

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، استاد 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

مرجان اسدی نیا
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:

فهرست مقالات

- یادداشت سردبیر ۱
حمید سربازی آزاد
- بررسی و طبقه‌بندی زبان‌های برنامه‌نویسی جنبه‌گرا و تکنیک‌های کاوش جنبه ۲
الهه حبیبی و عباس حیدرنوری
- ارائه ابزار شبیه‌ساز ابر داده‌ای بر پایه کلاسیم با قابلیت مدیریت داده و کپی ۱۶
سعیده مه‌ری، رضا انتظاری ملکی و علی موقر
- چارچوبی راهنما و پیشگیرانه، جهت برپائی ثمربخش و کم‌آسیب دوره‌های پسارشته‌ای کارشناسی ارشد علوم و مهندسی رایانش در ایران ۳۰
سید ابراهیم ابطحی
- آتوماتای بوخی رکوردی احتمالاتی: معناشناسی سیستم‌های هماهنگی با کانال‌هایی غیرقطعی ۴۰
هاجر عابدی و محمد ایزدی
- طراحی یک سیستم شناسایی چهره برخط با استفاده از پردازنده‌های گرافیکی ۵۹
محسن بیگلری، فائزه میرزائی و حسین ابراهیم‌پور کومله
- معرفی الگوریتم بهبودیافته رقابت استعماری براساس رگرسیون چندمتغیره به کمک منطق فازی ۶۷
محمد فیوضی، سید کمال‌الدین موسوی مشهدی و جواد حدادنیا

یادداشت سردبیر

انتشار فصلی مجله علوم و مهندسی کامپیوتر انجمن کامپیوتر ایران بصورت دو زبانه (دو شماره فارسی و دو شماره لاتین) از سال ۱۳۸۲ با سردبیری آقای دکتر رضا صفابخش استاد دانشگاه صنعتی امیرکبیر آغاز شد. بیشتر مشکلات اولیه نشر این مجله مربوط به معضلات مادی بود که کم‌کم با تلاش‌های رئیس انجمن رفع شد. پس از آن مهمترین مشکلی که عدم انتشار منظم مجله را باعث می‌شد داوری بسیار زمانبر مقالات بود. این مشکل متأسفانه سبب شد تا انتشار شماره‌های مجله در موعد مقرر خود انجام نشود. به محض اینکه سردبیری مجله توسط هیأت مدیره انجمن در سال ۱۳۸۴ بر دوش بنده نهاده شد، برنامه‌ریزی جهت پوشش عقب‌افتادگی شمارگان مجله آغاز شد و با برنامه و تلاش منسجم اعضای هیأت تحریریه توانستیم انتشار شماره‌های فارسی مجله را به روز کنیم. اما متأسفانه شماره‌های لاتین هنوز با تاخیر چاپ در موعد مقرر روبروست. از آنجا که هیأت تحریریه مجله و هیأت مدیره انجمن با هر گونه تسهیل در پذیرش مقالات و سبکتر کردن روند داوری مقالات مخالف بود (روشی که بسیاری از مجلات دیگر کشور را در نظم چاپ شماره‌های آنها یاری داد)، تعداد مقالات ارسالی به بخش لاتین به مراتب کمتر از مقالات ارسالی فارسی بوده، و داوری مقالات لاتین بطور معمول کندتر از مقالات فارسی انجام می‌شود، هیأت مدیره انجمن در یکی از جلسات اخیر خود تصمیم به تفکیک شماره‌های فارسی و لاتین مجله و انتشار آنها در دو مجله متفاوت گرفت. بدین ترتیب که شماره‌های مربوط به مقالات فارسی که هم‌اکنون به روز منتشر می‌شوند در مجله‌ای با عنوان *علوم رایانش و فناوری اطلاعات* انتشار یافته و مقالات انگلیسی در همین مجله فعلی *Journal on Computer Science and Engineering* بصورت فقط لاتین منتشر شوند. سردبیر مجله جدید فارسی *علوم رایانش و فناوری اطلاعات* آقای دکتر احمد خونساری دانشیار دانشگاه تهران خواهند بود که انتشار آن را از سال ۱۳۹۴ آغاز می‌کنند.

خوشبختانه انتشار مجله لاتین نیز بزودی به روز خواهد شد و انشالله انتشار آن با سردبیری یکی دیگر از همکاران هیأت علمی به طور منظم ادامه پیدا خواهد کرد. در طول این ده سال گذشته بنده فرصت تشکر از کسانی که در انتشار مجله بنده را یاری کردند را پیدا نکردم. در اینجا مایلم از همکاری اعضای محترم هیأت تحریریه مجله، داوران عزیز که بار اصلی ارزیابی مقالات را بر دوش داشتند، و همکاران مشغول در دفتر مجله که تهیه نسخه آماده چاپ مجله بدون تلاش مجدانه ایشان ممکن نبود تشکر کنم. ضمناً از هیأت مدیره انجمن که در مقاطع زمانی مختلف پشتیبانی لازم را از مجله داشتند، بخصوص رئیس انجمن آقای دکتر جعفر حبیبی، سپاسگزارم.

نهایی شدن شماره لاتین مجلد دهم مجله همراه شد با غم جانسوز از دست دادن دوست و همکار عزیز ما در دانشگاه تهران آقای دکتر سید مهدی فخرائی. مقاله آغازین این شماره یکی از آخرین انتشارات این عزیز سفر کرده است. ایشان همیشه و در همه حال از یاری‌کنندگان بی‌ادعای انجمن و مجله بودند و خدمات بسیاری در طول عمر پربرکت خود به جامعه علمی کشور داشتند. دوستان و همکاران ایشان در انجمن کامپیوتر ایران این ضایعه بزرگ را به خانواده بزرگ جامعه دانشگاهی کشور تسلیت گفته و برای ایشان مغفرت و برای خانواده گرامی ایشان صبر مسئلت می‌کند.

حمید سربازی آزاد

سردبیر

بررسی و طبقه‌بندی زبان‌های برنامه‌نویسی جنبه‌گرا و تکنیک‌های کاوش جنبه

الهه حبیبی عباس حیدرنوری

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

تولید نرم‌افزار به شیوه جنبه‌گرا^۱، رویکردی نوین برای مدیریت پیچیدگی، از طریق پیمانه‌ای نمودن کد برنامه در سیستم‌های نرم‌افزاری بزرگ است. زبان‌های برنامه‌نویسی جنبه‌گرا این امکان را می‌دهند تا کد مورد نیاز برای پیاده‌سازی یک وظیفه‌مندی^۲ خاص که در نقاط مختلف برنامه پراکنده شده‌اند را در قالب یک جنبه، پیمانه‌ای^۳ کنیم. زبان‌ها و ابزارهای متفاوتی برای برنامه‌نویسی جنبه‌گرا وجود دارد که هر کدام دارای قواعد دستوری و معنایی خاص خود می‌باشند. در این مقاله برآنیم تا ضمن بررسی مفاهیم پایه‌ای زبان‌های برنامه‌نویسی جنبه‌گرا، انواع رویکردها و زبان‌های موجود در این زمینه را مورد مطالعه و ارزیابی قرار دهیم. علاوه بر این، یکی از مسائل مهمی که در برنامه‌نویسی جنبه‌گرا وجود دارد، تعیین جنبه‌های سیستم (کاوش جنبه^۴) در فازهای مختلف فرآیند ایجاد نرم‌افزار است. تاکنون رویکردهای متفاوتی در این زمینه ارائه گردیده است. در این مقاله، این رویکردها به همراه ابزارهای مربوطه را مورد بررسی و تحلیل قرار داده‌ایم.

کلمات کلیدی: جنبه، برنامه‌نویسی، جنبه‌گرا، کاوش جنبه، زبان‌های برنامه‌نویسی، پیمانه‌ای نمودن.

۱- مقدمه

ابتدایی تولید نرم‌افزار مانند فازهای شناسایی نیازمندی‌ها و یا تحلیل سیستم نیز می‌توان، برای جداسازی و پیمانه‌ای نمودن دامنه مسئله، از آن بهره گرفت [۳، ۴، ۵]. بدین ترتیب رویکردهای متفاوتی در زمینه برنامه‌نویسی جنبه‌گرا وجود دارد. به طور کلی برنامه‌نویسی جنبه‌گرا را می‌توان به دو شاخه مفاهیم زبان برنامه‌نویسی و کاوش جنبه‌ها تقسیم نمود. در پیاده‌سازی یک برنامه جنبه‌گرا علاوه بر اهمیت وجود زبان‌ها و ابزارهای مناسب جهت برنامه‌نویسی، تکامل آن برنامه نیز از جایگاه ویژه‌ای برخوردار خواهد بود.

در این مقاله برآنیم تا مروری بر زبان‌ها، چارچوب‌ها^۵، و رویکردهای جنبه‌گرا داشته باشیم و آن‌ها را با یکدیگر مقایسه نماییم. مسائلی که در تکامل^۶ سیستم‌های جنبه‌گرا وجود دارند و همچنین روش‌های شناسایی و استخراج جنبه‌ها که در متون مهندسی نرم‌افزار به عنوان روش‌های کاوش جنبه نام برده می‌شوند، از جمله موارد دیگری می‌باشند که در این مقاله مورد بررسی قرار خواهیم داد.

در ادامه این مقاله، بخش ۲، به معرفی برنامه‌نویسی جنبه‌گرا و مفاهیم پایه‌ای آن می‌پردازد. سپس، بخش ۳، مروری بر زبان‌های برنامه‌نویسی جنبه‌گرا دارد. در

در اغلب سیستم‌های نرم‌افزاری، کدی که یک وظیفه‌مندی خاص را پیاده‌سازی می‌کند، در کل کد برنامه پراکنده^۵ شده و با کدی که دیگر وظیفه‌مندی‌ها را پیاده‌سازی می‌نماید، در هم تنیده^۶ شده است. به این نوع کد، در متون مهندسی نرم‌افزار، به عنوان دغدغه‌های سراسری^۷ نام برده می‌شود و اغلب، موجب کاهش پیمانه‌ای بودن سیستم می‌گردد [۱]. برای حل این مسئله در تولید نرم‌افزار، مدت‌هاست که مهندسين نرم‌افزار از قانون جداسازی مسائل^۸ استفاده می‌کنند و اغلب زبان‌های برنامه‌نویسی نیز، به صورت صریح (اما محدود) از این قانون پشتیبانی می‌نمایند. ولیکن، مشکل دغدغه‌های سراسری همچنان در اغلب سیستم‌های نرم‌افزاری وجود دارد. بنابراین، برای بهبود در جداسازی مسائل، G. Kickzales و همکارانش، برنامه‌نویسی جنبه‌گرا را معرفی نمودند [۱]. روش برنامه‌نویسی جنبه‌گرا در کنار دیگر زبان‌های برنامه‌نویسی، موجب افزایش پیمانه‌ای بودن تصمیمات طراحی و در نتیجه، سیستم‌های نرم‌افزاری می‌شود [۲]. تکنیک جنبه‌گرا، منحصر به فاز طراحی و تصمیمات آن نیست، بلکه از همان مراحل

جنبه‌گرا بوده است. اما بهره‌گیری از این شیوه، معایبی را نیز به همراه داشته است، مانند نیاز به استفاده از ابزارهای برنامه‌نویسی جنبه‌گرا، و ناسازگاری برخی از ابزارهای جنبه‌گرا با سیستم‌های موجود. در بخش ۴، مثال‌های بیشتری در زمینه استفاده از روش‌های برنامه‌نویسی جنبه‌گرا در صنعت و همچنین در حوزه آکادمیک آورده شده است.

ادامه، بخش ۴، ابزارها و چارچوب‌هایی که برای برنامه‌نویسی جنبه‌گرا وجود دارند را معرفی می‌نماید. بخش ۵، رویکردهای جنبه‌گرا در توسعه سیستم‌های نرم‌افزاری را مورد بررسی قرار می‌دهد. سپس، بخش ۶، در زمینه تکامل سیستم‌های نرم‌افزاری جنبه‌گرا بحث می‌کند. در ادامه، بخش ۷، روش‌های کاوش جنبه‌گرا را ارائه می‌نماید. در نهایت، بخش ۸، به نتیجه‌گیری می‌پردازد.

۲-۱- مفهوم جنبه در برنامه‌نویسی جنبه‌گرا

همانطور که در قبل نیز بیان شد، برنامه‌نویسی به روش جنبه‌گرا، با ایجاد مفهومی به نام جنبه، مشکل دغدغه‌های سراسری را برطرف ساخته است. به این ترتیب که دغدغه‌های سراسری در کد اصلی برنامه، به کدهای جنبه تبدیل می‌شوند. در واقع می‌توان گفت، جنبه به عنوان یک عنصر مستقل در کد خواهد بود که به راحتی می‌تواند به کد اضافه و یا از آن حذف شود [۷، ۱۲، ۱۳]. مقاله [۴] با نگاهی متفاوت، جنبه را مفهومی می‌داند که از همان ابتدای فرآیند تولید نرم‌افزار باید به آن پرداخته شود، تا در مراحل نهایی تولید، منجر به پیچیدگی کد و پراکندگی در آن نشود.

مقاله [۱۱] جنبه را به دو نوع همگن و ناهمگن تقسیم کرده است. جنبه همگن، جنبه‌ای است که بر مکان‌های مشخصی اعمال می‌شود، در حالی که جنبه ناهمگن ممکن است بر روی چندین مکان مختلف با رفتارهای متفاوت در هر مکان، اعمال گردد. ثبت وقایع و ردیابی^{۱۵} از جنبه‌های همگن می‌باشند و تصمیم‌گیری میان دو پروتکل برای هر بار انتقال اطلاعات در شبکه جنبه‌ی ناهمگن به شمار می‌رود، زیرا در هر بار انتقال اطلاعات، ممکن است پروتکل‌های متفاوتی (در مقایسه با انتقال قبلی)، وجود داشته باشد که البته نوع اطلاعات انتقالی نیز در ناهمگن بودن جنبه بی‌تأثیر نخواهد بود.

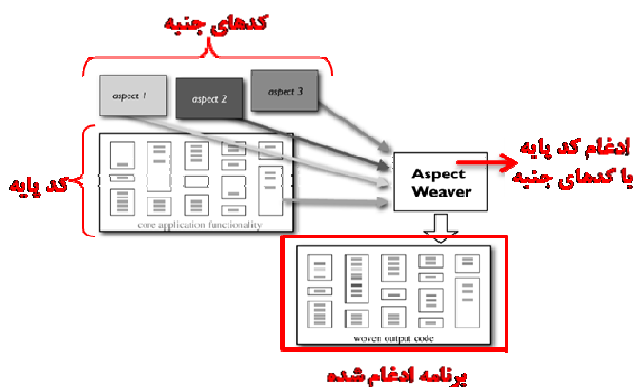
۲- معرفی برنامه‌نویسی جنبه‌گرا

در مهندسی نرم‌افزار، روش‌های متفاوتی برای پیمانه‌ای نمودن سیستم‌های نرم‌افزاری ارائه شده است. یکی از شناخته‌شده‌ترین این روش‌ها، روش برنامه‌نویسی شی‌گرا می‌باشد. هدف برنامه‌نویسی شی‌گرا، نمایش دنیای واقعی به کمک اشیا می‌باشد. اما در برخی مواقع به دلیل وجود مسائلی در طراحی و همچنین پیچیدگی دامنه، پیمانه‌ای بودن سیستم حفظ نمی‌شود [۶، ۱۷]. ویژگی اصلی این مسائل، که با نام دغدغه‌های سراسری شناخته می‌شوند، وجود تکراری آن‌ها در چندین واحد^{۱۱} برنامه است؛ به این معنا که در سراسر کد، پراکنده و در درون آن‌ها تکرار شده‌اند. بدین ترتیب دغدغه‌های سراسری، به هنگام تغییرات در کد برنامه، می‌توانند منبع بروز مشکلات در سیستم نرم‌افزاری شوند. بدین ترتیب، تأثیراتی منفی بر قابلیت نگهداری و تکامل نرم‌افزار خواهند داشت. برنامه‌نویسی جنبه‌گرا با فراهم آوردن یک مکان مشترک برای قرارگیری و پیاده‌سازی این مسائل، سعی در حل این مشکلات دارد. این مکان مشترک، یک واحد در کد به نام جنبه می‌باشد که ضمن افزایش میزان پیمانه‌ای بودن کد، صحت عملکرد برنامه را نیز حفظ می‌کند [۱، ۶، ۷، ۸]. لازم به ذکر است که کد جنبه، دارای قابلیت استفاده مجدد نیز می‌باشد. مسائلی مانند ثبت وقایع^{۱۲} یا مدیریت خطاها نمونه‌هایی از دغدغه‌های سراسری هستند.

تاکنون برنامه‌نویسی جنبه‌گرا را بر روی سیستم‌های بسیاری اعمال کرده‌اند. به عنوان مثال، در مقاله [۹]، برنامه‌نویسی جنبه‌گرا بر روی ابزار مدیریت کارایی مادون‌فرم^{۱۳} اعمال شده است. نویسندگان این مقاله، مسئله مانیتور کردن کارایی برنامه‌های سازمانی را یک دغدغه سراسری می‌دانند و همین امر را دلیلی بر اعمال شیوه جنبه‌گرا بر ابزار مدیریت کارایی بیان داشته‌اند. در این مقاله ضمن اعمال روش جنبه‌گرا، مقایسه‌ای نیز میان این شیوه و روش‌های قبلی انجام گرفته است. نتیجه این مقایسه، برتری روش جنبه‌گرا، به دلیل انعطاف‌پذیری بالاتر بوده است. در مقاله [۱۰] نیز، با استفاده از برنامه‌نویسی جنبه‌گرا به بهبود کارایی برنامه‌هایی که با پایگاه داده‌ها کار می‌کنند، پرداخته است. در این مقاله به‌روز شدن تحلیل‌های آماری را یک دغدغه سراسری معرفی می‌نماید.

این تحلیل‌های آماری برای بهینه‌سازی پرس‌وجوها^{۱۴} در پایگاه داده مفید می‌باشند. زمان‌بر بودن محاسبات آماری و حجیم بودن داده‌های مربوط به جدول‌های مختلف از جمله مشکلات بیان شده در این مقاله‌اند، که با برنامه‌نویسی جنبه‌گرا سعی در برطرف نمودن آن‌ها دارد. با وجود مشکلاتی مانند برپایی محیط مناسب برای اعمال شیوه جنبه‌گرا، استفاده از این شیوه موجب افزایش کارایی (تا ۲۵ درصد) در مقایسه با روش‌های دیگر بوده است.

استفاده از برنامه‌نویسی جنبه‌گرا در حوزه آکادمیک در مقایسه با صنعت، پیشرفت ملموس‌تری داشته است و مثال‌های بیشتری نیز در این حوزه یافت می‌شود. بدین ترتیب، برخی افراد آکادمیک تلاش کرده‌اند تا برنامه‌نویسی جنبه‌گرا را بیشتر به حوزه صنعت وارد نمایند. در مقاله [۱۱]، به بررسی مزایای استفاده از شیوه جنبه‌گرا در صنعت و برای نرم‌افزارهای تجاری پرداخته است. برای دستیابی به این هدف، مصاحبه‌ای با ۱۱ نفر از ۵ کشور مختلف، در زمینه استفاده از برنامه‌نویسی جنبه‌گرا انجام گرفته است. جمع‌بندی این مصاحبه‌ها، ذکر مزایایی مانند خوانایی کد، کاهش پیچیدگی و کاهش خطاها برای روش برنامه‌نویسی



شکل ۱- نحوه ادغام جنبه‌ها با کد پایه به وسیله Aspect Weaver [۶]

۲-۲- مفاهیم برنامه‌نویسی جنبه‌گرا

سه فاز مهم در برنامه‌نویسی جنبه‌گرا وجود دارد: کشف جنبه‌ها، استخراج جنبه‌ها، و تکامل جنبه‌ها. "آیا یک سیستم نرم‌افزاری، نیاز به برنامه‌نویسی جنبه‌گرا دارد؟ آیا دغدغه‌های سراسری در این نرم‌افزار وجود دارند؟" این دو مسئله، از سؤالاتی هستند که در فاز اول (کشف جنبه) مطرح می‌شوند. در طول این فاز، کاندیداهای مناسب برای جنبه شناسایی می‌گردند. در این فاز همچنین، چگونگی پیاده‌سازی جنبه و تأثیر آن بر کیفیت نرم‌افزار مورد بررسی قرار می‌گیرد. در فاز استخراج، که به معنای جداسازی کدهای دغدغه‌های سراسری از کد اصلی است، پس از شناسایی جنبه‌ها در فاز اول، سیستم به سمت برنامه‌نویسی جنبه‌گرا حرکت می‌کند. با تبدیل کاندیدها به جنبه، رفتار سیستم بار دیگر بررسی می‌شود تا از

برنامه اصلی ادغام گردد. نمونه‌ای از این کد، در شکل ۳ آورده شده است. در این شکل، Pointcut متدهایی را که عمل refresh بر روی آن‌ها اعمال می‌شود، مشخص می‌نماید. سپس، after advice کد اصلی مربوط به جنبه (متد refresh) را فراهم می‌کند.

۳- زبان‌های برنامه‌نویسی جنبه‌گرا

برای پشتیبانی از برنامه‌نویسی جنبه‌گرا، لازم است تا زبان‌های برنامه‌نویسی کنونی، گسترش یابند. برنامه‌نویسی جنبه‌گرا نیز مانند هر روش دیگر برنامه‌نویسی، دارای قواعد دستوری و معنایی با نمادهای خاص هر زبان و ابزارهای متفاوت برای پیاده‌سازی آن زبان‌هاست. زبان‌های برنامه‌نویسی جنبه‌گرا را می‌توان به دو دسته خاص جنبه‌گرا و مستقل از جنبه‌گرا تقسیم‌بندی نمود. در ادامه به بررسی چند نمونه از این زبان‌ها می‌پردازیم.

۳-۱-۱- زبان‌های برنامه‌نویسی خاص جنبه‌گرا

این گونه زبان‌های جنبه‌گرا، گسترشی^{۱۶} بر یک زبان برنامه‌نویسی خاص هستند، به این معنا که با قواعد دستوری همان زبان، سازگار می‌باشند و نمی‌توان از قواعد دستوری و معنایی این زبان جنبه‌گرا، در یک زبان دیگر استفاده نمود. در ادامه به دو نمونه از این زبان‌ها اشاره می‌کنیم.

۳-۱-۱-۱- زبان برنامه‌نویسی AspectJ

زبان برنامه‌نویسی AspectJ، که معروف‌ترین زبان جنبه‌گرا می‌باشد، گسترشی بر زبان جاوا است [۵، ۸]. این زبان که در ابتدا در مرکز تحقیقاتی Xerox PARC ایجاد شده است، اکنون به عنوان بخشی از پروژه متن باز Eclipse^{۱۷} می‌باشد [۸، ۲].

Join point در این زبان شامل فراخوانی متد، فراخوانی سازنده^{۱۸}، اجرای فراخوانی متد، اجرای فراخوانی سازنده، فیلدهای set و get، اجرای مدیریت خطاها و آماده‌سازی کلاس و شی، می‌باشد؛ اما حلقه‌ها و عبارات شرطی if، در این زبان پشتیبانی نمی‌شوند. نماد "*"، در این زبان به معنای "همه" می‌باشد. به عنوان مثال، عبارت call(**BankAccount:*(*)). تمامی فراخوانی‌های روی BankAccount را که دارای یک پارامتر می‌باشند، برمی‌گردانند. نماد "+" برای نشان دادن subtype کاربرد دارد. به عنوان مثال throws Exception، تمامی خطاهایی که از نوع Exception هستند را throw می‌کند. Pointcut در زبان AspectJ به صورت زیر نوشته می‌شود:

در جلوی این عبارت،
join point گذاشته می‌شود
نام pointcut
pointcut pointcutName ([parameters]):designator(joinpoint)

زبان AspectJ، از Advice‌های نوع after، before، و around پشتیبانی می‌کند. جنبه‌ها در AspectJ مانند کلاس در زبان جاوا تعریف می‌شوند و دارای فیلد و متد می‌باشند [۷، ۸]. بنابراین برای استفاده از این زبان نیاز است تا برنامه‌نویس یک فایل کلاس جاوا ایجاد کند و جنبه را در آن بنویسد. سپس کد اصلی برنامه با کد جنبه، از طریق کامپایلر AspectJ با یکدیگر ادغام می‌شوند [۸]. مدیریت خطاها، مکانیسم سودمندی است که زبان AspectJ فراهم می‌آورد [۷].

صحت عملکرد سیستم، اطمینان حاصل گردد. اما نرم‌افزاری که با برنامه‌نویسی جنبه‌گرا ایجاد شده است، مانند هر نرم‌افزار دیگری نیاز به تکامل دارد. این مسئله در فاز سوم، مورد بررسی قرار می‌گیرد [۶].

