indicates the superiority of the proposed method.

Keywords: Particle Swarm Optimization, Fuzzy Logic, Local Optimum, Global Optimum, Convergence.

Constrained Portfolio Optimization with Neural Network on the Grid

Ali Ghaffari Nejad Mohammad Kazem Akbari Farhang Nosrat Makuee

Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran

ABSTRACT

There are several efficient algorithms which are designed to present optimal solution for solving standard portfolio problem based on risk-variance introduced by Markowitz. Adding some constraints to the standard model is difficult and makes the new algorithms inefficient and new algorithms must be introduced. In this paper, we solve the problem of constrained portfolio optimization using neural network, The constraints we consider are limits on the proportion of the portfolio held in a given asset and limit on the number of assets. The other limit is that the total weight of assets must be equal one. This paper presents a new method, a constraint satisfaction neural network model, to have an optimized solution for portfolio problem. We use grid computing infrastructure for running the optimization algorithm fast. The constraints which were included in the model are limits on the proportion of the portfolio held in a given asset and limit on the number of assets to a known value. The results of implementation showed that the new model is able to solve the portfolio problem efficiently and properly.

Keywords: Optimization, Portfolio, Neural Network, Efficient Frontier, Grid Computing.

Extending the Constraint Solving Method for Analysis of Security Protocols with Commutative Encryption

Saeed Jalili

Sayyed Mahdi Sajjadi

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

ABSTRACT

We extend the constraint solving method by incorporating two new abilities in order to analyze security protocols. The first ability is: cryptographic primitives having lower degree of abstraction than the public or symmetric key encryption. This ability increases the expressive power of the method for the specification of security protocols. The second ability is: using a special variable in the specification of security properties. We prove that the extended constraint solving algorithm is correct and terminating. Moreover, we implemented the proposed method in *TOY-CS* tool. We verify the Shamir-Rivest-Adleman security protocol by *TOY-CS* and find four attacks against this protocol. To the best of our knowledge, from these attacks, two of them are reported for the first time.

Keywords: Constraint Solving, Algebraic Operation, Intruder Model, Verification of Security Protocols.

The CSI Journal on Computer Science and Engineering

Vol. 10

No. 1 & 3 (a)

2012

ABSTRACTS

- **Extending the Constraint Solving Method for Analysis of Security Protocols with Commutative Encryption** 1
Saeed Jalili and Sayyed Mahdi Sajjadi
- **Constrained Portfolio Optimization with Neural Network on the Grid** 2
Ali Ghaffari Nejad, Mohammad Kazem Akbari and Farhang Nosrat Makuee
- **Proposing A New Model of Fuzzy Particle Swarm Optimization (FPSO) with Increasing the Accuracy and Veracity of Convergence for Functions Optimization** 3
Mohammad Fiuzy, Javad Haddadnia, Seyed Kamaledin Mousavi Mashhadi and Mohammad Alipour
- **Research Areas in Agent Oriented Software Engineering (AOSE)** 4
Ahmad Abdollahzadeh Barforoush and Shiva Vafadar
- **Proposing A Framework for Automatic Testing of Web-Based Applications and Evaluating the Results Using Quantitative Metrics** 5
Behshid Behkamal, Mohsen Kahani and Soheila Dehghanzadeh
- **Proposing An Algorithm for the Process Section of ELINT and ESM Radar Identification Systems** 6
Ali Naseri
- **Performance Criteria Evaluation for Network Computation Management by Mobile Agents** 7
Sayyed MohammadReza Mirzababaei, Mohammad Kazem Akbari and Mahmoud Sadeghi

The CSI Journal on Computer Science and Engineering

A quarterly publication of Computer Society of Iran (CSI)

Editor-in-Chief

H. Sarbazi-Azad, Professor, Sharif University of Technology, and IPM, Tehran, Iran.

Editorial Board

G. Agha, Professor, University of Illinois at Urbana-Champaign, USA.
H. Arabnia, Professor, University of Georgia, USA.
F. Arbab, Professor, CWI and Leiden University, The Netherlands.
K. Badie, Associate Professor, Iran Telecommunication Research Center, Iran.
N. Bagherzadeh, Professor, University of California at Irvine, USA.
B. Bose, Professor, Oregon State University, USA.
A. Edalat, Professor, Imperial College, UK.
M. Fathi, Associate Professor, Iran University of Science and Tech., Iran.
M. H. Ghassemian, Professor, Tarbiat Modarres University, Iran.
M. Ghodsi, Professor, Sharif University of Technology, Iran.
A. R. Hurson, Professor, Pennsylvania State University, USA.
F. Jahanian, Professor, University of Michigan, USA.
E. Kabir, Professor, Tarbiat Modarres University, Iran.
F. C. M. Lau, Professor, University of Hong Kong, Hong Kong.
A. Movaghar, Professor, Sharif University of Technology, Iran.
N. Mahdavi-Amiri, Professor, Sharif University of Technology, Iran.
R. Meybodi, Professor, Amirkabir University of Technology, Iran.
K. Nakano, Professor, Hiroshima University, Japan.
M. Ould-Khaoua, Professor, University of Glasgow, UK.
B. Parhami, Professor, University of California at Santa Barbara, USA.
R. Safabakhsh, Professor, Amirkabir University of Technology, Iran.
H. Sarbazi-Azad, Professor, Sharif University of Technology, and IPM, Iran.
B. Shirazi, Professor, Washington State University, USA.
A. Zomaya, Professor, the University of Sydney, Australia

Assistants

M. Arjomand (Editorial Assistant)
S. Shahnazi (Publication Assistant)
L. Nourani (Publication Assistant)
A. Tavakkol (Webmaster)

Disclaimer: Publication of papers in CSI-JCSE does not imply that the editorial board, reviewers, or CSI-JCSE accept, approve or endorse the data and conclusions of authors.