همانطور که در بخش قبلی نیز بیان شد، هدف برنامه‌نویسی جنبه‌گرا، پیمانه‌ای نمودن سیستم، با استفاده از مفهوم انتزاعی جنبه می‌باشد. این جنبه‌ها به صورتی مستقل از برنامه اصلی تعریف می‌شوند و به وسیله ابزاری به نام Aspect Weaver با کد پایه، ادغام می‌گردند. این ابزار، همانند یک کامپایلر عمل کرده و می‌تواند عمل ادغام را در زمان کامپایل و یا در زمان اجرا انجام دهد [۶، ۸]. شکل ۱، عملکرد این ابزار را توصیف می‌نماید.

در زبان‌های برنامه‌نویسی جنبه‌گرا، سه مفهوم مهم و کلیدی وجود دارد: Join point، Pointcut، و Advice. مکانی که قرار است تا کد جنبه به برنامه اصلی متصل گردد، Join point نامیده می‌شود. مجموعه‌ای از این مکان‌ها (Join point ها)، Pointcut می‌باشد [۱، ۶، ۸]. لازم به ذکر است که اگر برنامه‌ای دارای تعداد کمی از Join point ها باشد، کاندیدای مناسبی برای برنامه‌نویسی جنبه‌گرا نیست. Advice ها، کدهای مربوط به دغدغه‌های سراسری بوده و قرار است به کدهای جنبه‌ها تبدیل شوند.

با استفاده از Aspect Weaver، این کدها به سه طریق مختلف می‌توانند در کد پایه برنامه ادغام شوند: اگر کد Advice قبل از Join point اضافه شود، before advice؛ اگر کد Advice بعد از Join point اضافه شود، after advice؛ و اگر کد Advice دقیقاً در محل Join point به کد اصلی برنامه اضافه شود، around advice نامیده می‌شود.

به عنوان مثال، شکل ۲، قسمت‌هایی از یک برنامه جاوا را نشان می‌دهد که برای ترسیم اشکال مختلف بر روی صفحه نمایش مورد استفاده قرار می‌گیرد. اما، قبل از ترسیم هر شکل جدید بر روی صفحه، بایستی که صفحه نمایش پاک و آماده شود. برای اجتناب از تکرار کد پاک کردن صفحه نمایش (متد refresh)، قبل از ترسیم هر شکل جدید، می‌توان این متد را به صورت یک جنبه تعریف نمود.

```
Public class Shape {...}
Public class Rectangle extends Shape {
    ...
    Void set Left Corner (corner: Point) {...}
    Void set Right Corner (corner: Point) {...}
}
Public class Circle extends Shape {
    ...
    Void set RADIUS (length: float) {...}
}
```

شکل ۲- کلاس‌های ترسیم اشکال هندسی در صفحه نمایش

```
Public aspect Update Page {
    Pointcut Display State Change () {
        Execution (void Shape+.set*(...))
    }
    After () return Display State Change () {
        Display.refresh ();
    }
}
```

شکل ۳- کلاس جنبه، دارای متد refresh

به طوری که تنها یک بار آن کد نوشته شود و سپس با استفاده از Aspect Weaver، به صورت خودکار در مکان مورد نظر در برنامه و در زمان کامپایل با

۳-۱-۲- زبان برنامه‌نویسی AspectC++

می‌توان به AspectJ، Aspect#، ABC، AOP#، Orthogonal Weaving، Model، Aspect.NET، AspectDNG، Loom.NET، PostShar و بسویاری دیگر، اشاره داشت. اطلاعات بیشتر در این زمینه در تحقیقات ارائه شده در مقالات [۱۷] و [۱۸] وجود دارد.

زبان AspectC++ از نظر مفهومی مشابه زبان AspectJ می‌باشد. طراحان این زبان، سعی کرده‌اند تا قواعد دستوری و معنایی این زبان را با قواعد زبان C++ سازگار نمایند [۸، ۱۴، ۱۵]. نماد "*" در زبان AspectJ به نماد "/" در زبان AspectC++ تغییر یافته است، زیرا نماد "*" در زبان C++ برای اشاره‌گرها استفاده می‌شود. ولیکن، Join pointها در این زبان، همانند زبان AspectJ می‌باشند. Pointcutها نیز به صورت عباراتی مشابه با عبارات منظم^{۱۹} نوشته می‌شوند که می‌توانند با توجه به نوع خود، در زمان کامپایل و یا در زمان اجرا، ارزیابی گردند. زبان AspectC++، از Adviceهای نوع after، around و before پشتیبانی می‌نماید. همچنین، جنبه را می‌توان مانند کلاسی که دارای فیلد و متد می‌باشد، تعریف نمود [۸، ۱۴، ۱۵].

۵- رویکردهای جنبه‌گرا

موضوع مهمی که در برنامه‌نویسی جنبه‌گرا وجود دارد، تعیین جنبه‌های سیستم در فازهای مختلف فرآیند ایجاد نرم‌افزار است. برخی بر این باورند که تعیین جنبه‌های سیستم منحصر به زمان پیاده‌سازی آن نیست و در همان فازهای ابتدایی نیز می‌توان، جنبه‌های سیستم را مشخص نمود [۳]. بر این اساس، همان‌گونه که در جدول ۳ نشان داده شده است، رویکردهای متفاوتی وجود دارند که در ادامه به برخی از آن‌ها اشاره می‌شود.

۳-۲- زبان‌های برنامه‌نویسی مستقل از جنبه‌گرا

۵-۱- مدل طراحی جنبه‌گرا

مدل طراحی جنبه‌گرا، رویکردی است که در مقاله [۱۹] به آن اشاره شده است. در این رویکرد، جنبه‌ها در زمان طراحی، برای زبان برنامه‌نویسی AspectJ تعریف می‌شوند و در زمان پیاده‌سازی با استفاده از این زبان، برنامه‌نویسی می‌گردند. برای شناسایی جنبه‌ها در این رویکرد، نیاز است تا مفاهیم UML و AspectJ در کنار هم قرار داده شوند. برای این منظور، مدل‌های رفتاری و ساختاری مانند نمودار کلاس^{۲۲}، نمودار ترتیب^{۲۳} و نمودار تعاملات برای شناسایی Join pointها و جنبه‌ها، مورد استفاده قرار می‌گیرند. اگر چه مثال‌هایی از این رویکرد مانند زمان‌بندی در سیستم خطوط تلفن وجود دارد [۱۹]، اما این رویکرد هنوز به طور عملی در دنیای واقعی مورد آزمون قرار نگرفته است.

این زبان‌ها، برخلاف زبان‌های خاص، به هیچ یک از زبان‌های برنامه‌نویسی وابسته نیستند. زبان Weave.NET، نمونه‌ای از زبان‌های مستقل می‌باشد. این زبان به عنوان یک مؤلفه NET، نوشته شده است و دارای دو زیرسیستم می‌باشد: تولید کد و مدل‌سازی جنبه. زیرسیستم تولید کد، زبان اسمبلی موجود (وردی زبان Weave.NET) را به یک اسمبلی پویا تبدیل می‌کند. زیرسیستم مدل‌سازی جنبه، فایل XML که شامل توصیف جنبه می‌باشد را ترجمه نموده و جنبه را به صورت Advice و Pointcut مدل می‌کند. همچنین در این زیرسیستم، تطابق Join pointها با جنبه‌ها مورد بررسی قرار می‌گیرد [۸، ۱۶]. می‌توان گفت که زبان Weave.NET به نوعی از زبان AspectJ ارث برده شده است، به گونه‌ای که تعاریف Advice، Pointcut و Join point مشابه زبان AspectJ می‌باشد.

۵-۲- رویکرد طراحی JAC

رویکرد طراحی JAC نیز همانند رویکرد طراحی جنبه‌گرا، در زمان طراحی، در پی یافتن جنبه‌ها می‌باشد [۲۰]. در این مقاله چارچوبی متن‌باز^{۲۴} که شامل یک محیط توسعه نیز می‌باشد، برای برنامه‌نویسی جنبه‌گرا معرفی شده است. رویکرد طراحی JAC، یک رویکرد شناخته شده است که تاکنون برای یافتن جنبه‌هایی مانند احراز هویت^{۲۵}، ردیابی و جلسه^{۲۶} در برنامه‌های کاربر-کارگزار^{۲۷} استفاده شده است. این رویکرد همچنین در طراحی وبسایت دروس و ابزارهای مدیریت امور تجاری مورد آزمون قرار گرفته است.

۴- ابزارها و چارچوب‌های جنبه‌گرا

چارچوب‌های جنبه‌گرا به طراحان و برنامه‌نویسان جنبه‌گرا از طریق یک مجموعه از جنبه‌های از پیش آماده شده، کمک می‌کنند. این چارچوب‌ها می‌توانند در دو دسته چارچوب‌های زمان پیاده‌سازی (جدول ۱) و چارچوب‌های زمان جمع‌آوری نیازمندی‌ها (جدول ۲) تقسیم‌بندی شوند. دسته اول در هنگام پیاده‌سازی و یا بعد از آن در زمان تکامل نرم‌افزار کاربرد دارند. نکته قابل توجه در این زمینه این است که چارچوب‌های زمان پیاده‌سازی ارائه شده برای زبان برنامه‌نویسی جاوا، در مقایسه با زبان‌های برنامه‌نویسی دیگر، پرکاربردتر می‌باشند. دسته دوم، ابزارهایی برای شناسایی دغدغه‌های سراسری در زمان جمع‌آوری نیازمندی‌ها می‌باشند. این ابزارها، خود به دو دسته متقارن^{۲۰} و نامتقارن^{۲۱} دسته‌بندی شده‌اند [۴]. ابزارهای نامتقارن مربوط به مجموعه‌ای از دغدغه‌های سراسری خاص می‌باشند که از یک بُعد مورد توجه هستند (مانند ابزارهای AO Use cases، Theme/Doc، Arcade، و غیره). این در حالی است که با استفاده از ابزارهای متقارن می‌توان هر دغدغه سراسری را از چندین بُعد مورد بررسی قرار داد. به عنوان نمونه، جنبه امنیت از چندین بُعد قابل بررسی است. امنیت در احراز هویت کاربر برای کارهای بانکی، یا امنیت در انتقال اطلاعات در شبکه، می‌تواند به هنگام برنامه‌نویسی جنبه‌گرا، رفتارهای متفاوتی را داشته باشد. بنابراین نمی‌توان مسئله امنیت در یک سیستم بانکداری را به یک صورت مدل نمود.

۵-۳- ایجاد نرم‌افزار جنبه‌گرا با استفاده از مدل‌سازی UML

ایجاد نرم‌افزار جنبه‌گرا با استفاده از مدل‌سازی UML، رویکرد دیگری است که در [۲۱] به آن اشاره شده است. در این رویکرد، دغدغه‌های سراسری در فاز جمع‌آوری نیازمندی‌ها به کمک موارد کاربرد^{۲۸} مشخص می‌گردند. سپس، در فاز تحلیل، با کمک نمودارهای کلاس و ارتباطات^{۲۹} مورد تحلیل قرار می‌گیرند. در فاز طراحی، به کمک نمودارهای ترتیب و مؤلفه طراحی می‌شوند و در فاز پیاده‌سازی، به وسیله زبان‌های AspectJ و یا HyperJ پیاده‌سازی می‌گردند. در [۲۱]، این رویکرد را برای پیاده‌سازی سیستم مدیریت هتل اعمال کرده‌اند. به غیر از این مقاله، تاکنون کاربردی از این رویکرد در صنعت گزارش نشده است. همچنین، هیچ گونه ابزاری برای پشتیبانی از این رویکرد وجود ندارد.

علاوه بر چارچوب‌های معرفی شده در جدول ۱، چارچوب‌های بسیاری برای برنامه‌نویسی به شیوه جنبه‌گرا (در زمان پیاده‌سازی) وجود دارد، که از آن جمله

۴-۵- رویکرد رفتاری Aspect Weaver

۵-۵- رویکرد پروفایل AOSD

رویکرد رفتاری Aspect Weaver با استفاده از نمودار ترتیب پیام^{۲۰} و با کمک الگوریتم weaving (الگوریتمی برای شناسایی Join point) به شناسایی جنبه‌ها می‌پردازد [۲۲]. تمرکز کنونی این رویکرد بر بهینه ساختن الگوریتم weaving، برای سیستم‌هایی با رفتار پیچیده می‌باشد. اما این رویکرد هنوز در دنیای واقعی مورد استفاده قرار نگرفته است. ولیکن، ابزارهای مدل‌سازی برای پشتیبانی از این رویکرد رفتاری وجود دارند.

رویکرد پروفایل AOSD، رویکردی مستقل از زبان‌های برنامه‌نویسی جنبه‌گرا است [۲۳]. از نمودار کلاس به دست آمده در فاز تحلیل، یک ماشین حالت، استخراج می‌گردد؛ رهنمودهایی در این رویکرد برای شناسایی جنبه‌ها از ماشین حالت، وجود دارد. اگرچه جنبه‌هایی مانند همزمانی، زمانبندی، نامگذاری و اداره کردن خطا در سیستم بافر، از جمله مثال‌های اعمال این رویکرد می‌باشد، اما همچنان در صنعت مورد استفاده قرار نگرفته است. مکانیسم پروفایل UML، از ابزارها و روش‌های مورد استفاده در این رویکرد می‌باشد.

جدول ۱- مقایسه چارچوب‌های جنبه‌گرا برای زمان پیاده‌سازی

چارچوب‌های جنبه‌گرا	زبان برنامه‌نویسی مربوطه	زمان ادغام با کد پایه (Weaving)	توضیحات
AspectWerkz	Java (Open Source)	- زمان اجرا - زمان بار شدن کلاس‌ها - زمان ساخت	طراحان، AspectWerkz را یک چارچوب ساده، پویا و سبک وزن می‌دانند که به راحتی با کدهای زبان جاوا یکپارچه می‌شود [۸، ۹، ۱۷، ۴۶].
JBoss	Java (Open Source)	- زمان اجرا و از طریق Dynamic Proxy	چارچوبی است که منحصراً برای جاوا تعریف شده است و دارای مجموعه‌ای از جنبه‌های از پیش آماده می‌باشد [۸، ۱۷].
HyperJ	Java (Open Source)	- زمان بار شدن کلاس‌ها	این چارچوب از دغدغه‌های سراسری چند بُعدی پشتیبانی می‌نماید [۲، ۴۷].
AspectC#	تحت .NET (Open Source)	- زمان کامپایل	چارچوبی ساده و با زمان یادگیری پایین است. از دیدگاه تعریف قواعد معنایی و دستوری مانند زبان AspectJ می‌باشد [۸، ۴۸].
Spring	Java (Open Source)	- زمان اجرا و از طریق Dynamic Proxy	تمرکز این چارچوب بر نرم‌افزارهای تجاری است و با زبان AspectJ سازگار می‌باشد [۷، ۴۹].

جدول ۲- مقایسه چارچوب‌های جنبه‌گرا برای زمان جمع‌آوری نیازمندی‌ها

چارچوب‌های جنبه‌گرا	توضیحات	ابزارهای نامتعارف
Probe Framework	توصیف نیازمندی‌ها، به صورت فایل XML و یا با استفاده از نمادهای UML، ورودی این ابزار است. تولید کننده منطق و تحلیل‌گر UML، برای تجزیه ورودی‌ها می‌باشد. نمونه‌ساز و ادغام‌کننده از جمله پیمان‌های این ابزار به شمار می‌روند [۴].	ابزارهای نامتعارف
Arcade	در این ابزار واحدهای مربوط به تجزیه نیازمندی‌های اصلی، جنبه به شمار می‌روند [۴].	
Theme/Doc	این ابزار، نمادهای گرافیکی را برای توصیف نیازمندی‌ها فراهم می‌آورد و از ابزار طراحی UML/theme برای این منظور استفاده می‌کند. در این ابزار، نیازمندی‌هایی که دارای رفتاری مشابه هستند، شناسایی شده و در یک واحد عملکردی قرار می‌گیرند. به این ترتیب حجم نیازمندی‌ها کاهش یافته و بهتر می‌توان دغدغه‌های سراسری را از میان آن‌ها انتخاب نمود [۳، ۴، ۲۴].	
Scenario Modeling	با استفاده از الگوهای تعاملات میان نیازمندی‌ها، دغدغه‌های سراسری را به شکل یک سناریو در قالب نمودار تعاملات ^{۳۱} UML مدل می‌کند. این نمودار برای مدل کردن رفتار خارجی سیستم ایجاد می‌شود. نمودار انتقال حالت نیز می‌تواند برای مدل کردن رفتار داخلی سیستم تولید گردد [۴].	
AO Use cases	این ابزار گسترشی بر مدل Use case با قالب AspectJ می‌باشد. برای پیمان‌های کردن دغدغه‌های سراسری، مانند بررسی خطاها، بهبود کارایی، مانیتور کردن و ثبت وقایع، مناسب است [۴].	
Requirements Description Language (RDL)	با تحلیل زبان طبیعی که ورودی این ابزار می‌باشد، سعی در کشف دغدغه‌های سراسری دارد. این روش مشابه با روش زبان توصیف معماری است. اما زبان طبیعی دارای مشکلاتی مانند نامفهوم و مبهم بودن برخی از جملات است. بنابراین ورودی این ابزار می‌بایست دارای قالب خاصی باشد که این قالب می‌تواند تحت زبان XML به ابزار داده شود [۴].	ابزارهای متعارف
Cosmos	شمایی برای نمایش ارتباط میان دغدغه‌های سراسری می‌باشد. این ابزار دغدغه‌های سراسری را به دو دسته فیزیکی (مسائل مربوط به سخت افزار، محصولات کاری، سرویس‌ها و یا منابع) و منطقی (مانند ویژگی‌ها، عنوان‌ها، حالت‌ها، توابع و به طور کلی مسائل غیرقابل لمس) تقسیم می‌کند [۴].	
Multi-Dimensional Requirement Modelling	این ابزار گسترش یافته ابزار Arcade می‌باشد. به گونه‌ای که دغدغه‌های سراسری را از چندین بُعد مورد بررسی قرار می‌دهد [۴].	

۵-۶- رویکرد Theme

تمامی این رویکردها، تمرکز بر استفاده از مدل‌ها و نمودارهای ایجاد شده در طول این فازها بوده است. بدین ترتیب آشنایی با مدل‌ها، پیش‌نیاز استفاده از این رویکردهاست. برای حل این مسئله، مقاله [۲۶]، مدل معماری مرجع را برای برنامه‌نویسی جنبه‌گرا معرفی می‌کند. همانگونه که در شکل ۴ نمایش داده شده است، این مدل مستقل از هر زبان پیاده‌سازی و نمودارهای فازهای تحلیل و طراحی می‌باشد و همچنین یک دید کلی از شیوه جنبه‌گرا را در اختیار می‌گذارد. با استفاده از مدل معماری مرجع بسیاری از مفاهیم شیوه جنبه‌گرا، به طور کامل مشخص شده است. مزیت دیگر این معماری مرجع، نگاهت نمودارها و مدل‌های فازهای تحلیل و طراحی به این مدل است.

در شکل ۴، بخش Concern Decomposition در ارتباط با شکست سیستم به دغدغه‌های سراسری و ارتباط آن‌ها با یکدیگر است. عملیات ادغام جنبه‌ها با کد پایه در این بخش از معماری، می‌تواند در زمان طراحی یا در زمان اجرا باشد. بخش Adaptation Rule، در ارتباط با سازگاری جنبه‌ها با Pointcut ها است (آیا جنبه مربوطه در مکان مناسبی قرار گرفته است؟). در بخش Language نیز، زبان جنبه‌گرای مورد نظر انتخاب می‌شود، که این زبان، وابسته به اینکه در کدام فاز از فرآیند تولید به شناسایی جنبه‌ها پرداخته می‌شود، یک زبان مدل‌سازی و یا یک زبان برنامه‌نویسی باشد.

۵-۷- رویکرد مدل معماری جنبه‌گرا

رویکرد مدل معماری جنبه‌گرا، رویکردی مستقل از بستر^{۳۳} نرم‌افزاری می‌باشد [۲۵]. در این رویکرد، دغدغه‌های سراسری به وسیله نمودار بسته‌های نرم‌افزاری^{۳۴} و نمودار کلاس مدل می‌شوند. لازم به ذکر است که این رویکرد، به طور خاص در فازهای تولید نرم‌افزار مورد استفاده قرار گیرد. همچنین، برخلاف دیگر رویکردهای ارائه شده تاکنون که از مدل‌سازی UML برای شناسایی دغدغه‌های سراسری بهره می‌گرفتند، رویکرد مدل معماری جنبه‌گرا، از یک زبان متامدل مبتنی بر نقش^{۳۵} که توسعه‌یافته‌ای از همان مدل‌سازی UML می‌باشد، استفاده می‌کند. این رویکرد در مقایسه با رویکردهایی که تاکنون معرفی نمودیم، بیشترین کاربرد را داشته است. مسئله احراز هویت در سیستم بانکداری و کنترل انتقال پول، از جمله مثال‌های این رویکرد می‌باشند، اما تاکنون در صنعت مورد استفاده قرار نگرفته است. مجموعه‌ای از ابزارها برای کمک در مدل‌سازی این رویکرد نیز وجود دارند.

۵-۸- رویکرد مبتنی بر مهندسی نیازمندی‌ها

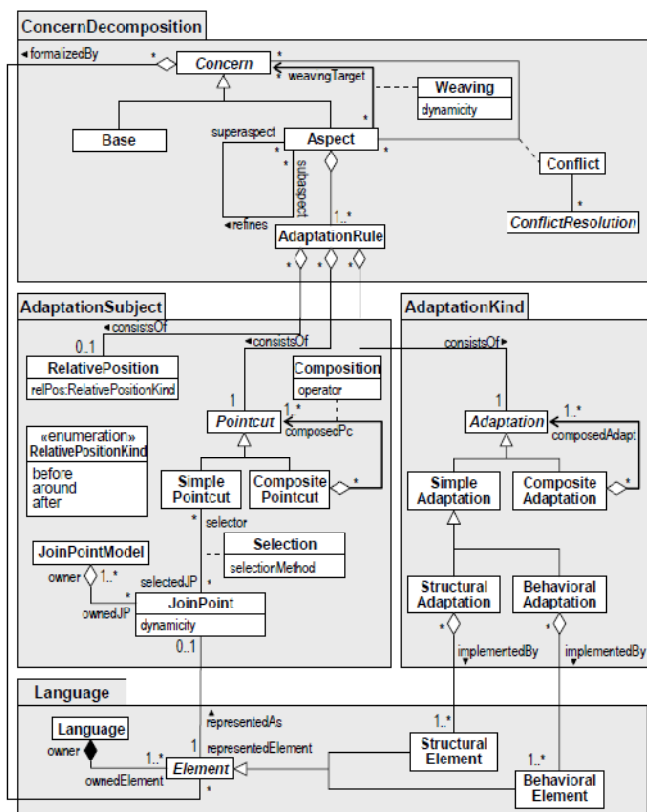
مقاله [۴]، رویکردی را مبتنی بر مهندسی نیازمندی‌ها معرفی نموده است. ایده اصلی این روش این است که دغدغه‌های سراسری از همان مراحل ابتدایی فرآیند توسعه سیستم که به جمع‌آوری نیازمندی‌ها می‌پردازد، در سیستم وجود دارند. عدم توجه به آن‌ها و همچنین به تأخیر انداختن شناسایی آن‌ها تا زمان پیاده‌سازی، می‌تواند منجر به ایجاد تناقض در میان نیازمندی‌ها شود. در این مقاله، تنها به بررسی ابزارهایی برای کمک به این ایده پرداخته و هیچ‌گونه مثالی را در این رابطه بیان نکرده است. نکته قابل توجهی که در این مقاله وجود دارد، دسته‌بندی ابزارها به دو دسته متقارن و نامتقارن است.

در مقاله‌ای مشابه با [۴]، ایده یافتن دغدغه‌های سراسری در فاز جمع‌آوری نیازمندی‌ها را با استفاده از روش‌های پردازش زبان طبیعی^{۳۶} عملی ساخته است [۵]. در این مقاله، با استفاده از ابزار WMATRIX، از طریق روش NLP (یکی از روش‌های کاوش جنبه که در بخش ۷ معرفی خواهد شد)، به تحلیل نیازمندی‌ها در یک سیستم پردازش داده‌های اطلاعاتی یک بزرگراه پرداخته است.

۵-۹- مدل معماری مرجع

برنامه‌نویسی جنبه‌گرا مانند هر روش برنامه‌نویسی دیگری، دارای مسائل و مشکلات خاص خود می‌باشد که نیازمند تحقیقات بیشتری است. در ادامه، برخی از مسائل و تحقیقاتی که تاکنون در این زمینه انجام شده است، بررسی می‌گردد.

تاکنون، چندین رویکرد برای شناسایی دغدغه‌های سراسری قبل از فاز پیاده‌سازی (مثلاً، فازهای جمع‌آوری نیازمندی‌ها، تحلیل و یا طراحی) را معرفی نمودیم. در



شکل ۴- مدل معماری مرجع [۲۶]

۶- مسائل و مشکلات بازنویسی برنامه‌ها به صورت جنبه‌گرا

جدول ۳- مقایسه رویکردهای جنبه‌گرا

رویکردهای جنبه‌گرا	زبان برنامه‌نویسی پیاده‌سازی	فاز تولید برای شناسایی جنبه و نمودار مورد استفاده	زبان مدل‌سازی	مثال از اعمال رویکرد به صورت آکادمیک	مثال از کاربرد در صنعت
رویکرد مدل طراحی جنبه‌گرا	AspectJ	فاز طراحی: CD, CoID, SD, JD و CD	UML 1.X	زمان‌بندی در سیستم خطوط تلفن	-
رویکرد طراحی JAC	JAC	فاز طراحی: CD	UML 1.X	برنامه‌های کاربر- کارگزار	سایت دروس و مدیریت امور تجاری
رویکرد ایجاد نرم افزار جنبه‌گرا با استفاده از مدل‌سازی UML	HyperJ AspectJ	فاز جمع‌آوری نیازمندی‌ها: UCD فاز تحلیل: CD و ComD فاز طراحی: SD و CPD	UML 2	مدیریت هتل	-
رویکرد رفتاری Aspect Weaver	-	فاز تحلیل: MSC	UML 2	فرآیند ثبت وقایع	-
رویکرد پروفایل AOSD	-	فاز تحلیل: CD	UML 1.X	سیستم بافر	-
رویکرد Theme	AspectWerkz HyperJ AspectJ	فازهای جمع‌آوری نیازمندی‌ها و تحلیل: CD فاز طراحی: SD و PD	UML 1.X	همزمانی در کتابخانه دیجیتال، login در سیستم مدیریت دروس	-
رویکرد مدل معماری جنبه‌گرا	-	فاز طراحی: CD و PD	UML 2, Role-Based	احراز هویت در سیستم بانکداری و کنترل انتقال پول	-
رویکردی مبنی بر مهندسی نیازمندی‌ها	-	فاز جمع‌آوری نیازمندی‌ها	سند نیازمندی‌ها (به زبان طبیعی)	خروجی ابزار جمع‌آوری اطلاعات یک بزرگراه	-
رویکرد مدل معماری مرجع	هر زبان برنامه‌نویسی	در کلیه فازهای توسعه سیستم	UML	-	-

CD: Class Diagram

CoID: Collaboration Diagram

SD: Sequence Diagram

ID: Interaction Diagram

MSC: Message Sequence Chart

UCD: Use Case Diagram

ComD: Communications Diagram

CPD: Component Diagram

PD: Package Diagram

این مقاله، با استفاده از دو روش ارائه شده، به انجام دو نوع تکامل مختلف پرداخته‌اند. در نوع اول با استفاده از روش Aspect-Aware، تکامل بر روی کد پایه انجام می‌شود و به این ترتیب، مشکلی که به هنگام اصلاحات کد پایه در یک برنامه جنبه‌گرا به وجود می‌آید، بر طرف خواهد شد. در نوع دوم تکامل به جای اعمال بر کد پایه با بهره‌گیری از روش Aspect-Oriented Refactoring، بر روی کد جنبه اعمال می‌گردد. با این عمل، تمامی کدهای جنبه، پس از تغییرات، به‌هنگام خواهند شد.

Walker و همکارانش [۲]، با استفاده از اعمال برنامه‌نویسی جنبه‌گرا بر روی دو سیستم مختلف، با اهدافی مانند راحتی در اشکال‌زدایی و راحتی در تغییرات کد برنامه، به مقایسه این روش با روش‌های دیگر می‌پردازند. در این مقاله از ابزار AspectJ برای برنامه‌نویسی جنبه‌گرا استفاده نموده است. تسریع در یافتن خطاها و تصحیح آن‌ها و همچنین افزایش کارایی سیستم، در صورت استفاده از شیوه جنبه‌گرا، از نتایج به دست آمده می‌باشد.

۶-۲- پیاده‌سازی دغدغه‌های سراسری پیچیده

Bruntink و همکارانش [۳۱]، مجموعه‌ای از قالب‌های کد آماده و ساده را برای پیاده‌سازی دغدغه‌های سراسری ارائه می‌دهند. برنامه‌نویسان این امکان را نیز دارند تا این کدها را برای سیستم خود سفارشی کنند.^{۳۸}

۶-۱- تأثیر دغدغه‌های سراسری بر کیفیت و تکامل نرم‌افزار

Bruntink و همکارانش [۲۷]، با استفاده از جنبه‌های مدیریت خطاها، تأثیر دغدغه‌های سراسری را بر کیفیت سیستم مورد تحلیل قرار داده‌اند. نتیجه به دست آمده از این بررسی حاکی از آن است که عدم استفاده از برنامه‌نویسی جنبه‌گرا، برنامه‌نویسی را بسیار مشکل می‌نماید و منجر به ایجاد خطاهای بسیاری می‌گردد. نویسندگان اعلام داشتند که سختی در پیاده‌سازی ممکن است از ماهیت پیچیده دغدغه‌های سراسری باشد.

Kulesza و همکارانش [۲۸]، با محاسبه مجموعه‌ای از معیارها برای دو شیوه جنبه‌گرا و شی‌گرا به مقایسه این دو روش پرداخته‌اند. نتایج به دست آمده از این مقایسه، تعداد خطوط کمتر، جداسازی بهتر مسائل، وابستگی^{۳۷} ضعیف‌تر و پیچیدگی کمتر برای شیوه جنبه‌گرا می‌باشد. اما در این شیوه، تعداد مؤلفه‌ها و عملیات افزایش می‌یابد.

Gibbs و همکارانش [۲۹]، تکامل نرم‌افزار را بر روی یک سیستم، در دو حالت مختلف بررسی کرده‌اند. حالت اول، تکامل بدون استفاده از روش جنبه‌گرا بوده است و حالت دوم با روش جنبه‌گرا. نتیجه به دست آمده در تایید عملکرد بهتر حالت دوم بوده است.

Hanenbergh و همکارانش [۳۰]، دو روش Aspect-Aware و Aspect-Oriented Refactoring را برای تکامل نرم‌افزارهای جنبه‌گرا معرفی نموده‌اند. در

فهرستی از ابزارهای معروف کاوش جنبه که در صنعت و یا محیط‌های آکادمیک مورد استفاده قرار گرفته‌اند را به همراه برخی از ویژگی‌های آن‌ها ارائه می‌دهد.

Marin و همکارانش [۳۲]، با استفاده از دسته‌بندی^{۳۹}، شیوه‌ای را برای پیاده‌سازی دغدغه‌های سراسری ارائه می‌نمایند.

۷-۱-۱- روش‌های ایستا

تحلیل تعداد فراخوانی‌های ورودی^{۴۸}: ایده اصلی این روش بر تعداد دفعاتی می‌باشد که یک متد فراخوانی می‌شود. به این معنا که اگر تعداد فراخوانی‌های یک متد زیاد باشد، آن متد می‌تواند به عنوان یک کاندید برای تبدیل شدن به جنبه در نظر گرفته شود [۳۹، ۳۸]. مسائل ثبت وقایع، ردیابی، بررسی پیش‌شرایط و پس‌شرایط، و مدیریت خطاها از جمله دغدغه‌های سراسری هستند که با استفاده از این روش، در کد برنامه‌ها شناسایی شده‌اند [۷]. این روش شامل سه مرحله اصلی زیر می‌باشد [۳۹]:

- برای کلیه متدهای برنامه، به صورت خودکار، تعداد دفعاتی که هر کدام از متدها در آن برنامه فراخوانی شده‌اند، محاسبه می‌گردد.
- متدهایی که تعداد فراخوانی‌های آن‌ها از یک حد آستانه^{۴۹} کمتر است و همچنین، متدهای setter، getter، و نیز متدهای کمکی مانند toString() حذف می‌شوند.
- متدها در قالب فراخوانی کننده، فراخوانی شده، قوانین نامگذاری^{۵۰} و نیز پیاده‌سازی آن متدها، به منظور شناسایی جنبه‌ها به صورت دستی، تحلیل می‌گردند.

استفاده از کلونی‌های برنامه^{۵۱}: کلونی‌های یک برنامه، قطعه‌کدهایی از برنامه می‌باشند که بر اساس یک معیار شباهت (مثلاً، شباهت متنی، شباهت معنایی، و یا شباهت در دستورات برنامه‌نویسی)، با یکدیگر مشابه می‌باشند [۶]. بنابراین، کلونی‌های یک برنامه می‌توانند کاندیدای مناسبی برای دغدغه‌های سراسری یا جنبه‌ها باشند [۴۰]. در این روش، آن دسته از کلونی‌های برنامه که تنیدگی بیشتری در کد برنامه دارند، دارای امتیاز بیشتری برای کاندیدا شدن هستند. برای یافتن کلونی‌های برنامه می‌توان از ابزارهایی مانند CCFinder استفاده نمود [۷، ۱۲].

تحلیل شناسه^{۵۲}: در برنامه‌هایی که از قوانین نامگذاری مشخص برای نوشتن کد استفاده می‌نمایند، کاندیدای جنبه می‌تواند با شناسایی گروه‌هایی با نام‌های مشابه انتخاب گردند [۴۱]. به عنوان مثال، شناسه createUndoActivity به سه بخش undo، create و activity تقسیم می‌شود. سپس با استفاده از الگوریتمی، لغاتی که ریشه مشابه دارند در یک گروه قرار می‌گیرند (مانند undoable و undo). در نهایت، گروه‌هایی را که دارای متدها و کلاس‌های بیشتری هستند به عنوان کاندیدای جنبه انتخاب می‌شوند [۷، ۱۲، ۳۸، ۴۱].

پردازش زبان طبیعی^{۵۳} (NLP): این روش، همانند روش تحلیل شناسه، از قوانین نامگذاری برای یافتن دغدغه‌های سراسری استفاده می‌نماید. یکی از روش‌های مورد استفاده در NLP، روش زنجیره لغات است که کلمات مشابه از دیدگاه معنایی را شناسایی می‌کند. این روش مشابهت، با معیار اندازه‌گیری فاصله تعیین می‌شود. الگوریتم زنجیره لغات بر روی توضیحات^{۵۴}، نام متدها، فیلدها و نیز کلاس‌ها اعمال می‌گردد. خروجی این الگوریتم، لیستی از زنجیره لغات است که از دید معنایی بسیار به هم شباهت دارند. استفاده کننده این روش می‌بایست به صورت دستی از میان زنجیره لغات، کاندیداهای جنبه را انتخاب نماید [۷، ۱۲].

شناسایی متدهای یکتا^{۵۵}: تعریف متد یکتا عبارت است از "متدی که مقدار بازگشتی^{۵۶} نداشته، تنها یک پیام را پیاده‌سازی می‌کند، و توسط هیچ متد دیگری پیاده‌سازی نمی‌شود". در این روش، پس از شناسایی متدهای یکتا، آن‌ها بر اساس تعداد فراخوانی مرتب می‌شوند. در این مرتب‌سازی، متدهای نامربوط، مانند متدهای دسترسی، حذف می‌گردند. کسی که از این روش استفاده می‌کند، بایستی

۶-۳- استخراج دغدغه‌های سراسری و اطمینان از صحت کد استخراج‌شده برای جنبه

Monteiro و Fernandes [۳۳] و همچنین Hanenberg [۳۰] و همکارانش، در ارتباط با انتقال کد Advice، مقالاتی را ارائه داده‌اند. اما هیچ یک از این دو روش، به صورت خودکار، قابل اجرا نمی‌باشند. در روش Hanenberg و همکارانش، قبل از انتقال کد^{۴۰}، متغیرهای محلی، برای مشخص شدن ارجاعات دیگر بخش‌های کد به آن‌ها، مورد بررسی قرار می‌گیرند و در صورت عدم وجود ارجاعات، انتقال کد انجام می‌شود. اما در روش Monteiro و Fernandes با اعمال ساده‌سازی^{۴۱} و روش متد جایگزین، سعی بر حل مشکلات متغیرهای محلی دارد. تعیین Join point، Advice و Pointcut مناسب نیز، با استفاده از همین شیوه می‌تواند انجام گیرد. ساده‌سازی ارائه شده می‌تواند بر تکامل نرم‌افزارهای جنبه‌گرا نیز موثر باشد. Binkley و همکارانش [۳۴]، روشی خودکار را برای استخراج کد و تعیین Join point و Advice مناسب، ارائه کرده‌اند (مانند آنچه در روش Monteiro و Fernandes وجود داشت). اما این روش هنوز به صورت عملی مورد استفاده قرار نگرفته است. در روش ارائه شده، هفت نوع استخراج کد به همراه تولید Pointcut مناسب آن‌ها، توصیف گردیده است.

Ettinger و Verbaere [۳۵]، روشی خودکار را بر مبنای روش قطعه‌بندی^{۴۲}، ارائه کرده‌اند. بر اساس این ایده، یک قطعه، شامل تمامی کد دغدغه سراسری می‌باشد. بنابراین می‌توان از روش‌های استخراج قطعه استفاده نمود. Braem و همکارانش [۳۶]، با استفاده از روش‌های یادگیری ماشین، به کشف الگوهایی برای تولید Pointcut از مجموعه‌ای از Join pointها می‌پردازند. Acher و همکارانش [۳۷]، استخراج دغدغه‌های سراسری را در خط تولید نرم‌افزار^{۴۳} و با استفاده از مدل ویژگی‌ها^{۴۴} مورد بررسی قرار می‌دهند. در واقع، این روش، دغدغه‌های سراسری را از مدل ویژگی‌ها به دست می‌آورد.

۷- کاوش جنبه

همانطور که در بخش ۲-۲ اشاره شد، سه فاز اصلی در برنامه‌نویسی جنبه‌گرا وجود دارد که اولین فاز آن کشف جنبه‌ها می‌باشد. فرآیند شناسایی دغدغه‌های سراسری و پیاده‌سازی آن‌ها به صورت کد جنبه، فرآیند کاوش جنبه نامیده می‌شود [۷]. در واقع، هدف از کاوش جنبه را می‌توان بهبود قابلیت درک سیستم و همچنین بهبود توانایی سیستم در مهاجرت به سمت برنامه‌نویسی جنبه‌گرا دانست [۳۸].

۷-۱- روش‌های کاوش جنبه

روش‌های کاوش جنبه را می‌توان در سه دسته ایستا^{۴۵}، پویا^{۴۶} و ترکیبی^{۴۷} تقسیم‌بندی نمود. در روش‌های ایستا، فرآیند کاوش جنبه مستقیماً بر روی کد برنامه انجام می‌گیرد. این در حالیست که روش‌های پویا، از طریق اجرای برنامه‌ها و با استفاده از اطلاعات به دست آمده از زمان اجرا، به شناسایی جنبه‌ها می‌پردازند. دسته سوم، روش‌های ترکیبی هستند. همانطور که از نام این روش‌ها مشخص می‌باشد، راهکار ارائه شده در این روش‌ها، ترکیب روش‌های ایستا و پویا است. در این بخش، به بررسی روش‌های موجود در هر سه دسته می‌پردازیم. در جدول ۴، مقایسه‌ای میان روش‌های کاوش جنبه انجام گرفته است. همچنین، جدول ۵،

برنامه‌ها به صورت ایستا قابل استنتاج و تصمیم‌گیری^{۶۳} نمی‌باشند و فقط در زمان اجرا خود را بروز می‌دهند. از سوی دیگر، روش‌های پویا، بسیار وابسته به ورودی‌ها و موارد کاربردی می‌باشند که برای اجرای آن برنامه مورد استفاده قرار گرفته‌اند. بنابراین، لزوماً تمام ویژگی‌ها و وظیفه‌مندی‌های سیستم اجرا نگردیده‌اند. هدف از ترکیب روش‌های ایستا و پویا و ارائه روش‌های ترکیبی این است که معایب هر دو روش را پوشش دهد. اولین روش ارائه شده در تکنیک پویا، می‌تواند یک روش ترکیبی به شمار آید.

روش DynAMiT که پیش از این معرفی نمودیم، می‌تواند به عنوان یک مثال از روش‌های ترکیبی نیز در نظر گرفته شود. زیرا تحلیل ردیابی‌های به دست آمده با استفاده از اطلاعات ایستا در کد مانند فراخوانی متدها، انجام می‌شود.

در مقاله [۲۸]، به مقایسه سه تکنیک تحلیل تعداد فراخوانی‌های ورودی، تحلیل شناسه و FCA با اعمال هر سه روش بر روی یک برنامه رسم اشکال به نام JHotDraw می‌پردازد. در نتایج به دست آمده، دو روش تحلیل تعداد فراخوانی‌های ورودی و FCA، به گونه‌ای مکمل یکدیگر می‌باشند. اما روش تحلیل شناسه، تا حدی از این دو روش متمایز است. در این مقاله، از نتایج به دست آمده برای ترکیب دو روش FCA و تحلیل تعداد فراخوانی‌های ورودی استفاده کرده است. در این روش ترکیبی، می‌توان متدها با تعداد فراخوانی‌های ورودی زیاد (روش تحلیل تعداد فراخوانی‌های ورودی) که عملکردهای مشابه دارند (روش FCA) را، به عنوان کاندیدای جنبه انتخاب نمود.

در مقاله [۴۵]، یک روش برای ارزیابی روش‌های کاوش جنبه، معرفی کرده است. سپس، با استفاده از روش ارزیابی ارائه شده، به بررسی کیفیت خروجی این روش‌ها و همچنین به بررسی صحت جداسازی جنبه‌ها در روش‌های کاوش جنبه، پرداخته است. روش ارائه شده، بر اساس مدل‌های رسمی می‌باشد که دارای فرضیاتی برای معیار اندازه‌گیری ارزیابی است. طراحان این روش، آن را بر روی تمامی روش‌های کاوش جنبه معرفی شده در بالا، اعمال کرده‌اند.

۸- نتیجه‌گیری

مسائل مهمی در تصمیمات طراحی وجود دارند که پیاده‌سازی آن‌ها با زبان‌های برنامه‌نویسی موجود، منجر به پراکندگی و در هم تنیده شدن این مسائل در کد برنامه شده و در نتیجه بر کاهش پیمانانه‌ای شدن سیستم اثر خواهند گذاشت. برنامه‌نویسی جنبه‌گرا با هدف برطرف ساختن این مشکل و افزایش پیمانانه‌ای بودن سیستم، در کنار برنامه‌نویسی شی‌گرا قرار گرفته است. علاوه بر مزیت بیان شده که منجر به کاهش پیچیدگی کد سیستم خواهد شد، اعمال آسان‌تر تغییرات به هنگام تکامل نرم‌افزار، از دیگر ویژگی‌های استفاده از این شیوه می‌باشد. اما این شیوه مانند هر روش دیگری در کنار مزایای خود، دارای معایبی نیز می‌باشد. طولانی‌تر شدن زمان کامپایل و ناسازگاری ابزارهای جنبه‌گرا با برخی از زبان‌های برنامه‌نویسی از جمله معایب اصلی این روش به شمار می‌روند. همچنین، برای مهاجرت از شیوه شی‌گرا به جنبه‌گرا لازم است تا به بسیاری از برنامه‌نویسان، آموزش‌هایی در زمینه روش برنامه‌نویسی جنبه‌گرا داده شود.

ابزارها و چارچوب‌های بسیاری برای پشتیبانی از شیوه جنبه‌گرا در مراحل شناسایی نیازمندی‌ها و نیز پیاده‌سازی وجود دارند. در این مقاله تلاش نمودیم تا برخی از مهم‌ترین این ابزارها، دسته‌بندی شده و ویژگی‌های آن‌ها نیز بیان گردد. موضوع مهمی که در برنامه‌نویسی جنبه‌گرا وجود دارد، تعیین جنبه‌های سیستم در فازهای مختلف فرآیند ایجاد نرم‌افزار است، برخی بر این باورند که تعیین جنبه‌های سیستم منحصر به زمان پیاده‌سازی آن نیست و در همان فازهای ابتدایی نیز می‌توان جنبه‌های سیستم را مشخص نمود. بر این اساس، رویکردهای متفاوتی را می‌توان داشت که در این مقاله به آن‌ها اشاره شد و در جدولی با یکدیگر مورد

که به صورت دستی از میان متدهای باقیمانده، جنبه‌های کاندید را انتخاب نماید [۷، ۱۲].

خوشه‌بندی^{۵۷} متدهای مرتبط: روش‌های مختلفی مبتنی بر استفاده از الگوریتم‌های خوشه‌بندی، برای کاوش جنبه‌ها وجود دارد [۷، ۱۲، ۴۲]. در روش‌های خوشه‌بندی، متدهای مشابه بر اساس یک معیار شباهت، در خوشه‌های جداگانه تقسیم‌بندی می‌شوند. در روش اول [۴۲]، این خوشه‌ها بر مبنای معیار کمترین فاصله میان خوشه‌ای، با یکدیگر ادغام می‌گردند. در این روش، متدهایی که در هر بار ادغام شرکت داشته‌اند، کاندیدای تبدیل شدن به جنبه هستند. در روش دوم [۱۲]، یک متد، زمانی کاندیدای تبدیل شدن به یک جنبه خواهد بود که در هنگام خوشه‌بندی، تمایل به قرارگیری در چندین خوشه را داشته باشد. یکی دیگر از روش‌های خوشه‌بندی، روش CBFA^{۵۸} می‌باشد که با ترکیب خوشه‌بندی و روش تحلیل تعداد فراخوانی‌های ورودی، به یافتن جنبه‌ها می‌پردازد [۴۳، ۴۴]. به این ترتیب که پس از یافتن تعداد فراخوانی‌های ورودی متدها، آن‌هایی را که دارای اسامی مشابه هستند، در یک خوشه قرار می‌دهد. در نهایت، خوشه‌ای که مجموع تعداد فراخوانی‌های ورودی آن‌ها بالاتر است، به عنوان کاندیدای جنبه برگزیده می‌شود [۴۳، ۴۴].

COMMIT: این روش، مبتنی بر تحلیل تاریخچه کد در سیستم‌های کنترل نسخه^{۵۹} مانند CVS و یا SVN می‌باشد. به این معنا که از کد اصلی برنامه، تمامی تغییراتی را که در یک بازه زمانی خاص رخ داده، استخراج می‌گردد. سپس آن متدها و یا متغیرهایی که در آن بازه زمانی خاص، همزمان با یکدیگر تغییر کرده‌اند را به دست می‌آورد. بر این اساس، گراف وابستگی میان متدها و متغیرها، رسم می‌شود. این روش، با استفاده از یک فرمول آماری، درجه وابستگی را نیز در گراف تعیین می‌کند. آن دسته از متغیرها و متدهایی که نسبت به یکدیگر وابستگی بیشتری دارند، به عنوان کاندیداهای جنبه‌ها شناسایی می‌شوند [۴۴].

۷-۱-۲- روش‌های پویا

DynAMiT: در این روش، ردیابی‌های^{۶۰} برنامه که رفتار زمان اجرای سیستم را نشان می‌دهند، تحلیل می‌شوند. برای تحلیل این ردیابی‌ها، ارتباط میان اجرای فراخوانی متدها، به دست می‌آید. چهار نوع ارتباط در این روش تعریف می‌شود. ارتباط outside-before (متد ۱، قبل از متد ۲، فراخوانی می‌شود)، ارتباط outside-after (متد ۲، بعد از متد ۱، فراخوانی می‌شود)، ارتباط inside-first (متد ۱، اولین متد فراخوانی شده در متد ۲ است)، ارتباط inside-last (متد ۱، آخرین متد فراخوانی شده در متد ۲ است). اگر یکی از این ارتباطات، بیش از یک‌بار وجود داشته باشد (به عنوان مثال، متد ۲ همواره بعد از متد ۱ فراخوانی شود)، آن ارتباط، به عنوان کاندیدای جنبه خواهد بود [۱۲].

تحلیل رسمی مفاهیم^{۶۱} (FCA): در این روش، سیستم بر اساس موارد کاربرد و وظیفه‌مندی‌های^{۶۲} آن اجرا می‌گردد [۱۳]. خروجی این اجرا، مجموعه‌ای از ردیابی‌های اجرایی برنامه است. این ردیابی‌ها توسط الگوریتم FCA تحلیل می‌شوند. در این تحلیل، موارد کاربرد به عنوان اشیای این الگوریتم و متدهای فراخوانی شده در زمان اجرا به عنوان ویژگی‌ها در نظر گرفته می‌شوند. به این ترتیب، تمامی مفاهیمی که شامل ردیابی‌های یک مورد کاربرد مشترک هستند، به عنوان کاندیدای جنبه انتخاب خواهند شد [۷، ۱۲، ۱۳، ۳۸].

۷-۱-۳- روش‌های ترکیبی

مشکل روش‌های ایستا آن است که نتایج خود را مستقیماً از کد برنامه‌ها استخراج می‌کنند، بدون آنکه آن برنامه‌ها را اجرا کنند. ولیکن، یک سری از ویژگی‌های

روش‌های متفاوت آزمون نرم‌افزار قابل بررسی می‌باشد. اما آزمون نرم‌افزارهای جنبه‌گرا به دلیل وجود مفهومی به نام جنبه، در مقایسه با آزمون نرم‌افزارهای غیرجنبه‌گرا دارای چالش‌هایی است. تمرکز آتی ارزیابی جنبه‌گرا، بر روی آزمون نرم‌افزارهای جنبه‌گرا برای اطمینان از صحت عملکرد آن‌ها خواهد بود.

مقایسه قرار گرفتند. اما مرحله مهمی که در شیوه جنبه‌گرا وجود دارد، کاوش جنبه است، که در این مقاله روش‌های مطرح شده در این زمینه و همچنین ابزارهای موجود، به صورتی جداگانه مورد بررسی قرار گرفتند. همانطور که در بخش ۶ اشاره شد، هنوز مسائل مبهم بسیاری در زمینه برنامه‌نویسی جنبه‌گرا وجود دارند که نیازمند تحقیقات بیشتر می‌باشند. حفظ رفتار سیستمی که به شیوه جنبه‌گرا تولید می‌گردد، از جمله مسائل مهمی است که با

جدول ۴- مقایسه روش‌های کاوش جنبه

روش‌های کاوش جنبه	نوع تکنیک	پیش شرایط	برنامه استفاده شده در مطالعه موردی	معیاب
تحلیل گنجایش ورودی	ایستا	حداقل یکی از متدها باید دارای تعداد فراخوانی‌های ورودی بالا باشد.	TomCat 5.5 API & JHotDraw	۱- وابسته به تعیین حد آستانه برای تعداد فراخوانی‌های ورودی است. ۲- متدهای کوچک و یا به عبارتی متدهای دارای تعداد فراخوانی‌های ورودی پایین حذف می‌شوند.
کلونی‌های برنامه	ایستا	برنامه باید دارای کلونی باشد.	TomCat & ASML C-Code	۱- اگر ابزاری که برای یافتن کلونی‌های برنامه استفاده می‌شود، نتواند به خوبی عمل نماید، در شناسایی نادرست دغدغه‌های سراسری بی‌تأثیر نخواهد بود.
تحلیل شناسه	ایستا	قوانین نامگذاری در برنامه‌نویسی باید رعایت شده باشد.	JHotDraw	۱- اطلاعات جمع‌آوری شده از کدهای برنامه در برخی مواقع شامل نویز است. ۲- در برخی مواقع، مفاهیم به دست آمده تمامی جنبه‌ها را پوشش نمی‌دهند.
پردازش زبان طبیعی	ایستا	برنامه باید دارای اسامی مترادف باشد.	PetStore	۱- اگر برنامه دارای اسامی مترادف نباشد، تجزیه برنامه با استفاده از این تکنیک، با شکست مواجه خواهد شد.
شناسایی متدهای یکتا	ایستا	وجود حداقل دو متد به گونه‌ای که یکی، مسئله Cross Cut باشد و دیگری نباشد.	Smalltalk image	۱- این تکنیک وابستگی زیادی به نحوه برنامه‌نویسی دارد. به گونه‌ای که در برخی مواقع ممکن است، برنامه‌نویس جنبه‌ها را به صورت متدهای یکتا ننوشته باشد.
خوشه‌بندی متدهای مربوطه	ایستا	برنامه باید دارای متدهایی با اسامی مترادف باشد.	JHotDraw & Banking example	۱- محاسبات طولانی و نیاز به داشتن دانش در زمینه روش‌های مختلف داده‌کاو، علاوه بر نیاز به دانش برنامه‌نویسی (که در تمام روش‌ها نیاز است)، از مشکلات این روش می‌باشد.
COMMIT	ایستا	کد منبع برنامه در دسترس باشد.	PostgreSQL & NetBSD	۱- انتخاب نامناسب بازه زمانی برای استخراج تغییرات، می‌تواند در یافتن جنبه‌ها تأثیرگذار باشد.
تحلیل رسمی مفاهیم	پویا	وجود حداقل دو مورد کاربرد به گونه‌ای که یکی، مسئله Cross Cut باشد و دیگری نباشد.	JHotDraw	۱- جنبه‌هایی که در تمامی اجزای برنامه شرکت داشته‌اند، شناسایی نمی‌شوند. ۲- کدهایی که قابل اجرا نیستند (در چارچوب برنامه)، در نظر گرفته نمی‌شوند.
DynAMiT	پویا/ ترکیبی	ترتیب فراخوانی‌ها باید مشخص باشد.	Graffiti	۱- ترتیب فراخوانی متدهای برنامه باید مشخص باشد.
تحلیل گنجایش ورودی - FCA	ترکیبی	وجود حداقل دو مورد کاربرد به گونه‌ای که یکی، مسئله Cross Cut باشد و دیگری نباشد و حداقل یکی از متدها باید دارای تعداد فراخوانی‌های ورودی بالا باشد.	JHotDraw	۱- این روش بیشتر در حد تئوری است و هنوز در عمل، خیلی مورد استفاده قرار نگرفته است.

جدول ۵- ابزارهای کاوش جنبه

ابزارهای کاوش جنبه	ویژگی‌ها
Aspect Mining Tool (AMT)	تحلیل‌های انجام شده برای یافتن جنبه بر اساس دو نوع متن و نوع و بر روی کد برنامه انجام می‌شوند. این ابزار می‌تواند با تحلیل‌های دیگری نیز مانند مبتنی بر امضا ^{۶۴} کار کند. هر یک از این تحلیل‌ها به عنوان یک پرس‌وجو عمل کرده و خروجی به صورت گرافیکی نمایش داده می‌شود [۷، ۵۰، ۵۱].
Aspect Browser	یک Eclipse Plugin می‌باشد. ورودی این ابزار می‌تواند به صورت عبارات منظم باشد، که با تحلیل لغوی این ورودی، خروجی به صورت گرافیکی نمایش داده می‌شود [۷، ۵۲].
Prism	یک Eclipse Plugin می‌باشد. ورودی این ابزار می‌تواند مجموعه‌ای از الگوهای جستجو باشد. ابزار با استفاده از این الگوها به یافتن جنبه در کد برنامه می‌پردازد و خروجی را نمایش می‌دهد. در حال حاضر این ابزار جای خود را به (PQL) Prism Query Language که شباهتی با زبان SQL دارد، داده است [۵۳].
JQuery	یک Eclipse Plugin می‌باشد. ورودی این ابزار می‌تواند به صورت عبارات منظم باشد [۵۱]. این ابزار همچنین می‌تواند به صورت سلسله مراتبی، کد برنامه را پیمایش نماید [۷].
DynAMiT	ابزاری است که در روش کاوش جنبه DynAMiT کاربرد دارد [۱۲].
Dynamo & ToscanaJ	ابزارهایی هستند که در روش کاوش جنبه FCA کاربرد دارند [۱۳].

مراجع

[8] N. Bhatnagar, "A Survey of Aspect-oriented Programming Languages," <http://classes.soe.ucsc.edu/cms203/Fall04/finalreports/NeerjapaperAOP.pdf>, June 2004.

[9] K. Govindraj, S. Narayanan, B. Thomas, P. Nair, and S. Peeru, "On Using AOP for Application Performance Management," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 206-212, 2006.

[10] U. Hohenstein, "Improving the Performance of Database Applications with Aspect-oriented Programming," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 12-18, 2006.

[11] A. Duck, "Implementation of AOP in Non-academic Projects," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 68-78, 2006.

[12] A. Kellens, K. Mens, and P. Tonella, "A Survey of Automated Code-level Aspect Mining Techniques," *IEEE Trans. Aspect-oriented Software Development*, vol. 4, no. 6, pp. 143-162, 2007.

[13] P. Tonella, and M. Ceccato, "Aspect Mining through the Formal Concept Analysis of Execution Traces," *Proc, IEEE Int'l Conf. Reverse Engineering*, pp. 112-121, 2004.

[14] O. Spinczyk, A. Gal, and W. Schröder-Preikschat, "Aspectc++: An Aspect-oriented Extension to the C++ Programming Language," *Proc, IEEE Int'l Conf. Tools Pacific: Objects for Internet, Mobile and Embedded Applications*, pp. 53-60, 2002.

[1] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. Loingtier, and J. Irwin, "Aspect-oriented Programming," *Proc, IEEE Int'l Conf. Object-oriented Programming*, pp. 220-242, 1997.

[2] R. J. Walker, E. Baniassad, and G. Murphy, "An Initial Assessment of Aspect-oriented Programming," *Proc, IEEE Int'l Conf. Software Engineering*, pp. 16-22, 1999.

[3] A. Schauerhuber, W. Schwinger, E. Kapsammer, W. Retschitzegger, M. Wimmer, and G. Kappel, *A Survey on Aspect-oriented Modeling Approaches*, Technical Report, Vienna University of Technology, Wien, Austria, 2007.

[4] S. Vijay, and A. Shetty, "A Study on Different Approaches Towards Aspect-oriented Requirements Engineering," *Indian Journal of Computer Science and Engineering*, vol. 2, no. 4, pp. 407-419, 2011.

[5] A. Sampaio, N. Loughran, A. Rashid, and P. Rayson, "Mining Aspects in Requirements," *Proc, IEEE Int'l Workshop on Aspect-oriented Requirements Engineering and Architecture Design*, pp. 62-66, 2005.

[6] T. Mens, and S. Demeyer, "Software Evolution," *Proc, IEEE Int'l Conf. Software Engineering*, pp. 116-122, 2008.

[7] M. Ceccato, "Migrating Object-oriented Code to Aspect-oriented Programming," *Proc, IEEE Int'l Conf. Software Maintenance*, pp. 497-498, 2007.

- Pace," *Proc, IEEE Int'l Conf. Object-oriented Programming*, pp. 241-261, 2005.
- [30] S. Hanenberg, C. Oberschulte, and R. Unland, "Refactoring of Aspect-oriented Software," *Proc, IEEE Int'l Conf. Object-oriented and Internet-based Technologies, Concepts, and Applications for a Networked World*, pp.19-35, 2003.
- [31] M. Bruntink, A. Van Deursen, and T. Tourwé, "Discovering Faults in Idiom-based Exception Handling," *Proc, IEEE Int'l Conf. Software Engineering*, pp. 242-251, 2006.
- [32] M. Marin, L. Moonen, and A. Van Deursen, "A Common Framework for Aspect Mining based on Cross Cutting Concern Sorts," *Proc, IEEE Int'l Conf. Reverse Engineering*, pp. 29-38, 2006.
- [33] M. P. Monteiro, and J. M. Fernandes, "Object-to-Aspect Refactorings for Feature Extraction," *Proc, IEEE Int'l Conf. Software Engineering*, pp. 31-37, 2004.
- [34] D. Binkley, M. Ceccato, M. Harman, F. Ricca, and P. Tonella, "Automated Refactoring of Object-oriented Code into Aspects," *Proc, IEEE Int'l Conf. Software Maintenance*, pp. 27-36, 2005.
- [35] R. Ettinger, and M. Verbaere, "Untangling: A Slice Extraction Refactoring," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 93-101, 2004.
- [36] M. Braem, K. Gybels, A. Kellens, and W. Vanderperren, "Automated Pattern-based Pointcut Generation," *Proc, IEEE Int'l Symp. Software Composition*, pp. 42-46, 2006.
- [37] M. Acher, P. Collet, P. Lahire, and R. France, "Separation of Concerns in Feature Modeling: Support and Applications," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 1-12, 2012.
- [38] M. Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, and T. Tourwé, "A Qualitative Comparison of Three Aspect Mining Techniques," *Proc, IEEE Int'l Workshop on Program Comprehension*, pp. 13-22, 2005.
- [39] M. Marin, A. Van Deursen, and L. Moonen, "Identifying Aspects Using Fan-in Analysis," *Proc, IEEE Int'l Conf. Reverse Engineering*, pp. 8-12, 2004.
- [40] M. Bruntink, A. Van Deursen, R. Van Engelen, and T. Tourwé, "An Evaluation of Clone Detection Techniques for Identifying Cross-cutting Concerns," *Proc, IEEE Int'l Conf. Software Maintenance*, pp. 200-209, 2004.
- [41] T. Tourwe, and K. Mens, "Mining Aspectual Views Using Formal Concept Analysis," *Proc, IEEE Int'l Workshop on Source Code Analysis and Manipulation*, pp. 97-106, 2004.
- [42] F. Klafke, and S. Vergilio, "A Clustering based Approach for Aspect Mining and Pointcut Identification," [15] M. Urban, and O. Spinczyk, "Aspect C++ Language Reference," <http://www.Aspectc.Org>, May 2011.
- [16] D. LaErt, and V. Cahill, "Language-independent Aspect-oriented Programming," *ACM Trans. Object-oriented Programing, Systems, Languages and Applications*, vol. 38, no. 11, pp. 1-12, 2003.
- [17] M. Kerste, "Aop Tools Comparison," <http://Www.Ibm.Com/Developerworks/Java/Library/J-Aopwork1/>, September 2005.
- [18] E. Bodden, "Aspect-oriented Approaches Targeting the .Net Framework," <http://Www.Bodden.De/Tools/Aop-Dot-Net>, June 2009.
- [19] D. Stein, S. Hanenberg, and R. Unland. "An Uml-based Aspect-oriented Design Notation," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 106-112, 2002.
- [20] R. Pawlak, L. Seinturier, L. Duchien, L. Martelli, F. Legon, and G. Florin, "Aspect-oriented Software Development with Java Aspect Components," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 343-369, 2005.
- [21] I. Jacobson, and P. Wei, *Aspect-oriented Software Development with Use-cases*, Addison-Wesley, 2005.
- [22] J. Klein, L. H'Elou'Et, and J. J'Éz'Équel, "Semantic-based Weaving of Scenarios," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 27-38, 2006.
- [23] O. Aldawud, T. Elrad, and A. Bader, "UML Profile for Aspect-oriented Software Development," *Proc, IEEE Int'l Workshop on Aspect-oriented Modeling*, pp. 45-49, 2003.
- [24] S. Clarke, and E. Banaissad, *Aspect-oriented Analysis and Design the Theme Approach*, Addison-Wesley, 2005.
- [25] R. France, I. Ray, G. Georg, and S. Ghosh, "Aspect-oriented Approach to Early Design Modeling," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp.173-185, 2004.
- [26] A. Schauerhuber, W. Schwinger, E. Kapsammer, W. Retschitzegger, and M. Wimmer, "Towards A Common Reference Architecture for Aspect-oriented Modeling," *Proc, IEEE Int'l Workshop on Aspect-oriented Programming*, pp. 23-27, 2006.
- [27] M. Bruntink, A. Van Deursen, and T. Tourwé, "Discovering Faults in Idiom-based Exception Handling," *Proc, IEEE Int'l Conf. Software Engineering*, pp. 242-251, 2006.
- [28] U. Kulesza, C. Santanna, A. Garcia, R. Coelho, A. Von Staa, and C. Lucena, "Quantifying the Effects of Aspect-oriented Programming: A Maintenance Study," *Proc, IEEE Int'l Conf. Software Maintenance*, pp. 223-233, 2006.
- [29] C. Gibbs, C. R. Liu, and Y. Coady, "Sustainable System Infrastructure and Big Bang Evolution: Can Aspects Keep

پژوهشی ایشان عبارتند از مهندسی نرم‌افزار، آزمون عملکردی و غیرعملکردی نرم‌افزار و پروتکل‌های کنتور هوشمند (DLMS).
آدرس پست‌الکترونیکی ایشان عبارت است از:
e.habibi1367@gmail.com



عباس حیدرنوری مدرک دکترای خود در علوم کامپیوتر را در سال ۱۳۸۸ از دانشگاه واترلو کانادا و مدارک کارشناسی و کارشناسی ارشد در مهندسی نرم‌افزار خود را به ترتیب در سال‌های ۱۳۷۸ و ۱۳۸۰ از دانشگاه صنعتی شریف دریافت نموده است. بعد از اتمام دوره دکترا، ایشان به مدت یک

سال به عنوان پژوهشگر پسادکترا در دانشگاه لوگانو سوییس مشغول به انجام پژوهش بوده است. ایشان سپس به مدت یک سال به عنوان مهندس ارشد نرم‌افزار در زمینه تولید نرم‌افزارهای هوشمند همراه در شرکت در تورنتو کانادا مشغول به کار بوده است. دکتر حیدرنوری از بهار ۱۳۹۲ به عنوان عضو هیأت علمی در دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف مشغول به فعالیت می‌باشد. زمینه‌های پژوهشی ایشان عبارتند از مهندسی نرم‌افزار، مهندسی معکوس و بازمهندسی سیستم‌های نرم‌افزاری، و تولید سیستم‌های نرم‌افزاری به روش مؤلفه‌گرا.

آدرس پست‌الکترونیکی ایشان عبارت است از:

heydarnoori@sharif.edu

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۲/۹/۱۰

تاریخ اصلاح: ۹۲/۱۲/۱۱

تاریخ قبول شدن: ۹۲/۱۲/۱۸

نویسنده مرتبط: الهه حبیبی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران.

<http://Www2.Dc.Ufscar.Br/~Lawasp/2012/Artigos/03.Pdf>, May 2011.

[43] D. Zhang, Y. Guo, and X. Chen, "Automated Aspect Recommendation through Clustering-based Fan-in Analysis," *Proc, IEEE/ACM Int'l Conf. Automated Software Engineering*, pp. 278-287, 2008.

[44] B. Adams, Z. Ming Jiang, and A. E. Hassan, "Identifying Cross-cutting Concerns Using Historical Code Changes," *Proc, IEEE/ACM Int'l Conf. Software Engineering*, pp. 305-314, 2010.

[45] G. Czibula, G. Cojocar, and I. Czibula, "Evaluation Measures for Partitioning-based Aspect Mining Techniques," *Journal of Computers, Communications and Control*, vol. 6, no. 1, pp. 72-80, 2011.

[46] J. Bonér, "Aspectwerkz: Plain Java AOP," *Proc, IEEE Int'l Conf. Aspect-oriented Software Development*, pp. 20-26, 2012.

[47] Multi-Dimensional Separation of Concerns for Java, <http://Www.Research.Ibm.Com/Hyperspace/Hyperj/Hyperj.Htm>, May 2001.

[48] H. Kim, *AspectC#: An Aoad Implementation for C#*, Master Thesis, Trinity College Dublin, Dublin, Ireland, 2002.

[49] Spring Framework, <http://www.Springframework.Org>, July 2012.

[50] J. Hannemann, and G. Kiczales, "Overcoming the Prevalent Decomposition of Legacy Code," *Proc, IEEE Int'l Workshop on Advanced Separation of Concerns*, pp. 167-171, 2001.

[51] A. Deursen, M. Marin, and L. Moonen, "Aspect Mining and Refactoring," *Proc, IEEE Int'l Workshop on Refactoring: Achievements, Challenges, Effects*, pp. 11-21, 2003.

[52] W. G. Griswold, Y. Kato, and J. J. Yuan, *AspectBrowser: Tool Support for Managing Dispersed Aspects*, Technical Report, University of California, San Diego, USA, 1999.

[53] C. Zhang, and H. A. Jacobsen, "Prism is Research in Aspect Mining," *Proc, ACM Int'l Conf. Object-oriented Programming, Systems, Languages and Applications*, pp. 20-21, 2004.



الهه حبیبی مدرک کارشناسی ارشد و کارشناسی خود را در رشته مهندسی نرم‌افزار به ترتیب در دانشگاه صنعتی شریف در سال ۱۳۹۲ و دانشگاه الزهرا (س) در سال ۱۳۸۹ دریافت نموده است. پس از اتمام دوره کارشناسی به مدت ۱ سال در شرکت همکاران سیستم در زمینه آزمون عملکردی نرم‌افزار به فعالیت پرداخته است. مهندس الهه حبیبی در حال حاضر در پژوهشگاه نیرو با عنوان برنامه‌نویس و آزمون‌گر مشغول به کار و پژوهش می‌باشد. زمینه‌های

¹ Aspect-Oriented Software Development

² Functionality

³ Modularity

⁴ Aspect Mining

⁵ Scattered

⁶ Tangled

⁷ CrossCutting Concern

⁸ Separation of Concern

⁹ Frameworks

¹⁰ Evolution

¹¹ Unit

¹² Logging

¹³ InfraRED

¹⁴ Query

¹⁵ Tracing

¹⁶ Extension

¹⁷ Open Source

¹⁸ Constructor

¹⁹ Regular Expression

²⁰ Symmetric

²¹ Asymmetric

²² Class Diagram

²³ Sequence Diagram

²⁴ Open Source Framework

²⁵ Authentication

²⁶ Session

²⁷ Client Server

²⁸ Use Cases

²⁹ Communications Diagram

-
- ³⁰ Message Sequence Chart
 - ³¹ Interaction Diagram
 - ³² Synchronization
 - ³³ Platform
 - ³⁴ Package Diagram
 - ³⁵ Role-Based
 - ³⁶ Natural Language Processing (NLP)
 - ³⁷ Coupling
 - ³⁸ Customize
 - ³⁹ Classification
 - ⁴⁰ Code Transformation
 - ⁴¹ Refactoring
- ^{۴۲} قطعه‌بندی یا Slicing روشی می‌باشد که در آن قطعه‌ای از کد که از متغیرها و دیگر بخش‌های کد تأثیر می‌پذیرد (backward slicing) و یا بر آن‌ها تأثیر می‌گذارد (forward slicing) را شناسایی می‌کنیم.
- ⁴³ Software Product Line (SPL)
 - ⁴⁴ Feature Model
 - ⁴⁵ Static
 - ⁴⁶ Dynamic
 - ⁴⁷ Hybrid
 - ⁴⁸ Fan-in
 - ⁴⁹ Threshold
 - ⁵⁰ Naming Convention
 - ⁵¹ Software Clones
 - ⁵² Identifier Analysis
 - ⁵³ Natural Language Processing (NLP)
 - ⁵⁴ Comments
 - ⁵⁵ Unique Method
 - ⁵⁶ Return Value
 - ⁵⁷ Clustering
 - ⁵⁸ Clustering-Based Fan-in Analysis
 - ⁵⁹ Version Control Systems
 - ⁶⁰ Trace
 - ⁶¹ Formal Concept Analysis (FCA)
 - ⁶² Functionalities
 - ⁶³ Undecidable
 - ⁶⁴ Signature Based

ارائه ابزار شبیه‌ساز ابر داده‌ای بر پایه کلاسیم با قابلیت مدیریت داده و کپی

علی موقر

رضا انتظاری ملکی

سعیده مهری

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

مضاعف‌سازی داده یکی از مهم‌ترین مکانیزم‌های مدیریت داده در سیستم‌های توزیع شده است. از آنجا که ارزیابی کارایی سیاست‌های مربوط به آن و نیز تخصیص منابع در محیط واقعی محاسبات ابر، به علت محدودیت‌های موجود عملاً غیرممکن است، برای این منظور از ابزارهای شبیه‌ساز استفاده می‌گردد. کلاسیم (Cloud Sim) چارچوبی کلی و توسعه‌پذیر برای مدل‌سازی، شبیه‌سازی و تست خدمات کاربردی و زیرساخت محاسبات ابری می‌باشد. اما در کلاسیم برای شبیه‌سازی ابر داده‌ای چندین مشکل اساسی وجود دارد: (۱) نبود قابلیت اجرای درخواست‌های مبتنی بر داده و نحوه تخصیص منبع به آن‌ها (۲) نبود امکان مدیریت داده و کپی (۳) نبود قابلیت پرس‌وجو در مورد تعداد و مکان کپی‌ها، هزینه‌ها، میزان دسترس‌پذیری و موارد مشابه برای ارائه الگوریتم‌های نوین مضاعف‌سازی داده. از طرف دیگر، ابزار شبیه‌ساز دیگری برای شبیه‌سازی ابر داده‌ای وجود ندارد. با هدف حل سه مشکل مطرح شده، بستر کلاسیم برای شبیه‌سازی ابر داده‌ای بدون تغییر در لایه‌های معماری کلاسیم و صرفاً با توسعه موجودیت‌های موجود در معماری آن، گسترش داده شده و ابزار جدیدی برای شبیه‌سازی ابر داده‌ای تحت عنوان Data Cloud Sim ارائه شده است که امکان توسعه سیاست‌های مدیریت کپی و مضاعف‌سازی داده به صورت آگاهانه‌تر با هدف بهبود کیفیت سرویس و کاهش هزینه‌ها فراهم می‌شود.

کلمات کلیدی: ابر داده‌ای، مرکز داده، مضاعف‌سازی داده، مدیریت داده و کپی، شبیه‌ساز کلاسیم.

۱- مقدمه

است. استفاده از زیرساخت‌های واقعی ابر برای ارزیابی کارایی و تکرار آزمون به علت محدودیت‌های موجود عملاً غیرممکن است. بنابراین از ابزارهای شبیه‌ساز برای این منظور استفاده می‌گردد. ابزارهای شبیه‌ساز متفاوتی در محیط‌های توزیع شده مانند گرید ارائه شده‌اند که از آن جمله می‌توان به گریدسیم [۳]، سیم‌گرید [۴] و گنگسیم [۵] اشاره کرد.

با وجود تشابه‌های زیاد بین دو محیط گرید و ابر، هیچ کدام از این شبیه‌سازها را به صورت مستقیم برای مدل‌سازی و شبیه‌سازی محیط محاسبات ابری نمی‌توان بکار برد. برای شبیه‌سازی محیط ابر، ابزاری به نام کلاسیم (Cloud Sim) [۱] ارائه شده که در واقع چارچوبی کلی و توسعه‌پذیر برای مدل‌سازی، شبیه‌سازی و آزمون خدمات کاربردی و زیرساخت محاسبات ابری است. با استفاده از کلاسیم، محققان و توسعه‌گران می‌توانند کارایی برنامه‌های کاربردی جدید خود را در یک محیط کنترل شده با صرفه‌جویی در زمان، تست و ارزیابی کنند.

با وجود استفاده روزافزون از مراکز داده ابری، چالش‌های زیادی در رابطه با مدیریت داده و کپی برای بهبود کارایی دسترسی مطرح می‌شود. مضاعف‌سازی

محاسبات ابر^۱، سیستم توزیع شده‌ای است که خدمات زیرساخت، بستر و نرم‌افزار را به مشتریان ارائه می‌دهد. نحوه محاسبه هزینه پرداختی مشتریان برای خدمات ارائه شده از مدل پرداخت مبتنی بر استفاده^۲ پیروی می‌کند. خدمات ابر در سه لایه زیرساخت به عنوان سرویس^۳، بستر به عنوان سرویس^۴ و نرم‌افزار به عنوان سرویس^۵ ارائه می‌گردد [۱]. به عبارت دیگر محاسبات ابر، مدلی برای دسترسی راحت و بر حسب تقاضا به منابع محاسباتی از طریق شبکه است. در محاسبات ابر تخصیص یا آزاد کردن منابع با کمترین تلاش مدیر یا فراهم‌کننده سرویس امکان‌پذیر است [۲].

بیشتر خدمات ابری مثل شبکه‌های اجتماعی، میزبانی وب، تحویل محتوا و پردازش داده بلادرنگ، دارای ترکیب، پیکربندی و نیازمندی‌های توسعه متفاوتی هستند. ارزیابی کارایی سیاست‌های زمان‌بندی، تخصیص منابع و مدیریت داده در محیط واقعی محاسبات ابر برای مدل‌های کاربردی مختلف، بسیار چالش برانگیز

این ابزار و نحوه عملکرد آن در این بخش آورده شده است. در بخش ۴، ابزار جدید ارائه شده بر پایه کلاسیسم، در سناریوهای مختلف با استفاده از مجموعه داده مصنوعی و واقعی مورد ارزیابی و تحلیل قرار گرفته و در بخش ۵، نتیجه‌گیری و کارهای آتی برای توسعه این ابزار بیان شده است.

۲- پیش‌زمینه

۲-۱- کارهای مرتبط

سیر تکاملی سیستم‌های محاسباتی از محاسبات خوشه‌ای^{۱۳}، محاسبات گرید^{۱۴} به سمت محاسبات ابر پیش‌رفته است. برای توسعه و تست اجزای جدید و ارزیابی سیاست‌های زمان‌بندی پیشنهادی در گرید، چندین شبیه‌ساز مثل سیم‌گرید [۴]، گریدسیم [۳] و گنگ‌سیم [۵] ارائه شده است. سیم‌گرید ابزاری مبتنی بر زبان C برای شبیه‌سازی کاربردهای توزیع شده در بستر گرید است که امکان مدل‌سازی منابع زمان اشتراکی^{۱۵}، با اعمال بارکاری ثابت یا بارکاری واقعی را فراهم می‌کند.

با استفاده از سیم‌گرید امکان شبیه‌سازی تخصیص منابع به وظایف براساس سیاست زمان‌بندی تعریف‌شده وجود دارد. از طرف دیگر، گرید سیم ابزار شبیه‌ساز مبتنی بر رخداد برای منابع ناهمگن گرید است که با زبان جاوا پیاده‌سازی شده است و قابلیت مدل‌سازی موجودیت‌های گرید، کاربران، کاربردها، واسط‌های منابع و زمان‌بندها را دارد. هم‌چنین امکان مدل‌سازی منابع در هر دو حالت زمان اشتراکی و فضا اشتراکی^{۱۶} را فراهم می‌کند. گنگ‌سیم ابزار شبیه‌ساز دیگری برای مدل‌سازی سازمان‌ها و منابع مجازی مبتنی بر گرید است که تحلیل سیاست‌های مختلف زمان‌بندی را پشتیبانی می‌کند. اما هیچ یک از این شبیه‌سازها قابلیت شبیه‌سازی زیرساخت و نیازمندی‌های سطح کاربردی ابر را ندارند.

کلاسیسم [۱] ابزاری برای مدل‌سازی رفتاری اجزای ابری مثل مراکز داده، ماشین‌های مجازی و سیاست‌های تأمین منابع است که به زبان جاوا پیاده‌سازی شده است. سیاست‌های تأمین منابع در کلاسیسم عمومی، ساده و قابل توسعه است. هم‌چنین امکان توسعه موجودیت‌های کلاسیسم برای افزودن قابلیت‌های بیشتر در شبیه‌سازی محیط‌های ابری توسط محققان و توسعه‌گران وجود دارد. در بخش ۲، کلاسیسم، معماری آن و نحوه شبیه‌سازی در آن به تفصیل آمده است. کلادانالیست (Cloud Analyst) [۹] ابزار شبیه‌سازی و مدل‌سازی دیداری بر پایه کلاسیسم برای کاربردهای ابر در مقیاس بزرگ است. کلادانالیست بر بالای کلاسیسم ساخته شده و یک واسط کاربر گرافیکی به آن اضافه شده است. یکی از اهداف اصلی کلادانالیست، جدا کردن بخش آزمون‌های شبیه‌سازی از بخش برنامه‌نویسی است؛ بنابراین یک مدل‌ساز می‌تواند بدون صرف وقت زیاد برای برنامه‌نویسی، بر روی پیچیدگی‌های شبیه‌سازی تمرکز کند.

از خصوصیات اصلی کلادانالیست داشتن واسط کاربر گرافیکی است که به آسانی و به سرعت می‌توان آزمون‌های شبیه‌سازی را تنظیم نمود و با ذخیره سناریوهای شبیه‌سازی در قالب xml، آزمون‌های شبیه‌سازی را به تعداد زیاد تکرار نمود. خصوصیت دیگر کلادانالیست امکان ایجاد خروجی گرافیکی در قالب نمودار و جدول است که خلاصه‌ای از مقادیر زیاد نتایج آماری در حین شبیه‌سازی را نمایش می‌دهد. خصوصیت اصلی دیگر کلادانالیست، قابلیت تعریف شبیه‌سازی با درجه بالایی از انعطاف‌پذیری و قابل پیکربندی بودن آن است. پیکربندی مراکز داده، ماشین‌های مجازی، سیاست‌های تخصیص منابع (زمان اشتراکی و فضا اشتراکی)، کاربران و توزیع جغرافیایی آن‌ها، پارامترهای متغیر اینترنت شبیه تأخیر شبکه و پهنای باند، زمان شبیه‌سازی و سیاست‌های واسط مراکز داده از جمله تنظیمات موجود در کلادانالیست است.

داده^۶ (یا کپی‌سازی داده) یکی از مهم‌ترین مکانیزم‌های مدیریت داده در سیستم‌های توزیع شده است که با ایجاد کپی‌های متعدد از داده‌ها و توزیع آن‌ها در شبکه، کارایی دسترسی به داده را بهبود می‌بخشد. مکانیزم مضاعف‌سازی داده در سیستم‌های تجاری ابر مثل سیستم ذخیره‌سازی ساده آمازون^۷ [۶]، سیستم فایل گوگل^۸ [۷] و سیستم فایل توزیع شده هادوپ^۹ [۸] استفاده می‌شود. در این سیستم‌های توزیع شده، به صورت پیش‌فرض الگوریتم ایستای سه کپی به ازای هر داده، برای مضاعف‌سازی داده بکار برده می‌شود. ارائه الگوریتم‌های نوین برای مضاعف‌سازی داده در مراکز داده ابری با هدف بهبود کارایی و کاهش هزینه‌های ذخیره‌سازی و ارزیابی کارایی آن‌ها، موضوع چالش برانگیزی است. مسائل مهمی که در مضاعف‌سازی داده مطرح می‌شود عبارتند از زمان ایجاد کپی جدید از هر داده، تعداد کپی، مکان ذخیره کپی، انتخاب داده مناسب برای کپی کردن از آن و جایگزینی کپی به هنگام نبود فضای کافی برای ذخیره‌سازی. اما کلاسیسم امکان شبیه‌سازی ابر داده‌ای^{۱۰} و ارزیابی کارایی الگوریتم‌های مضاعف‌سازی داده را فراهم نمی‌آورد.

با وجود چالش‌های مطرح شده، سه مسئله اساسی برای شبیه‌سازی الگوریتم‌های مضاعف‌سازی داده در کلاسیسم وجود دارد. مسئله اول این است که کلاسیسم قابلیت اجرای درخواست‌های مبتنی بر داده-درخواست‌هایی که برای اجرا نیاز به داده دارند-را ندارد. از طرف دیگر نحوه تخصیص منبع به درخواست‌های مبتنی بر داده موضوعی چالش برانگیز برای محققان است که در کلاسیسم امکان شبیه‌سازی الگوریتم‌های تخصیص منبع برای درخواست‌های مبتنی بر داده فراهم نشده است. مسئله دوم این است که برای مدیریت داده و کپی نیاز به دو موجودیت مدیر کپی^{۱۱} برای انجام عملیات مضاعف‌سازی و کاتالوگ کپی^{۱۲} برای نگهداری اطلاعات کپی‌ها وجود دارد که در کلاسیسم چنین قابلیت‌هایی نیز پیاده‌سازی نشده است. مسئله سوم نیز این می‌باشد که برای پاسخگویی به مسائل مطرح شده در مضاعف‌سازی داده و ارائه الگوریتم‌های نوین مضاعف‌سازی داده نیاز به اطلاعاتی مانند تعداد و مکان کپی‌ها، میزان دسترس‌پذیری هر داده، هزینه ذخیره‌سازی و هزینه مضاعف‌سازی مراکز داده و موارد مشابه است که چنین امکانی در کلاسیسم وجود ندارد.

بنابراین در این مقاله بستر کلاسیسم با هدف حل هر سه مسئله اساسی مطرح شده، گسترش داده شده تا بستر لازم برای مدیریت آگاهانه داده و کپی در مراکز داده ابری فراهم گردد. ابزار ارائه شده قابلیت اجرای درخواست‌های مبتنی بر داده را داشته و هم‌چنین بستری برای پیاده‌سازی سیاست‌های تخصیص منبع برای درخواست‌های مبتنی بر داده را فراهم می‌آورد. از طریق ایجاد دو نقش جدید مدیر کپی و کاتالوگ کپی در موجودیت‌های معماری کلاسیسم و توسعه آن علاوه بر فراهم نمودن امکان مدیریت داده و کپی، اطلاعات مفیدی در مورد تعداد و مکان کپی‌ها، میزان دسترس‌پذیری هر داده، هزینه ذخیره‌سازی و هزینه مضاعف‌سازی داده‌های ذخیره شده در هر مرکز داده ارائه می‌شود. با استفاده از این اطلاعات، امکان ارزیابی برنامه‌های کاربردی ابر که از مدل پرداخت مبتنی بر استفاده پیروی می‌کنند، در ابزار ارائه شده فراهم گردیده است. هم‌چنین ابزار ارائه شده این امکان را فراهم می‌کند تا محققان بتوانند الگوریتم‌های مضاعف‌سازی نوین را پیاده‌سازی، تست و ارزیابی کنند.

ساختار این مقاله به این صورت می‌باشد که در بخش ۲ و در قسمت کارهای مرتبط، ابتدا شبیه‌سازهای ارائه شده برای سیستم توزیع شده گرید معرفی و سپس شبیه‌سازهای ارائه شده در محیط ابر و ابزارهای توسعه‌یافته بر پایه کلاسیسم معرفی و موارد استفاده هر یک بیان شده است. در قسمت دوم این بخش، دانش لازم برای آشنایی با کلاسیسم، معماری و نحوه شبیه‌سازی در آن، بیان شده است. بخش ۳، ابزار جدید ارائه شده بر پایه کلاسیسم برای شبیه‌سازی ابر داده‌ای را معرفی و معماری آن را توضیح داده است. هم‌چنین قابلیت‌های جدید ارائه شده در

در طول اجرا، را فراهم می‌کند.

ریل کلاسیم (Real Cloud Sim) [۱۸]، شبیه‌ساز تخصیص ماشین‌های مجازی بر پایه موتور اصلی در کلاسیم است که از موتور خود نیز برای شبیه‌سازی تخصیص استفاده می‌کند که بر پایه الگوریتم ژنتیک، برنامه‌ریزی مختلط صحیح با نرم‌افزار Lingo و شبیه‌ساز شبکه NS2 است. هم‌چنین واسط گرافیکی آن امکان نمایش همبندی در قالب brute را فراهم می‌کند و یک گزارش کامل به عنوان خروجی تولید می‌کند. کلاداکشن (Cloud Auction) [۱۹] نیز گسترش یافته کلاسیم است که با توسعه کتابخانه‌ای، کلاسیم را برای کار کردن با خدمات مبتنی بر حراج قادر می‌سازد. کلادمیگ اکسپرس (Cloud MIG Xpress) [۲۰] امکان مقایسه و برنامه‌ریزی فازهای مرتبط با مهاجرت سیستم‌های نرم‌افزاری به محیط ابری بستر به عنوان سرویس یا زیرساخت به عنوان سرویس را فراهم می‌کند. این ابزار همچنین امکان تبدیل خودکار مدل سیستم به مدل کلاسیم برای شبیه‌سازی مختلف در مورد توسعه ابر را دارد.

ابزارهای ارائه شده هر یک برای منظور خاصی توسعه داده شده‌اند اما برای ابر ذخیره‌ساز^{۲۳} یا ابر داده‌ای که امکان مدیریت داده و کپی در آن وجود داشته باشد، ابزاری با کد باز ارائه نشده است. تنها برای شبیه‌سازی ابر ذخیره‌ساز، لانگ و همکارش [۲۱] ابزاری ارائه کرده‌اند که با اضافه کردن لایه جدید به معماری کلاسیم، قابلیت‌های تقسیم فایل به بلوک قبل از ذخیره‌سازی، ذخیره چند کپی از داده در مراکز داده متفاوت و امکان افزایش دسترس‌پذیری با مدیریت کپی را ارائه نموده است، اما به نحوه اجرای درخواست‌های مبتنی بر داده و تخصیص منابع به آن، اشاره‌ای نشده است. بر خلاف مقاله مذکور، در این مقاله، بستر کلاسیم برای شبیه‌سازی ابر داده‌ای بدون تغییر در لایه‌های معماری کلاسیم و صرفاً با توسعه موجودیت‌های موجود در معماری کلاسیم، گسترش داده شده و ابزار جدیدی برای شبیه‌سازی ابر داده‌ای ارائه شده است که قابلیت اجرای درخواست‌های مبتنی بر داده را داشته و امکان توسعه و ارزیابی سیاست‌های جدید تخصیص منبع برای درخواست‌های مبتنی بر داده را برای محققان و توسعه‌گران فراهم می‌کند.

ابزار ارائه شده علاوه بر فراهم کردن امکان ذخیره چند کپی از داده در مراکز داده مختلف و امکان پرس‌وجو از مکان کپی، قابلیت پرس‌وجوی تعداد کپی، دسترس‌پذیری و هزینه ذخیره‌سازی هر داده را نیز فراهم کرده است. با استفاده از این اطلاعات، امکان توسعه سیاست‌های مدیریت کپی و مضاعف‌سازی داده به صورت آگاهانه‌تر با هدف بهبود کیفیت سرویس و کاهش هزینه ذخیره‌سازی فراهم شده است. همچنین ابزار ارائه شده دارای قابلیت پرس‌وجو از هزینه ذخیره‌سازی و مضاعف‌سازی هر مرکز داده است که با استفاده از آن، امکان ارزیابی برنامه‌های کاربردی ابر که از مدل پرداخت مبتنی بر استفاده پیروی می‌کنند، فراهم شده است که در بخش ۳ قابلیت‌های جدید ابزار ارائه شده با جزئیات بیشتر توضیح داده شده است.

۲-۲- معرفی کلاسیم

کلاسیم [۱] چارچوبی برای مدل‌سازی و پیاده‌سازی زیرساخت و سرویس‌های ابر است که در آزمایشگاه ابر دانشگاه ملیورن ارائه شده است. کلاسیم مجموعه‌ای از کتابخانه‌ها و کلاس‌های پیاده‌سازی شده به زبان جاوا است که امکان مدل‌سازی رفتاری و سیستمی اجزای ابری مثل مراکز داده، ماشین‌های مجازی و سیاست‌های تأمین منابع را فراهم می‌کند.

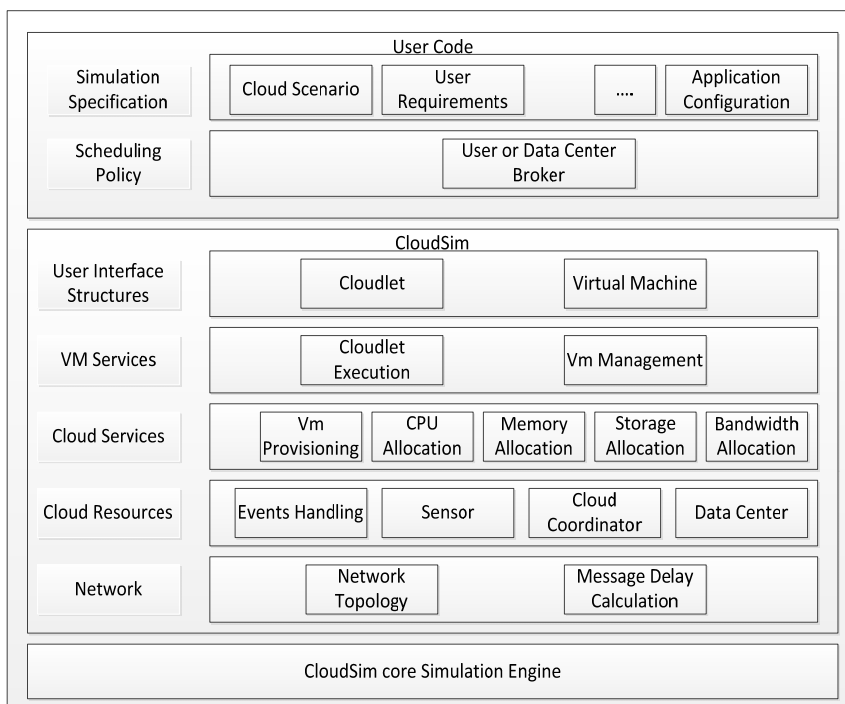
شکل ۱ معماری چند لایه‌ای کلاسیم را نشان می‌دهد. بالاترین لایه پشته‌ی کلاسیم به نام کد کاربر^{۲۴}، در برگیرنده دو بخش اصلی مشخصات شبیه‌سازی و سیاست‌های زمان‌بندی است.

گرین کلاود (Green Cloud) [۱۰] یک شبیه‌ساز شبکه در سطح بسته^{۱۷} برای مراکز داده آگاه از انرژی^{۱۸}، با تمرکز بر ارتباطات ابری است. گرین کلاود برای گرفتن اطلاعات جزئی در مورد انرژی مصرفی اجزای مراکز داده (سرورها، سوئیچ‌ها و پیوند^{۱۹}ها) طراحی گردیده است. گرین کلاود بر پایه شبیه‌ساز شبکه NS2، پروتکل TCP/IP را به صورت کامل پیاده‌سازی کرده است و امکان شبیه‌سازی پروتکل‌های دیگر را نیز فراهم می‌آورد. تنها نقطه‌ضعف گرین کلاود این است که مناسب مقیاس کوچکی از مراکز داده است زیرا که زمان و حافظه‌ی زیادی برای شبیه‌سازی لازم دارد. نتورک کلاسیم (Network Cloud Sim) [۱۱] گسترشی از کلاسیم، برای مدل‌سازی مراکز داده ابر واقعی و کاربردهای تعمیم‌یافته مثل محاسبات با کارایی بالا^{۲۰}، تجارت الکترونیکی و گردش کاری است.

شبیه‌سازی کاربردهای موزی و توزیع شده در شبیه‌سازهای دیگر به صورت کامل در نظر گرفته نشده است. مدل‌های کاربرد در محاسبات ابری از کاربردهای وب چند لایه‌ای مثل تجارت الکترونیکی تا کاربردهای علمی می‌تواند متغیر باشد. برخی از این کاربردها از چندین وظیفه تشکیل یافته است که نیاز به ارتباط با یکدیگر دارند. برای شبیه‌سازی رفتار چنین کاربردهای پیچیده موزی و توزیع شده در نتورک کلاسیم، ساختار و توابع جدیدی به کلاسیم اضافه شده است. خصوصیت مهم دیگر نتورک کلاسیم، امکان مدل‌سازی کامل همبندی^{۲۱} واقعی در داخل مراکز داده بین ماشین مجازی، ماشین میزبان و سوئیچ است. در کلاسیم، صرفاً همبندی بین مراکز داده در نظر گرفته شده است. برای ارزیابی دقیق کاربردهایی با ارتباطات بین وظایف و مهاجرت ماشین‌های مجازی، در نظر گرفتن نحوه همبندی شبکه، تأخیر و پهنای باند بسیار مهم و تأثیرگذار است.

اموسیم (EMUSIM) [۱۲] محیط شبیه‌سازی برای مدل‌سازی، ارزیابی و اعتبارسنجی کارایی کاربردهای محاسبات ابری است که ترکیبی از نمونه‌سازی و شبیه‌سازی است. اموسیم بر روی دو بستر ساخته شده است: چارچوب نمونه‌سازی خودکار^{۲۲} برای نمونه‌سازی و کلاسیم برای شبیه‌سازی. اموسیم، از طریق نمونه‌سازی، اطلاعاتی خودکار، از رفتار کاربردها به دست می‌آورد و از این اطلاعات برای تولید مدل شبیه‌سازی استفاده می‌نماید تا تأثیر تعداد منابع مختلف و الگوی درخواستی کاربردهای ابری را ارزیابی کند. ام‌دی‌سی‌سیم (MDC Sim) [۱۳] شبیه‌ساز رخداد گسسته تجاری است که در دانشگاه پنسیلوانیا توسعه داده شده است. ام‌دی‌سی‌سیم، امکان مدل‌سازی خصوصیات سخت‌افزاری اجزای مختلف مرکز داده مانند سرورها، پیوندهای ارتباطی و سوئیچ‌ها از تأمین‌کنندگان مختلف و هم‌چنین تخمین انرژی مصرفی آن‌ها را فراهم می‌کند. این ابزار دارای سربار شبیه‌سازی کمی بوده و علاوه بر این، همبندی مراکز داده را به شکل گراف جهت‌دار نگهداری می‌کند [۱۴].

کلادریپورت (Cloud Reports) [۱۵] ابزار گرافیکی برای شبیه‌سازی محیط‌های توزیع شده محاسبات ابری است که از کلاسیم برای شبیه‌سازی استفاده می‌کند. کلادریپورت دارای واسط کاربر گرافیکی است که استفاده از آن را آسان می‌سازد. همچنین این ابزار امکان ایجاد گزارش به قالب HTML و فایل‌های داده که با استفاده از نرم‌افزارهای دیگر مثل متلب قابل استفاده است را فراهم می‌کند. کلادریپورت فراهم‌کننده‌ی لایه زیرساخت به عنوان سرویس را با تعداد دلخواه مراکز داده و پیکربندی سفارشی، شبیه‌سازی می‌کند. علاوه بر این، مشتریان لایه زیرساخت به عنوان سرویس نیز به صورت سفارشی در این ابزار قابل شبیه‌سازی هستند. ورک‌فلوسیم (Work Flow Sims) گسترش‌یافته کلاسیم است [۱۶] که آماده‌سازی و اجرای گردش کار را از طریق پیاده‌سازی پشته‌ای تجزیه‌کننده گردش کار، موتور گردش کار و زمان‌بند وظیفه، فراهم می‌کند. داینامیک کلاسیم (Dynamic Cloud Sim) [۱۷] نیز گسترش یافته کلاسیم است که امکان مدل (۱) ناهمگنی در کارایی منابع محاسباتی، (۲) عدم قطعیت در تغییرات متغیر کارایی ماشین‌های مجازی و (۳) ماشین‌های سرگردان و شکست‌ها



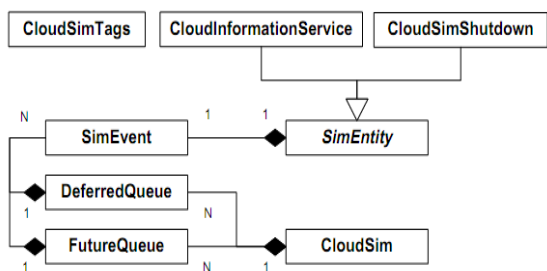
شکل ۱- معماری چند لایه‌ای کلاسیسم [۱]

زیادی برچسب رخداد یا دستور است که نوع اقدام لازم برای انجام، موقع ارسال یا دریافت رخداد را تعیین می‌کند [۲۲]. مدل‌سازی در ابر با استفاده از کلاسیسم بسیار گسترده و دارای جنبه‌های مختلفی است. در ادامه موجودیت‌های اصلی ابر در کلاسیسم توضیح داده می‌شود:

• مرکز داده:

یک مرکز داده شامل تعدادی موجودیت میزبان است که هر میزبان نشان‌دهنده‌ی یک سرور فیزیکی محاسباتی در ابر است. برای هر میزبان می‌توان تنظیمات سخت‌افزاری همگن یا ناهمگن در مورد میزان حافظه، فضای ذخیره‌سازی، قدرت محاسباتی و موارد دیگر را تعیین کرد. هر میزبان می‌تواند دارای یک یا تعداد بیشتری ماشین مجازی باشد که توسط سیاست‌های تخصیص ماشین مجازی^{۲۹} تعریف شده در فراهم‌کننده سرویس ابر، ماشین‌های مجازی به میزبان‌ها تخصیص داده می‌شود.

هر مرکز داده می‌تواند شامل شبکه‌های ذخیره‌سازی به نام SAN^{۳۰} باشد که امکان شبیه‌سازی بازبازی داده را فراهم می‌کند. دسترسی به فایل‌ها در SAN دارای تأخیری اضافه بر زمان اجرای کار است که به علت انتقال داده در شبکه داخلی مرکز داده رخ می‌دهد.



شکل ۲- نمودار کلاسی از کلاس‌های اصلی در هسته کلاسیسم [۱]

که موجودیت‌های اصلی برای میزبان‌ها (مانند تعداد ماشین‌ها و خصوصیات آن‌ها)، برنامه‌های کاربردی (مانند تعداد وظایف و نیازمندی آن‌ها)، ماشین‌های مجازی^{۲۵}، تعداد کاربران، نوع برنامه کاربردی و سیاست‌های زمان‌بندی واسط را تعیین می‌کند. امکان توسعه این موجودیت‌ها برای افزودن قابلیت‌های جدید وجود دارد. لایه میانی با نام شبیه‌ساز کلاسیسم قابلیت مدل‌سازی و شبیه‌سازی محیط‌های مبتنی بر مراکز داده مجازی ابر را فراهم می‌کند که شامل واسط‌های مدیریتی مشخص برای مدیریت ماشین‌های مجازی، حافظه، فضای ذخیره‌سازی و پهنای باند است.

مسائل اساسی مثل تخصیص میزبان به ماشین‌های مجازی، مدیریت اجرای برنامه کاربردی و نظارت سیستم‌های پویا توسط این لایه صورت می‌گیرد. در پایین‌ترین لایه، کلاسی به نام کلاسیسم است که مسئول مدیریت صف رخدادها و کنترل گام به گام اجرای رخدادهای شبیه‌سازی است.

شکل ۲ نمودار کلاسی عنصرهای اصلی و هسته در کلاسیسم را نشان می‌دهد. هر عنصری در کلاسیسم به نام موجودیت شناخته می‌شود که از کلاس Sim Entity ارث می‌برد و دارای سه تابع Start Entity، Process Entity و Shutdown Entity است که عملیات مقداردهی، پردازش رخدادها و از بین رفتن موجودیت را انجام می‌دهند. هر موجودیتی که ایجاد می‌شود به صورت خودکار در سرویس اطلاعاتی ابر^{۲۶} ثبت می‌شود. Cloud Sim Shutdown موجودیتی است که منتظر اتمام همه موجودیت‌ها است تا در پایان شبیه‌سازی به سرویس اطلاعاتی ابر اطلاع بدهد.

هر موجودیت قابلیت ارسال پیغام به موجودیت‌های دیگر و پردازش رخدادها را دارد. هر رخدادی که توسط موجودیتی در کلاسیسم ایجاد می‌شود، در صفی به نام صف آینده^{۲۷} ذخیره می‌گردد. رخدادها با توجه به پارامتر زمانی مرتب و در صف درج می‌شوند. در هر مرحله از شبیه‌سازی، رخدادها زمان‌بندی شده و از صف آینده حذف و به صف معوق^{۲۸} افزوده می‌شوند. مشابه همین عملکرد در هر موجودیت در تابع پردازش رخداد، صورت می‌گیرد؛ رخدادها از صف معوق انتخاب و اقدام متناسب با رخداد انجام می‌شود. اقدام متناسب با رخداد از طریق برچسب موجود در آن رخداد قابل شناسایی است. کلاس Cloud Sim Tags در برگزیده تعداد

• واسط ابر یا واسط مراکز داده:

در ادامه معماری ابزار جدید ارائه شده مورد بررسی قرار گرفته و قابلیت‌های جدید ارائه شده معرفی شده است.

۳-۱- معماری ابزار شبیه‌ساز ابر داده‌ای ارائه شده

همان‌طور که در بخش ۲ اشاره شد، تنها ابزار ارائه شده برای شبیه‌سازی ابر ذخیره‌ساز [۲۱] یک لایه جدید با نام ابر داده‌ای به معماری کلاسیک اضافه کرده است. شکل ۱ معماری ابزار ارائه شده در [۲۱] بر پایه کلاسیک را نشان می‌دهد. این لایه شامل موجودیت‌های جدید مدیر کپی، کاتالوگ کپی، مدیر بلوک، کاتالوگ بلوک و مجموعه داده است. این ابزار قابلیت‌های تقسیم فایل به بلوک قبل از ذخیره‌سازی، ذخیره چند کپی از داده در مراکز داده متفاوت، امکان پرس‌وجو از مکان کپی و افزایش دسترس‌پذیری بر مدیریت کپی را فراهم نموده است. اما در این مقاله بستر کلاسیک برای شبیه‌سازی ابر داده‌ای بدون تغییر در لایه‌های معماری کلاسیک و صرفاً با توسعه موجودیت‌های موجود در معماری پایه کلاسیک، گسترش داده‌شده و ابزار جدیدی برای شبیه‌سازی ابر داده‌ای با قابلیت‌های جدید ارائه شده است. در معماری ابزار ارائه شده، دو موجودیت جدید واسط ابر داده‌ای و مرکز داده ابر داده‌ای که به ترتیب گسترش یافته موجودیت‌های واسط ابر و مرکز داده هستند، اضافه شده است. با ارسال پیغام‌های متعدد بین این دو موجودیت جدید و اضافه نمودن رخداد‌های جدید، قابلیت‌های زیر به کلاسیک اضافه شده است:

- قابلیت اجرای درخواست‌های مبتنی بر داده
 - قابلیت پیاده‌سازی سیاست‌های تخصیص منبع برای درخواست‌های مبتنی بر داده
- قابلیت ذخیره چند کپی از داده در مراکز داده
- امکان پرس‌وجو در مورد تعداد کپی از هر داده و مکان کپی‌ها
- امکان پرس‌وجو در مورد میزان دسترس‌پذیری هر داده در زمان اجرا
- امکان پرس‌وجو در مورد هزینه ذخیره‌سازی هر داده و هزینه مضاعف‌سازی هر مرکز داده در زمان اجرا
- امکان جایگزینی کپی به هنگام نبود فضای کافی برای ذخیره‌سازی

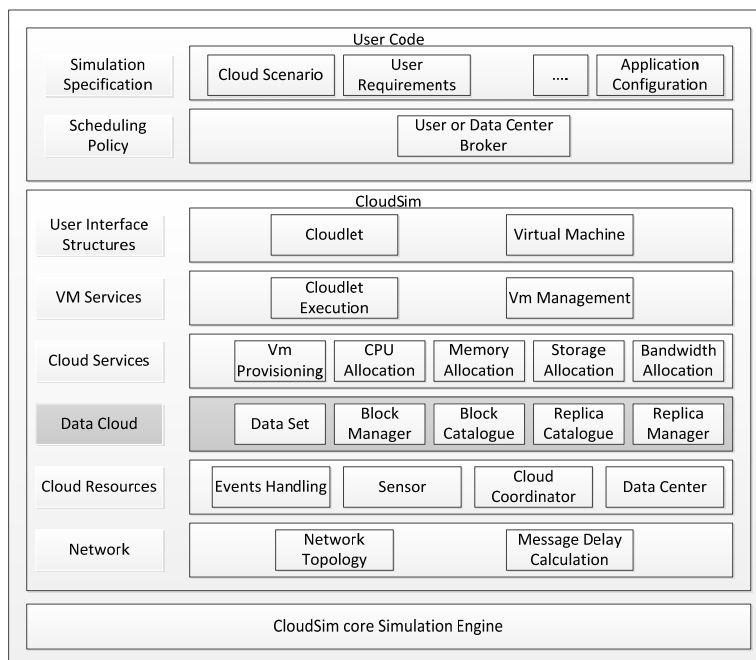
واسط ابر یا واسط مراکز داده مسئولیت مذاکرات در مورد نیازمندی‌های کیفیت سرویس مطلوب بین خدمات نرم‌افزار به عنوان سرویس و فراهم‌کننده ابر را بر عهده دارد. واسط با سرویس اطلاعاتی ابر در ارتباط بوده و تصمیمات بر خط^{۳۱} برای تخصیص منابع و خدمات را با توجه به نیازمندی کاربران انجام می‌دهد. امکان توسعه واسط ابر برای پیاده‌سازی و ارزیابی سیاست‌های جدید برای توسعه‌دهندگان سرویس‌های ابری وجود دارد. در این مقاله واسط ابر توسعه داده شده و قابلیت مدیریت داده و کپی به آن افزوده شده است.

• کلادلت^{۳۲}:

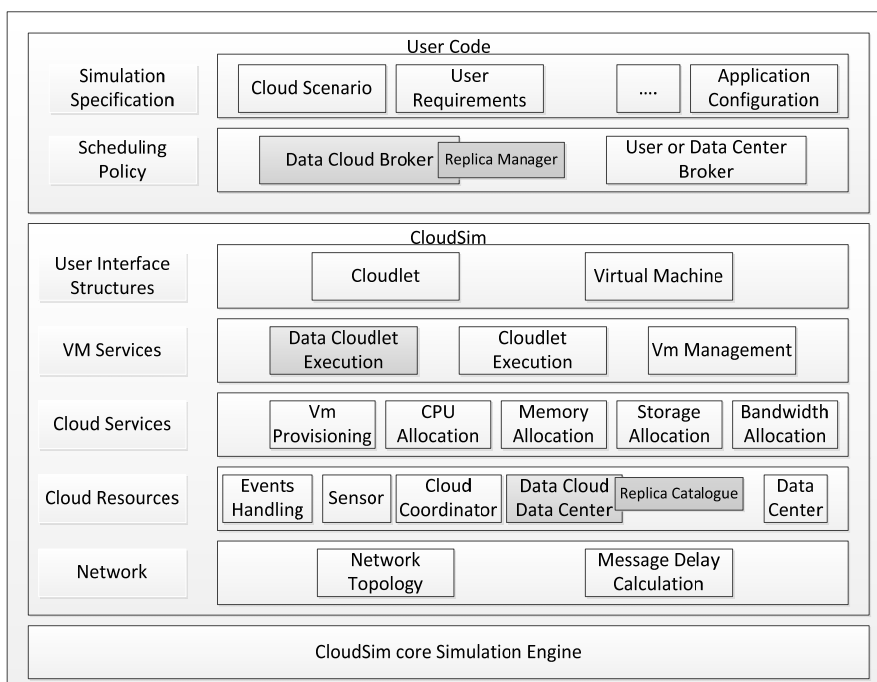
کلادلت مدلی برای سرویس‌های کاربردی مبتنی بر ابر مانند تحویل محتوا، شبکه‌های اجتماعی و گردش کار تجاری است. کلادلت درخواستی است که از سوی کاربر برای اجرا به واسط ابر ارسال می‌شود. محققان می‌توانند با توسعه مدل کلادلت قابلیت مدل‌سازی برنامه‌های کاربردی مبتنی بر داده را نیز فراهم کنند [۲۲] که در این مقاله کلادلت به عنوان درخواست مبتنی بر داده توسعه داده شده است.

۳- ابزار شبیه‌ساز ابر داده‌ای ارائه شده (Data Cloud Sim)

با توجه به اینکه در کلاسیک قابلیت تخصیص منابع به درخواست‌های مبتنی بر داده و امکان مدیریت کپی در ابر داده‌ای وجود ندارد، ابزاری برای شبیه‌سازی ابر داده‌ای بر پایه کلاسیک ارائه شده است که قابلیت اجرای درخواست‌های مبتنی بر داده و توسعه سیاست‌های تخصیص منبع برای این درخواست‌ها را دارد. هم‌چنین مدیریت آگاهانه داده و کپی در ابر داده‌ای نیازمند اطلاعاتی در مورد تعداد و مکان کپی‌ها، میزان دسترس‌پذیری و هزینه ذخیره‌سازی هر داده است که در کلاسیک چنین امکانی فراهم نشده است. ابزار ارائه شده با فراهم نمودن امکان پرس‌وجو در مورد این اطلاعات، مدیریت آگاهانه داده و کپی را ممکن می‌سازد.



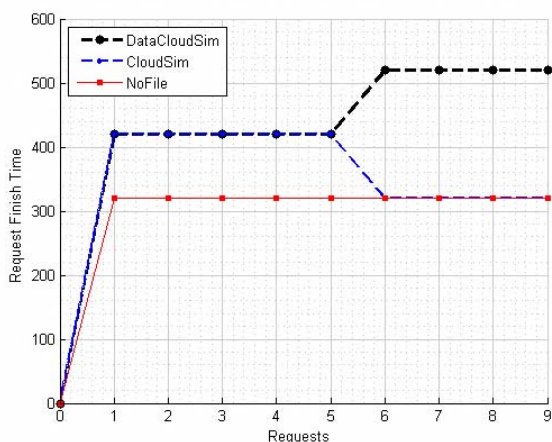
شکل ۳- معماری ابزار ارائه شده برای ابر ذخیره‌ساز در [۲۱]



شکل ۴- معماری ابزار جدید شبیه‌ساز ابر داده‌ای ارائه شده بر پایه کلاسیسیم (Data Cloud Sim) در این مقاله با قابلیت جدید مدیریت داده و کپی

ندارد؛ بنابراین بدون هیچ خطایی فرض می‌کند که کلادلت با موفقیت اجرایش به اتمام رسیده است. اما در Data Cloud Sim ابزار ارائه شده، همان طوری که انتظار می‌رفت کلادلت ۶ تا ۹ زمان اتمام بیشتری دارند زیرا که فایل ۲ دارای اندازه بیشتری نسبت به فایل ۱ است. بنابراین در ابزار جدید، با توسعه کلاس واسط مراکز داده و ایجاد موجودیت جدید واسط ابر داده‌ایدر قسمت ارسال کلادلت‌ها به روی ماشین‌های مجازی، عملیات زیر انجام می‌شود:

- پیدا کردن فایل مورد نیاز درخواست کاربر یا کلادلت
- پیدا کردن لیست مراکز داده که کپی از فایل مورد درخواست را دارند
- انتخاب مرکز داده مناسب با استفاده از سیاست‌های تعریف شده
- انتخاب ماشین مجازی مناسب مستقر در میزبان‌های آن مرکز داده
- ارسال کلادلت به روی آن ماشین مجازی برای اجرا و برگرداندن نتایج



شکل ۵- ضعف کلاسیسیم در اجرای درخواست‌هایی که نیاز به فایل دارند

در صورتی که کلادلتی برای اجرا، به مرکز داده دارای آن فایل فرستاده شود، اما به هر علتی موقع رسیدن کلادلت به مرکز داده، فایل پاک شده باشد، در این

- بستری برای پیاده‌سازی سیاست‌های جدید انتخاب داده مناسب برای جایگزینی کپی
- قابلیت مدیریت کپی
- قابلیت پیاده‌سازی سیاست‌های جدید برای مدیریت کپی و مضاعف‌سازی داده

شکل ۴ معماری ابزار جدید ارائه شده بر پایه کلاسیسیم در این مقاله را نشان می‌دهد. موجودیت‌های جدید اضافه‌شده به معماری با رنگ متفاوت نشان داده شده است که در ادامه توضیحات بیشتری در مورد موجودیت‌های جدید و نقش آن‌ها داده شده است.

۳-۲- افزودن قابلیت جدید اجرای درخواست‌های مبتنی بر داده به کلاسیسیم

درخواست مبتنی بر داده، درخواستی است که برای اجرای کار نیازمند داده است. به عبارت دیگر، کلادلت برای اجرای کامل و موفق نیاز به فایلی دارد؛ اما در کلاسیسیم در تخصیص منبع برای کلادلت، فایل مورد نیاز در نظر گرفته نشده است و به صورت ترتیبی کلادلت را به روی ماشین‌های مجازی موجود ارسال می‌کند. برای نمایش ضعف کلاسیسیم در اجرای درخواست‌های مبتنی بر داده، تعدادی کلادلت که نیاز به فایل دارند برای اجرا به واسط ابر در کلاسیسیم فرستاده شده است. شکل ۵ زمان اتمام کلادلت‌ها در کلاسیسیم و ابزار ارائه شده را نشان می‌دهد. کلادلت ۱ تا ۵ نیاز به فایل ۱ با اندازه ۱۰۰۰۰ (MB) و کلادلت ۶ تا ۹ نیاز به فایل ۲ با اندازه ۲۰۰۰۰ (MB) دارند. فایل ۱ در مرکز داده ۱ و فایل ۲ در مرکز داده ۲ ذخیره شده است.

منحنی No File مربوط به حالتی است که همان کلادلت‌ها نیاز به فایل ندارند. همان طوری که در شکل ۵ دیده می‌شود زمان اتمام کلادلت‌های ۶ تا ۹ در Cloud Sim برابر با حالتی است که نیاز به فایل ندارند. علت این است که کلادلت ۶ تا ۹ نیاز به فایل ۲ داشتند که چون کلاسیسیم بدون توجه به فایل مورد نیاز، این کلادلت‌ها را برای اجرا به مرکز داده ۱ ارسال کرده است و مرکز داده ۱، فایل ۲ را

دارد. برای ارتباطات جدید از برچسب‌های تعریف‌شده در کلاس Data Cloud Tags موجود در کلاسسیم استفاده شده است. در این کلاس تعداد زیادی برچسب تعریف شده است اما در کلاسسیم فعلی از این برچسب‌ها برای پردازش رخدادهای استفاده نشده است. برای ایجاد یک کپی از داده، واسط ابر داده، مرکز داده مبدأ و مقصد کپی را تعیین می‌کند و از موجودیت Cloud Sim همراه با کپی فایل از مرکز برچسب DataCloudTags.FILE_ADD_REPLICA همراه با کپی فایل از مرکز داده مبدأ به مرکز داده مقصد ارسال نماید. سپس در مرکز داده مقصد با دریافت این رخداد با برچسب جدید DataCloudTags.FILE_ADD_REPLICA اقدامی متناسب انجام شود. برای این منظور در قسمت پردازش رخدادهای مرکز داده، این برچسب و اقدام متناسب با آن تعریف شده است تا هرگاه رخدادی با این برچسب دریافت نمود، کپی فایل را در فضای ذخیره‌سازی خود قرار دهد و اطلاعات کاتالوگ کپی خود را بروزرسانی نماید.

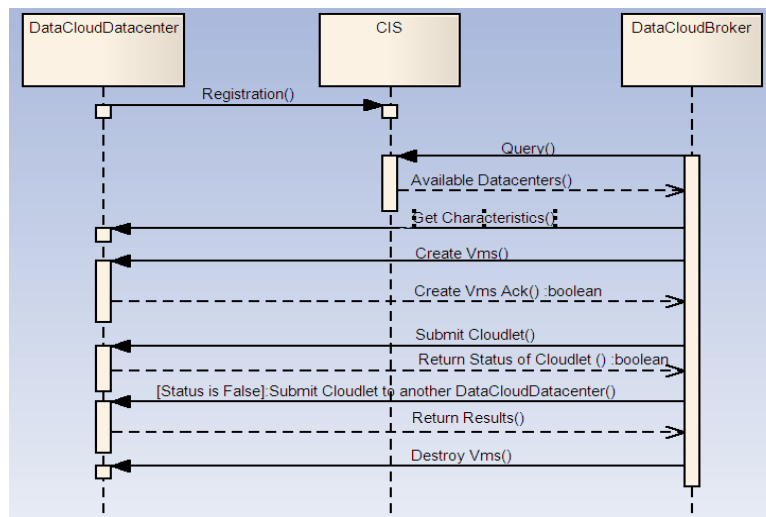
تعداد کپی از هر داده یا انتخاب مرکز داده مناسب برای ذخیره در آن از جمله تصمیمات مضاعف‌سازی داده است که با استفاده از اطلاعات مدیر کپی و کاتالوگ کپی اتخاذ می‌شود. زمانی که کپی داده از مرکز داده مبدأ به مقصد فرستاده می‌شود، اگر در مقصد فضای ذخیره‌سازی کافی در دسترس نباشد، به صورت تصادفی داده‌ای انتخاب و با آن جایگزین می‌گردد. امکان پیاده‌سازی سیاست‌های جدید برای انتخاب جایگزین مناسب نیز وجود دارد. در شکل ۵ نمودار توالی مربوط به این سناریو نمایش داده شده است.

صورت مرکز داده ابر داده‌ای پیغامی به واسط ابر داده ارسال می‌کند تا واسط دوباره منبع مناسب برای آن تخصیص داده و برای اجرا ارسال نماید. اگر درخواستی فایل مورد نظر برای اجرا را نداشته باشد، وضعیت آن درخواست ناموفق خواهد بود. شکل ۶ نمودار توالی مربوط به این سناریو را نشان می‌دهد. با بررسی وضعیت تمامی درخواست‌های ارسال شده برای اجرا در نتایج، می‌توان نتیجه گرفت که عملکرد ابزار صحیح می‌باشد.

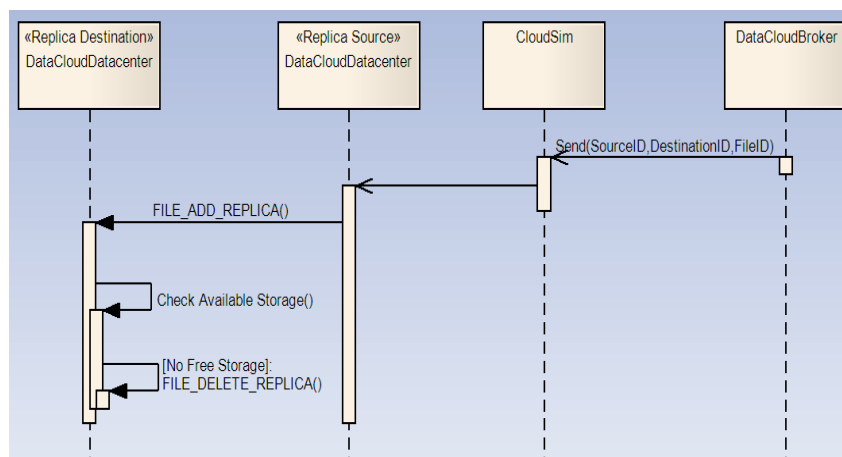
۳-۳- افزودن قابلیت جدید مدیریت داده و کپی در مراکز داده به کلاسسیم

نقش جدیدی که به واسط ابر داده اضافه شده است، مدیرکپی است. در هر مرکز داده ابر، یک کاتالوگ کپی قرار داده شده است که اطلاعات مربوط به کپی‌های موجود در آن مرکز داده را نگهداری می‌کند. مدیر کپی به صورت متناوب اطلاعات موجود در کاتالوگ کپی مراکز داده ابر را گرفته و اطلاعات خود را بروزرسانی می‌نماید. واسط ابر با استفاده از اطلاعات مدیر کپی از جمله تعداد کپی از هر داده، تعداد کل کپی‌ها و میزان دسترس‌پذیری داده‌ها، می‌تواند تصمیمات مضاعف‌سازی داده را با آگاهی بیشتری انجام دهد.

برای پیاده‌سازی امکان ذخیره کپی داده در مراکز داده مختلف، نیاز به ارتباط جدید بین واسط داده و مراکز داده و همچنین ارتباط مراکز داده با یکدیگر وجود



شکل ۶- نمودار توالی مربوط به سناریوی ارتباط واسط ابر داده و مرکز داده- زمانی که فایل موردنیاز در مرکز داده، بعد از ارسال پاک شده باشد



شکل ۷- نمودار توالی مربوط به سناریو ایجاد کپی و ارسال آن از مرکز داده مبدأ به مقصد

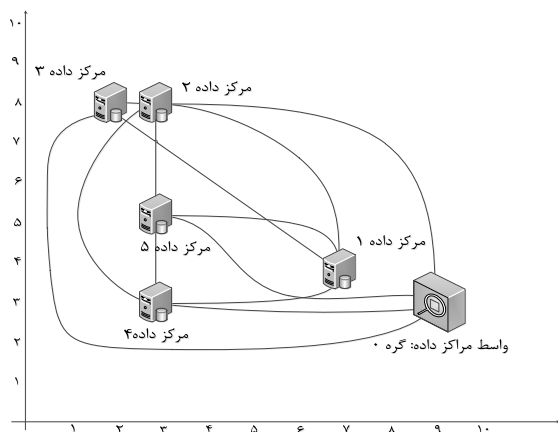
اجرای درخواست‌هایی باشد که به فایل ۱ نیاز دارند. منحنی Data Cloud Sim نشان می‌دهد که زمان اجرای درخواست‌های فایل ۱ به میزان ۱۰۰ ثانیه و درخواست‌های فایل ۲ به میزان ۲۰۰ ثانیه (دقیقا دو برابر حالت اول)، بیشتر از درخواست‌های بدون نیاز به فایل است. همچنین برای تست عملکرد درست ابزار، الگوریتم مضاعف‌سازی ایستای سه کپی به عنوان الگوریتم محک شبیه‌سازی شده است و در هر بخش نشان داده شده است که نتایج حاصل شده صحیح می‌باشد.

۴- شبیه‌سازی و نتایج

برای بررسی نحوه عملکرد ابزار ارائه شده برای شبیه‌سازی ابر داده‌ای، محیطی با ۵ مرکز داده با همبندی نشان داده شده در شکل ۸ را در نظر می‌گیریم. همبندی نشان داده شده توسط ابزاری به نام BRITE [۲۳]، ابزار ایجادکننده همبندی شبیه همبندی اینترنت، تولید شده است که از مدل Waxman با پارامترهای پیش‌فرض آلفا برابر ۰.۱۵ و بتا برابر ۰.۲، با قراردادی تصادفی گره‌ها و توزیع ثابت پهنای باند بین پیوندها استفاده کرده است. همبندی شامل ۶ گره و ۱۱ یال است. نحوه نمایش همبندی قالب brite در دو قسمت گره و یال است که در جدول ۱ نشان داده شده است. اعداد در قسمت گره به ترتیب نشان‌دهنده شناسه گره، موقعیت x، موقعیت y، درجه ورودی، درجه خروجی، شناسه AS، نوع گره و در قسمت یال به ترتیب نشان‌دهنده شناسه یال، گره مبدأ، گره مقصد، فاصله هندسی، تأخیر پیوند، پهنای باند پیوند، AS مبدأ، AS مقصد است. این قالب قابل خواندن با کلاسیسم می‌باشد.

برای اینکه انواع متفاوتی از مراکز داده در شبیه‌سازی وجود داشته باشد، دو نوع مرکز داده با نام‌های قوی و ضعیف تعریف و استفاده شده است که مشخصات هر یک از مراکز داده در جدول ۲ قابل مشاهده است. مشخصاتی مثل سیستم‌عامل^{۲۳}، نام فوق ناظر^{۲۴}، حافظه میزبان^{۲۵} و پهنای باند بر اساس مقادیر پیش‌فرض موجود در کلاسیسم تعریف شده است. مقادیر مشخصات دیگر مثل پهنای باند و تأخیر شبکه SAN با استفاده از اطلاعات موجود در [۲۴] مقدار تأخیر شبکه SAN، ۰.۱ تا ۰.۰۱ ثانیه و پهنای باند آن ۱۰۰ MB/s تعریف شده است. ۲ مرکز داده از نوع قوی و ۳ مرکز داده از نوع ضعیف در نظر گرفته شده است.

جدول ۳ تعداد ماشین‌های مجازی، مراکز داده و فایل‌های ذخیره شده در مراکز داده را نشان می‌دهد. مشخصات ماشین‌های مجازی و جدول ۵ مشخصات کلادلت‌ها را نشان می‌دهد. در ماشین مجازی برای زمان‌بندی کلادلت‌ها از الگوریتم زمان اشتراکی تعریف شده در کلاسیسم استفاده شده است.



شکل ۸- همبندی مراکز داده و واسط مراکز داده

۴-۴- افزودن قابلیت پرس‌وجو در مورد هزینه مضاعف‌سازی مراکز داده، هزینه ذخیره‌سازی و دسترس‌پذیری داده‌ها در حین اجرا

ابزار ارائه شده این امکان را فراهم می‌کند که هزینه مضاعف‌سازی هر مرکز داده و هزینه ذخیره‌سازی و دسترس‌پذیری هر فایل ذخیره شده در مراکز داده در حین اجرا محاسبه شده تا با استفاده از این اطلاعات، واسط ابر بتواند تصمیمات مضاعف‌سازی آگاهانه تری را با هدف کاهش هزینه یا زمان اجرای درخواست‌ها اتخاذ نماید. همچنین این اطلاعات به عنوان خروجی نمایش داده می‌شود. هزینه مضاعف‌سازی، هزینه صرف شده برای پهنای باند جهت کپی داده از مرکز داده مبدأ به مقصد مورد نظر است. رابطه ۱ نحوه محاسبه هزینه مضاعف‌سازی برای یک مرکز داده را نشان می‌دهد که برابر است با حاصل ضرب هزینه به ازای واحد پهنای باند در مرکز داده در اندازه فایل (FS) و هزینه انتقال فایل (TC).

$$replicationCost_{Datacenter} = CostPerBW_{Datacenter} * FS * TC \quad (1)$$

هزینه ذخیره‌سازی هر فایل برابر است با حاصل ضرب هزینه واحد ذخیره‌سازی در مرکز داده‌ای که در آن ذخیره شده است در اندازه فایل (FS) و تعداد کل کپی‌های آن فایل (NR) که از رابطه ۲ محاسبه می‌شود.

$$storageCost_{File} = CostPerStorage_{Datacenter} * FS * NR \quad (2)$$

دسترس‌پذیری داده معیار مهم دیگری است که در تصمیم‌گیری مدیریت داده و کپی در نظر گرفته می‌شود. هر چه تعداد کپی بیشتری از داده موجود باشد، دسترس‌پذیری داده افزایش می‌یابد؛ بنابراین مصالحه‌ای بین هزینه ذخیره‌سازی و دسترس‌پذیری داده وجود دارد. رابطه ۳ نحوه محاسبه دسترس‌پذیری یک فایل را نشان می‌دهد. ρ نشان‌دهنده نرخ شکست مرکز داده‌ای است که فایل در آن ذخیره شده است و NR_f نشان‌دهنده تعداد کپی‌های موجود از فایل مورد نظر هست.

$$Availability_{f_n} = 1 - \rho^{NR_f} \quad (3)$$

۴-۵- اعتبارسنجی ابزار ارائه شده

دقت یک ابزار شبیه‌ساز و اعتبار نتایج آن تا حد زیادی به دقت جزئیات اجرای شبیه‌سازی شده در آن ابزار وابسته است. برای شبیه‌سازی رفتار زیرساخت ابر با صحت و دقت بالا، استفاده از اجزای اساسی مانند مدل کاربردی عمومی و مدل شبکه مراکز داده، در ابزارهای شبیه‌ساز ابر امری ضروری است [۱۱] و با توجه به اینکه ابزار ارائه شده در این مقاله بر پایه کلاسیسم بوده که چارچوبی کلی و توسعه‌پذیر برای مدل‌سازی و شبیه‌سازی ابر است و معماری کلاسیسم تغییر داده نشده است، بنابر این نتایج حاصل قابل اتکا خواهد بود.

مهم‌ترین بخش اعتبارسنجی ابزار ارائه شده مربوط به اجرای درخواست‌های مبتنی بر داده است که پایه‌ای برای بقیه قابلیت‌های افزوده شده است. همان‌طور در شکل ۵ نیز نشان داده شده است، درخواست‌هایی که برای اجرا نیاز به فایل ندارند دارای زمان اتمام ۳۲۰ ثانیه هستند. با وجود شرایط یکسان در مراکز داده انتظار می‌رود زمان اجرای درخواست‌هایی که به فایل ۲ نیاز دارند، دو برابر زمان

جدول ۱- همبندی استفاده شده در شبیه‌سازی

Topology: (۶ Nodes, ۱۱ Edges)										
Model (۱ - RTWaxman): ۶۱۰۱۰۱۲۰.۱۵۰.۲۱۱۱۰۰۱۰۲۴۰۰										
Nodes: (۶)										
۰	۹	۳	۴	۴	-۱	RT_NONE				
۱	۷	۴	۴	۴	-۱	RT_NONE				
۲	۳	۸	۴	۴	-۱	RT_NONE				
۳	۲	۸	۳	۳	-۱	RT_NONE				
۴	۳	۳	۴	۴	-۱	RT_NONE				
۵	۰	۵	۳	۳	-۱	RT_NONE				
Edges: (۱۱)										
۰	۲	۱	۵.۶۵۶۸۵۴	۰.۱	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۱	۲	۰	۷.۸۱۰۲۵	۰.۲	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۲	۳	۲	۱.۰	۰.۱	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۳	۳	۱	۶.۴۰۳۱۲۴	۰.۵	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۴	۴	۱	۴.۱۲۳۱۰۵	۰.۲	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۵	۴	۲	۵.۰	۰.۳	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۶	۵	۴	۳.۶۰۵۵۵۱	۰.۲	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۷	۵	۰	۹.۲۱۹۵۴۴	۰.۴	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۸	۰	۴	۶.۰	۰.۹	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۹	۰	۳	۸.۶۰۲۳۲۵	۰.۸	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	
۱۰	۱	۵	۷.۰۷۱۰۶۸	۰.۵	۱۰۰۰	-۱	-۱	E_RT_BACKBONE	U	

جدول ۲- مشخصات مراکز داده استفاده شده در شبیه‌سازی

SAN Capacity (MB)	SAN BW (MB/s)	SAN Delay (ms)	VMM	OS	Host Storage (MB)	BW (MB/s)	Host CPU	MIPS	Hosts	RAM (MB)	مرکز داده
۲۰۰۰۰۰۰	۱۰۰	۱۰ ^{-۲}	Xen	Linux	۱۰۰۰۰۰	۱۰۰۰۰	quad-core	۳۰۰۰۰	۱۰	۲۰۴۸	قوی
۲۰۰۰۰۰۰	۱۰۰	۱۰ ^{-۳}	Xen	Linux	۱۰۰۰۰۰	۱۰۰۰۰	dual-core	۱۰۰۰۰	۱۰	۲۰۴۸	ضعیف

به صورت پیش‌فرض در سیستم‌های تجاری ابر مثل آمازون S3، سیستم فایل گوگل و سیستم فایل توزیع شده هدوپ استفاده می‌شود [۲۸]. در هر سه الگوریتم، ابتدا داده‌های اصل با توزیع یکنواخت بین مراکز داده پخش و ذخیره می‌شوند که کپی اصل نامیده می‌شوند. سپس از هر داده، یک یا دو کپی ایجاد و با توزیع یکنواخت بین مراکز داده توزیع می‌گردد به طوری که هیچ مرکز داده‌ای دو کپی یکسان از داده را نداشته باشد. همچنین فضای کافی برای ذخیره‌سازی کپی جدید را داشته باشد تا کپی‌های قبلی از مراکز داده پاک نگردد و حالت ایستا بودن و تعداد ثابت کپی پایدار بماند.

به عبارت بهتر در حالت یک کپی، تنها یک کپی اصل از هر داده وجود دارد و در حالت سه کپی، از هر داده یک کپی اصل و دو کپی دیگر نیز ایجاد می‌شود. همچنین برای سبب فایل‌های ذخیره‌شده در مراکز داده مشابه [۲۱]، از توزیع پرتو^{۴۳} استفاده شده است که در شکل ۹ نمایش داده شده است.

جدول ۳- مشخصات شبیه‌سازی

مشخصات شبیه‌سازی	
تعداد ماشین‌های مجازی	۱۴۰
تعداد فایل‌ها	۱۰
تعداد مراکز داده	۵

هر کلادلت فرض شده است که به یک فایل برای اجرا شدن نیاز دارد که نام فایل درخواستی باید مشخص باشد. از مشخصات کلادلت، اندازه فایل^{۴۶} و اندازه خروجی^{۴۷} بر حسب بایت، به ترتیب نشان‌دهنده اندازه کلادلت بعد و قبل از اجرا شدن است که متفاوت با فایل مورد نیاز کلادلت است و از اندازه پیش‌فرض موجود در کلادسیم ۳۰۰ بایت استفاده شده است. طول کلادلت بر حسب میلیون دستورالعمل در ثانیه^{۴۹} نشان‌دهنده تعداد دستورالعمل لازم برای اجرا شدن است.

در تمام سناریوهای شبیه‌سازی از هزینه‌های موجود در جدول ۱ برای محاسبه هزینه ذخیره‌سازی و مضاعف‌سازی استفاده شده است. هزینه ذخیره‌سازی ذکر شده در جدول ۶ با استفاده از مقادیر پیش‌فرض موجود در کلاس‌های پایه کلادسیم تعیین شده است. مجموعه داده مورد نیاز برای آزمون عملکرد ابزار ارائه شده، شامل نحوه توزیع دسترسی به فایل‌ها و نرخ دسترسی به هر فایل است.

توزیع دسترسی به فایل‌ها در [۲۵] و [۲۶] توزیع پواسون^{۴۰} فرض شده است و با توجه به این که نرخ دسترسی به فایل‌ها در صفحات اینترنتی از توزیع زیپف^{۴۱} پیروی می‌کند [۲۷]، در سناریوی اول شبیه‌سازی توزیع دسترسی به فایل‌ها، پواسون و توزیع نرخ دسترسی به فایل‌ها، زیپف فرض شده است. در سناریوی دوم شبیه‌سازی از مجموعه داده واقعی رسانه ذخیره‌سازی ویکی مدیا^{۴۲} استفاده شده است.

برای تست عملکرد درست شبیه‌ساز، سه الگوریتم ایستای یک کپی، دو کپی و سه کپی به عنوان الگوریتم محک پیاده‌سازی شده است. الگوریتم ایستای سه کپی

جدول ۴- مشخصات ماشین‌های مجازی استفاده شده در شبیه‌سازی

RAM (MB)	BW (MB)	MIPS	VMM name	Number of CPUs	Cloudlet Scheduler	Size (MB)
۵۱۲	۱۰۰۰	۱۰۰۰	Xen	۱	Timeshared	۱۰۰۰۰

۴-۱- مدیریت داده و کپی

ابزار ارائه شده، اطلاعاتی در مورد تعداد دقیق کپی از هر فایل، مکان دقیق کپی‌های موجود و میزان دسترس‌پذیری هر فایل را به عنوان خروجی تولید می‌کند. به طور مثال یک خروجی از اجرای الگوریتم سه کپی در جدول ۷ نمایش داده شده است. این اطلاعات در زمان اجرا نیز قابل دسترسی است.

جدول ۵- مشخصات هر کلادلت استفاده شده در شبیه‌سازی

Output Size (Byte)	File Size (Byte)	Number of CPUs	Number of Required File	Length (M)
۳۰۰	۳۰۰	۱	۱	۱۰۰۰

جدول ۷- اطلاعات مدیریت داده و کپی در الگوریتم سه کپی

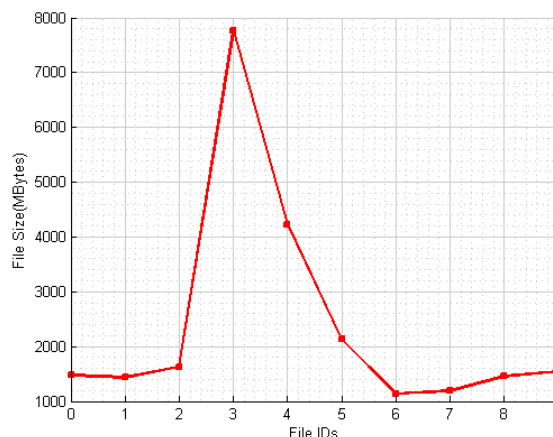
شناسه فایل	سایز فایل (MB)	شناسه مراکز داده	دسترس‌پذیری	تعداد کپی‌ها
۰	۱۴۷۲	[۲و۳و۴]	۹۹.۹۹۹۹	۳
۱	۱۴۳۷	[۵و۴و۳]	۹۹.۹۹۹۹	۳
۲	۱۶۲۹	[۴و۳و۵]	۹۹.۹۹۹۹	۳
۳	۷۷۵۵	[۳و۵و۴]	۹۹.۹۹۹۹	۳
۴	۴۲۳۳	[۶و۴و۵]	۹۹.۹۹۹۹	۳
۵	۲۱۳۳	[۴و۵و۶]	۹۹.۹۹۹۹	۳
۶	۱۱۳۵	[۴و۶و۵]	۹۹.۹۹۹۹	۳
۷	۱۱۹۱	[۲و۵و۴]	۹۹.۹۹۹۹	۳
۸	۱۴۵۷	[۵و۴و۲]	۹۹.۹۹۹۹	۳
۹	۱۵۴۸	[۶و۲و۴]	۹۹.۹۹۹۹	۳

جدول ۶- هزینه‌های استفاده شده در شبیه‌سازی

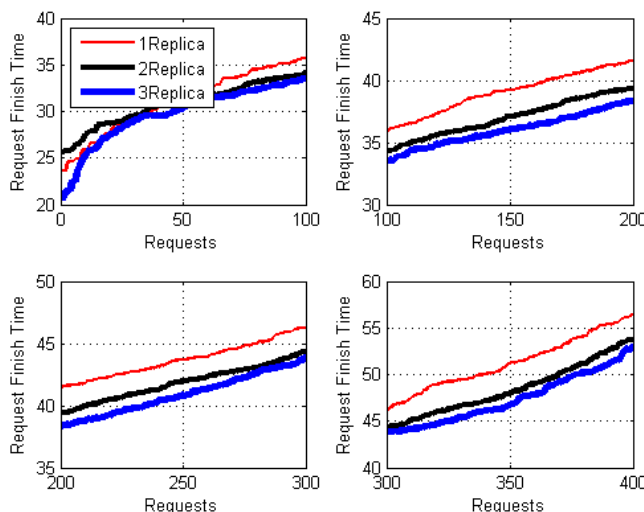
هزینه ذخیره‌سازی	هزینه انتقال داده	هزینه پهنای باند
۰.۱ \$ PerMB	۱ \$ PerMB	۰.۱ \$ PerMB/s

۴-۲- زمان اتمام کلادلت

انتظار می‌رود هر چه تعداد کپی بیشتری از فایل‌ها موجود باشد، زمان دسترسی به داده‌ها کاهش یافته و در نتیجه زمان اتمام کلادلت کاهش یابد. در ابتدای شبیه‌سازی، در هر سه الگوریتم ابتدا فقط یک کپی اصل از داده‌ها وجود دارد و بعد از دوره تناوب اول کپی دوم و یا سوم از داده‌ها در مراکز داده ذخیره می‌شود. همان طوری که در شکل ۱۰ دیده می‌شود، بعد از ۲۰ درخواست اول، الگوریتم ایستای سه کپی، نسبت به الگوریتم دو کپی و الگوریتم دو کپی نسبت به الگوریتم یک کپی، زمان اتمام کلادلت کمتری داشته است. برای نمایش بهتر، نمودار به ازای هر ۱۰۰ درخواست، در نمای بزرگ‌تر رسم شده است.



شکل ۹- نمودار سایز فایل با توزیع پرتو



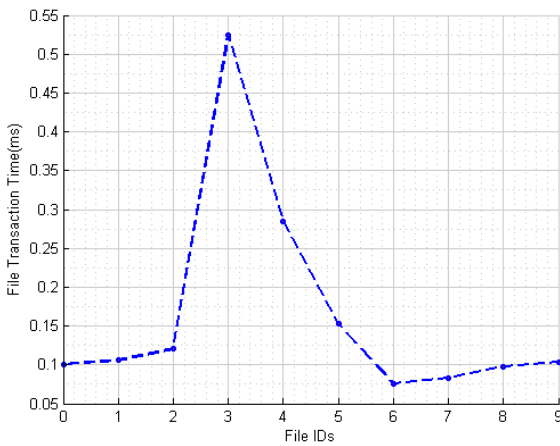
شکل ۱۰- نمودار زمان اتمام کلادلت سه الگوریتم ۱ کپی، ۲ کپی و ۳ کپی

۳-۴- هزینه ذخیره‌سازی

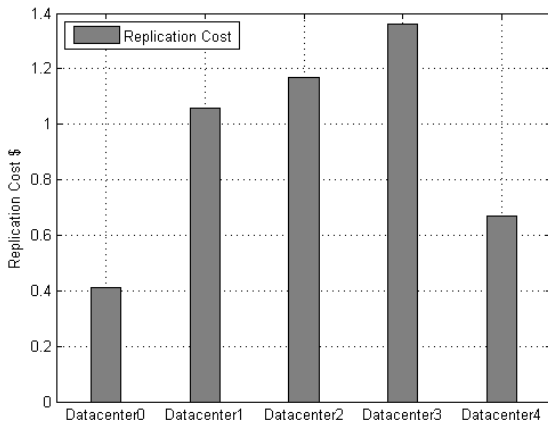
همان طوری که دیده می‌شود هزینه ذخیره‌سازی هر فایل با توجه به مکان ذخیره‌سازی آن و هزینه آن مرکز داده دارای هزینه ذخیره‌سازی متفاوتی می‌باشد. به عنوان مثال، فایل شماره ۵ در هر سه حالت در مرکز داده‌ای ذخیره شده است که در هر سه دارای هزینه یکسانی بوده است، بنابراین هزینه ذخیره‌سازی این فایل ثابت مانده است.

۴-۴- زمان تراکنش فایل

زمان تراکنش فایل با اندازه فایل ارتباط مستقیم دارد هر چه فایل دارای اندازه بیشتری باشد، دارای زمان تراکنش بیشتری خواهد بود. همان طور که در شکل ۱۳ دیده می‌شود، زمان تراکنش نیز مشابه نمودار اندازه فایل است که خود این مطلب نیز دلیل دیگری بر اعتبار عملکرد صحیح ابزار ارائه شده است.



شکل ۱۳- زمان بازیابی فایل



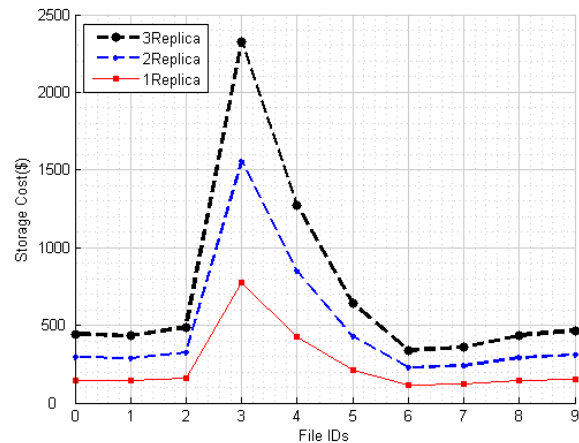
شکل ۱۴- هزینه مضاعف‌سازی هر مرکز داده

۵-۴- هزینه مضاعف‌سازی

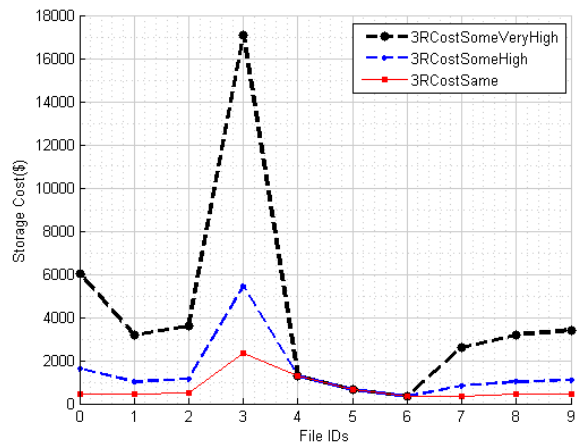
در شکل ۱۴ هزینه مضاعف‌سازی هر مرکز داده نشان داده شده است که بر حسب تعداد فایل‌ها و اندازه فایل‌های ذخیره شده در آن، هزینه هر مرکز داده متفاوت بوده است. با توجه به نمودار می‌توان فهمید که در مرکز داده سوم، فایل بیشتری یا فایل با اندازه بزرگ‌تری ذخیره شده است.

در شکل ۱۱ نمودار هزینه ذخیره‌سازی هر فایل به ازای هر سه الگوریتم ایستای سه کپی، دو کپی و یک کپی نشان داده شده است. براساس رابطه ۲ هزینه ذخیره‌سازی با اندازه فایل رابطه مستقیم دارد، هر چه فایل دارای اندازه بیشتری باشد بنابراین هزینه ذخیره‌سازی بیشتری خواهد داشت. در سناریوی بکار برده شده در شکل ۱۱، هزینه ذخیره‌سازی در تمام مراکز داده یکسان فرض شده است، به همین دلیل هزینه ذخیره‌سازی تنها تابعی از اندازه فایل است. همین طور که قابل پیش‌بینی بود، هر چه تعداد کپی‌ها بیشتر باشد، هزینه ذخیره‌سازی نیز بیشتر می‌شود که شکل ۱۱ نیز همین مطلب را تأیید می‌کند. دلیل دیگر بر اعتبار نتایج حاصل از ابزار ارائه شده این است که هزینه ذخیره‌سازی هر فایل در الگوریتم سه کپی دقیقاً سه برابر هزینه آن در الگوریتم یک کپی است.

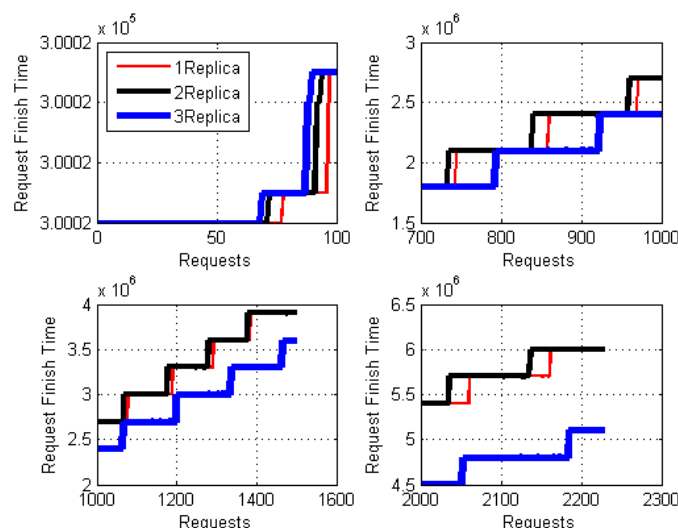
در سناریوی دیگر، هزینه ذخیره‌سازی در مراکز داده متفاوت فرض شده و هزینه ذخیره‌سازی هر فایل با الگوریتم سه کپی در سه حالت مختلف در شکل ۱۲ نشان داده شده است. 3R Cost same حالتی است که مراکز داده دارای هزینه یکسانی برای ذخیره‌سازی هستند. 3R Cost Some High حالتی است که تعدادی از مراکز داده دارای هزینه بیشتر و 3R Cost Some Very High حالتی که همان مراکز داده دارای هزینه خیلی بیشتری هستند.



شکل ۱۱- نمودار هزینه ذخیره‌سازی هر فایل در سه الگوریتم ۱ کپی، ۲ کپی و ۳ کپی



شکل ۱۲- نمودار هزینه ذخیره‌سازی هر فایل در سه الگوریتم ۱ کپی، ۲ کپی و ۳ کپی در سناریوی با هزینه ذخیره‌سازی متفاوت در مراکز داده



شکل ۱۵- نمودار زمان اتمام درخواست‌های رسانه ذخیره‌سازی ویکی‌مدیا، در بازه‌های تصادفی مختلف

تخصیص منابع ابری بر روی بسترهای واقعی ابر مورد بررسی قرار گرفت و سپس شبیه‌سازهای موجود برای سیستم‌های توزیع شده گرید و ابر معرفی شد. شبیه‌ساز ابر به نام کلاسیسم که چارچوب کلی برای شبیه‌سازی محیط‌های ابری است با جزئیات بیشتر معرفی و نحوه کار آن نیز بیان شد. با وجود چالش‌های مطرح شده، چند مشکل اساسی برای شبیه‌سازی الگوریتم‌های مضاعف‌سازی داده در کلاسیسم مطرح شد که عبارتند از نبود قابلیت اجرای درخواست‌های مبتنی بر داده و نحوه تخصیص منبع به این نوع درخواست‌ها، نبود امکان مدیریت داده و کپی و عدم ارائه اطلاعات لازم برای پاسخگویی به مسائل مطرح شده در مضاعف‌سازی داده، ارائه الگوریتم‌های نوین مضاعف‌سازی داده و ارزیابی برنامه‌های کاربردی ابر با استفاده از مدل پرداخت مبتنی بر استفاده. بنابراین در این مقاله، بستر کلاسیسم با هدف حل مشکلات مطرح شده‌ی فوق گسترش داده شده تا ابزاری برای شبیه‌سازی ابر داده‌ای بر پایه کلاسیسم ارائه شود تا بدین وسیله بستر لازم برای مدیریت آگاهانه داده و کپی در مراکز داده ابری فراهم گردد.

ابزار جدید ارائه شده دارای قابلیت اجرای درخواست‌های مبتنی بر داده، ذخیره چند کپی از داده در مراکز داده متفاوت، امکان پرس‌وجو در مورد تعداد کپی از هر داده و مکان کپی‌ها، امکان پرس‌وجو در مورد میزان دسترسی‌پذیری و هزینه ذخیره‌سازی هر داده و هزینه مضاعف‌سازی هر مرکز داده است. همچنین امکان مدیریت کپی و پیاده‌سازی سیاست‌های جدید برای مدیریت کپی‌ها در ابزار جدید ارائه شده است. مضاعف‌سازی داده یکی از مهم‌ترین مکانیزم‌های مدیریت داده در سیستم‌های توزیع شده است که تصمیم‌گیری در مورد انتخاب داده مناسب برای کپی، زمان ایجاد کپی، تعداد کپی از هر داده و مکان کپی از جمله مباحثی است که به عنوان کارهای آینده می‌تواند در نظر گرفته شود که با استفاده از آن می‌توان سیاست‌های جدید مضاعف‌سازی داده در مراکز داده ابری و سیاست‌های جایگزینی کپی ارائه نموده و با استفاده از این ابزار، کارایی الگوریتم‌های پیشنهادی جدید را مورد ارزیابی قرار داده و بهبود بخشید. همچنین گسترش سیاست‌های تخصیص منابع به درخواست‌های مبتنی بر داده از مباحثی است که محققان می‌توانند سیاست‌های نوین خود را با این ابزار ارزیابی کنند.

با گسترش ابزار ارائه شده بر پایه کلاسیسم می‌توان قابلیت‌های جدید دیگری مثل تقسیم فایل به بلوک، مدیریت بلوک و کپی‌ها را اضافه نمود تا نتایج واقعی‌تری از ارزیابی کارایی سیاست‌های مضاعف‌سازی داده بدست آورد. با استفاده از اطلاعاتی که این ابزار در حین اجرا به واسطه مراکز داده ابری به عنوان مدیر کپی‌ها ارسال می‌نماید، می‌توان سیاست‌های مضاعف‌سازی آگاهانه‌تری پیاده‌سازی

می‌توان با استفاده از این اطلاعات، ماشین‌های مجازی بیشتری در این مرکز داده فراهم نمود تا درخواست‌ها با سرعت بیشتری پاسخ داده شوند. از موارد دیگر کاربرد اطلاعاتی مثل هزینه ذخیره‌سازی و مضاعف‌سازی در محاسبه هزینه درخواست‌های صورت گرفته برای یک برنامه کاربردی خاص است تا از مدل پرداخت مبتنی بر استفاده پیروی کند.

برای اطمینان از عملکرد درست ابزار ارائه شده، در سناریوی دیگری از مجموعه داده ترکیبی از داده‌های واقعی موجود در رسانه ذخیره‌سازی ویکی‌مدیا استفاده شده است. رسانه ذخیره‌سازی ویکی‌مدیا، امکان ذخیره و بارگیری محتواهای مختلف مثل عکس، فایل پی‌دی‌اف، فایل صوتی و فایل تصویری را از سایت upload.wikimedia.org فراهم می‌کند. Ganglia نام ابزار نظارتی سیستم توزیع شده مقیاس‌پذیر برای سیستم‌های محاسباتی مثل گرید و خوشه است. این ابزار به کاربران اجازه می‌دهد تا به اطلاعات آماری همه ماشین‌های تحت نظارت به صورت زنده یا اطلاعات قدیمی دسترسی داشته باشند. مجموعه داده مورد استفاده مربوط به آمار درخواست‌های صورت گرفته از سایت ذخیره‌سازی ویکی‌مدیا از ساعت ۴:۴۰ تا ۶:۴۰ تاریخ ۱۹-۱۲-۲۰۱۳ است [۲۹] که شامل اطلاعات درخواستی ۲۷ ماشین نظارت شده است. از آنجایی که درخواست‌ها در محیط شبیه‌سازی کلاسیسم به واسطه مراکز داده فرستاده می‌شود، متوسط درخواست‌های ۵ ماشین اول، به عنوان درخواست‌های ورودی شبیه‌سازی در نظر گرفته شده است. آماری در مورد نرخ دسترسی به فایل به تفکیک نام در این سایت در دسترس نبود، به همین علت از توزیع تصادفی برای دسترسی به فایل‌ها استفاده شده است. شکل ۱۵ نمودار زمان اتمام بیش از ۲۰۰۰ درخواست صورت گرفته از رسانه ذخیره‌سازی ویکی‌مدیا در بازه زمانی دو ساعت را نشان می‌دهد. از آنجایی که تعداد درخواست‌ها زیاد بود و نمایش زمان اتمام همه درخواست‌ها امکان‌پذیر نبود، به صورت تصادفی چهار بازه مختلف از درخواست‌ها در نمای بزرگتر در شکل ۱۵ نمایش داده شده است. همان طوری که قابل پیش‌بینی بود، الگوریتم سه کپی نسبت به الگوریتم دو کپی و الگوریتم دو کپی نسبت به الگوریتم یک کپی بعد از ۱۰۰ درخواست اول، زمان اتمام کمتری داشته است.

۵- نتیجه‌گیری

در این مقاله ابتدا چالش‌های موجود در ارزیابی کارایی سیاست‌های زمان‌بندی و

[12] R. N. Calheiros, M. A. Netto, C. A. De Rose, and R. Buyya, "EMUSIM: An Integrated Emulation and Simulation Environment for Modeling, Evaluation, and Validation of Performance of Cloud Computing Applications," *Journal of Software: Practice and Experience*, vol. 40, no. 4, pp. 1-18, 2012.

[13] S. H. Lim, B. Sharma, G. Nam, E. K. Kim, and C. R. Das, "MDCSim: A Multi-tier Data Center Simulation, Platform," *Proc. IEEE Intl Conf. Cluster Computing and Workshops*, pp. 1-9, 2009.

[14] R. Malhotra, and P. Jain, "Study and Comparison of CloudSim Simulators in the Cloud Computing," *IEEE Trans. Computer Science Engineering and its Applications*, vol. 1, no. 4, pp. 407-419, 2013.

[15] Cloud Reports, <https://github.com/thiagotts/CloudReports>, January 2014.

[16] Work flow Sim, <https://github.com/WorkflowSim>, January 2014.

[17] M. Bux, and U. Leser, "DynamicCloudSim: Simulating Heterogeneity in Computational Clouds," *Proc. ACM Intl Workshop on Scalable Workflow Execution Engines and Technologies*, pp. 1-12, 2013.

[18] Real Cloud Sim, <http://sourceforge.net/projects/realcloudsim>, January 2014.

[19] Cloud Simulation Tools, <http://www.cloudbus.org/cloudsim>, January 2014.

[20] Cloud MIG Xpress, <http://sourceforge.net/projects/cloudmigxpress>, January 2014.

[21] S. Long, and Y. Zhao, "A Toolkit for Modeling and Simulating Cloud Data Storage: An Extension to CloudSim," *Proc. IEEE Intl Conf. Control Engineering and Communication Technology*, pp. 597-600, 2012.

[22] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities," *Proc. IEEE Intl Conf. High Performance Computing and Simulation*, pp. 1-11, 2009.

[23] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An Approach to Universal Topology Generation," *Proc. IEEE Intl Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 346-353, 2001.

[24] SAN Bandwidth, and SAN Delay, <http://www.allsan.com>, January 2013.

[25] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "CDRM: A Cost-effective Dynamic Replication Management Scheme for Cloud Storage Cluster," *Proc. IEEE Intl Conf. Cluster Computing*, pp. 188-196, 2010.

نمود و مورد ارزیابی قرار داد. همچنین می توان با طراحی واسط گرافیکی کاربر استفاده از این ابزار را راحت تر نمود. ابزارهای دیگر ارائه شده بر پایه کلاسیسم مثل کلاسیسم و کلاسیسم را نیز می توان گسترش داد تا بتوان از این ابزارها هم برای تحلیل و گزارش گیری از مراکز داده ابری با درخواست های مبتنی بر داده نیز استفاده نمود.

مراجع

[1] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Journal of Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.

[2] D. R. Malhotra, and P. Jain, "An EMUSIM Techniques and its Components in a Cloud Computing Environment," *Journal of Computer Trends and Technology*, vol. 4, no. 8, pp. 2435-2440, 2013.

[3] R. Buyya, and M. Murshed, "Gridsim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," *Journal of Concurrency and Computation Practice and Experience*, vol. 14, no. 13-15, pp. 1175-1220, 2002.

[4] A. Legrand, L. Marchal, and H. Casanova, "Scheduling Distributed Applications: The SimGrid Simulation Framework," *Proc. IEEE/ACM Intl Symp. Cluster Computing and the Grid*, pp. 138-145, 2003.

[5] CL. Dumitrescu, and I. Foster, "GangSim: A Simulator for Grid Scheduling Studies," *Proc. IEEE Intl Symp. Cluster Computing and the Grid*, pp. 1151-1158, 2005.

[6] Amazon-S3, Amazon simple storage service, <http://www.amazon.com/s>, November 2013.

[7] S. Ghemawat, H. Gobioff, and S. Leung, "The Google File System," *Proc. ACM Intl Symp. Operating*, pp. 29-43, 2003.

[8] The Apache Software Foundation, <http://hadoop.apache.org/core>, November 2013.

[9] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications," *Proc. IEEE Intl Conf. Advanced Information Networking and Applications*, pp. 446-452, 2010.

[10] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: A Packet-level Simulator of Energy-aware Cloud Computing Data Centers," *Journal of Supercomputing*, vol. 62, no. 3, pp. 1263-1283, 2012.

[11] S. K. Garg, and R. Buyya, "NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations," *Proc. IEEE Intl Conf. Utility and Cloud Computing*, pp. 105-113, 2011.

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۲/۱۱/۲۶

تاریخ اصلاح: ۹۳/۴/۲۲

تاریخ قبول شدن: ۹۳/۵/۳۰

نویسنده مرتبط: رضا انتظاری ملکی، دانشکده مهندسی کامپیوتر، دانشگاه

صنعتی شریف، تهران، ایران.

[26] X. Xu, S. Wang, K. Yao, and X. Zhou, "Research on the Strategy of FLDC Replication Dynamically Created in Cloud Storage," *Proc. IEEE Intl Conf. Consumer Electronics, Communications and Networks*, pp. 2815-2818, 2012.

[27] A. L. Adami, and A. B. Huberman, "Zipf's Law and the Internet," *Glottometrics Journal*, vol. 3, no. 1, pp. 143-150, 2002.

[28] K. Sashi, and A.S. Thanamani, "A New Dynamic Replication Algorithm for European Data Grid," *Proc. ACM Intl Bangalore Conf.*, pp. 1-4, 2010.

[29] Data Set, <http://ganglia.wikimedia.org>, January 2013.

- ¹ Cloud Computing
- ² Pay-As-You-Go Model
- ³ Infrastructure As a Service (IaaS)
- ⁴ Platform As a Service (PaaS)
- ⁵ Software As a Service (SaaS)
- ⁶ Data replication
- ⁷ Amazon Simple Storage System (Amazon S3)
- ⁸ Google File System (GFS)
- ⁹ Hadoop Distributed File System (HDFS)
- ¹⁰ Data Cloud
- ¹¹ Replica Manager
- ¹² Replica Catalogue
- ¹³ Cluster Computing
- ¹⁴ Grid Computing
- ¹⁵ Time-Shared
- ¹⁶ Space-Shared
- ¹⁷ Packet-Level
- ¹⁸ Energy-Aware
- ¹⁹ Link
- ²⁰ High Performance Computing (HPC)
- ²¹ Topology
- ²² Automated Emulation Framework (AEF)
- ²³ Cloud Data Storage
- ²⁴ User Code
- ²⁵ VMs (Virtual Machines)
- ²⁶ Cloud Information Service (CIS)
- ²⁷ Future Queue
- ²⁸ Deferred Queue
- ²⁹ VM Allocation
- ³⁰ Storage Area Network (SAN)
- ³¹ On-Line
- ³² Cloudlet
- ³³ OS
- ³⁴ VMM
- ³⁵ RAM
- ³⁶ Size File
- ³⁷ Output Size
- ³⁸ Length
- ³⁹ MI
- ⁴⁰ Poisson Distribution
- ⁴¹ Zapf Distribution
- ⁴² Wikimedia's Media Storage
- ⁴³ Pareto Distribution



سعیده مهري مدرک کارشناسی خود را در رشته مهندسی فناوری اطلاعات در سال ۱۳۸۹ از دانشگاه تبریز و مدرک کارشناسی ارشد مهندسی فناوری اطلاعات با گرایش شبکه‌های کامپیوتری را در سال ۱۳۹۲ از دانشگاه صنعتی شریف دریافت نموده است. پایان نامه وی در مورد الگوریتم‌های مضاعف‌سازی داده در مراکز داده ابری با هدف بهبود کارایی است. زمینه‌های پژوهشی وی در زمینه محاسبات ابری، محاسبات گرید، الگوریتم‌های مضاعف‌سازی داده، تعادل بار و تخصیص منابع با هدف بهبود کیفیت سرویس می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

mehri@ce.sharif.edu



رضا انتظاری ملکی در حال حاضر دانشجوی دکتری مهندسی کامپیوتر (گرایش نرم‌افزار) در دانشگاه صنعتی شریف می‌باشد. وی مدرک کارشناسی و کارشناسی ارشد خود را در همین رشته -گرایش در سال‌های ۱۳۸۶ و ۱۳۸۸ از دانشگاه علم و صنعت ایران دریافت نموده است. رساله دکتری وی در رابطه با مدل‌سازی و ارزیابی کارایی‌پذیری در محیط گرید محاسباتی بوده و عمده فعالیت پژوهشی وی نیز در ارتباط با مفاهیمی همچون ارزیابی کارایی و اتکاپذیری، مدل‌سازی با روش‌های ریاضی، الگوریتم‌های زمانبندی و محاسبات گرید و ابر می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

entezari@ce.sharif.edu



علی موقر استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف می‌باشد. ایشان مدرک کارشناسی خود را در سال ۱۳۵۶ در رشته مهندسی برق از دانشگاه تهران، مدرک کارشناسی ارشد و دکتری خود را در سال‌های ۱۳۵۸ و ۱۳۶۴ در رشته مهندسی کامپیوتر، اطلاعات و کنترل از دانشگاه میشیگان دریافت کرده‌اند. زمینه تحقیقاتی وی شامل مدل‌سازی کارایی و اتکاپذیری، درستی‌یابی صوری شبکه‌های بی‌سیم و سیستم‌های بی‌درنگ توزیع شده می‌باشد. وی همچنین عضو ارشد IEEE و ACM می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

movaghar@sharif.edu

چارچوبی راهنما و پیشگیرانه، جهت برپائی ثمربخش و کم آسیب دوره‌های پسارشته‌ای کارشناسی ارشد علوم و مهندسی رایانش در ایران

سید ابراهیم ابطحی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

گذر به دوران پسارشته‌ای در آموزش‌های دانشگاهی علوم و مهندسی رایانش، نیازمند تحقق کمینه دانش و یادگیری مهارت‌های لازم، در دانش‌آموختگان است. در این مقاله چارچوبی برای تحقق این نیاز، پیشنهاد شده است. این چارچوب بر مبنای مدل حوزه‌ها و واحدهای دانشی *curricula* شکل گرفته و با افزودن بعد ارزش افزای زنجیره‌های درسی به آن، در قالب مدلی پنج بعدی برای تولید برنامه و طرح درس‌های دانشگاهی پیشنهاد شده است. سه بعد دیگر این چارچوب که زد - ۴ نامیده شده است شامل تناسب مضمونی (بر پایه ویژگی‌های شغلی)، ملاحظات راهبردی (تناسب با اسناد بالادستی شامل سند ملی اشتغال) و اقتضات میان‌رشته‌ای دروس است. تبیین دوران پسارشته‌ای در پی اعصار پیشارشته‌ای و رشته‌ای و تدقیق بیشینه‌ی گزینه‌های پسارشته‌ای، شامل چندرشته‌ای، میان‌رشته‌ای، بین‌رشته‌ای و فرارشته‌ای از دیگر موضوعات این مقاله است. استفاده از مفهوم همجوشی در شرایط تلفیق رشته‌ای با تکیه بر مطالعات موضوعی این حوزه، ویژگی چارچوب پیشنهادی مقاله است.

کلمات کلیدی: چارچوب‌های راهنمای برنامه درسی، دوره‌های میان‌رشته‌ای، پیوند ارزش افزای رشته‌ها، حوزه‌های دانشی رایانش، زنجیره‌های درسی مکمل، مهارت‌های حرفه‌ای فناوری اطلاعات.

۱- مقدمه

بنظر می‌رسد در شرایط گسترش دوره‌های پسارشته^۱، ماندن ما در عصر رشته‌ای^۲ و یا حتی برگشت به پیشا رشته‌ای^۳، گامی به پیش نیست و با نیازهای جامعه، سنخیت ندارد. انگاره برخی خبرگان با قیاس با دوران تحصیل خود در گذشته، یا با تکیه بر تجارب دانشگاه‌های پرسابقه و معتبر که با تغییرات عموماً محافظه کارانه برخورد می‌کنند، نباید ما را با تاخیری پر هزینه مواجه کند. هر چند حرکت در این جهت هم کم‌آسیب نیست.

چارچوبی راهنما می‌تواند این گذار را کم‌تهدید و پر فرصت کند. هر چند اقبال در مواردی ناشی از اضطرار دانشجویان ورودی تحصیلات تکمیلی، به دوره‌های میان‌رشته‌ای^۴ (گاه با تصور سادگی این دوره‌ها) را هم باید آسیب‌شناسی کرد. برای آغاز پر توان گسترش این دوره‌ها می‌توان از ابتدا با گلچین کردن دانشجویان توانا، سطح توقع و انتظار و کیفیت دانشجویان ورودی این رشته‌ها را افزایش داد بلکه فرآورده‌های این دوره‌ها نقش الگو را ایفا کنند.

چارچوب زد-۴، مدل پنج بعدی پیشنهادی این مقاله برای تولید برنامه و طرح درس‌های کارشناسی ارشد دوره‌های مهندسی و علوم رایانش^۱، در استمرار سیر تکامل روشنگان‌ها^۲ و مضامین مرتبط، پیشنهاد شده است. ابتدا بر نگره نوین مدل *curricula* یعنی گذر از حوزه‌ها و واحدهای موضوعی به حوزه‌ها و واحدهای دانشی، عامل یکپارچه ساز و تکوینی زنجیره‌های درسی افزوده شده است. سپس براساس ارزش ویژه الگوهای مدل - پایه تولید برنامه‌های درسی دانشگاهی، مدل چهار بعدی زد-۳ با افزودن ملاحظات میان‌رشته‌ای ارتقاء یافته و تبدیل به مدلی پنج بعدی برای چارچوب مورد نظر شده است.

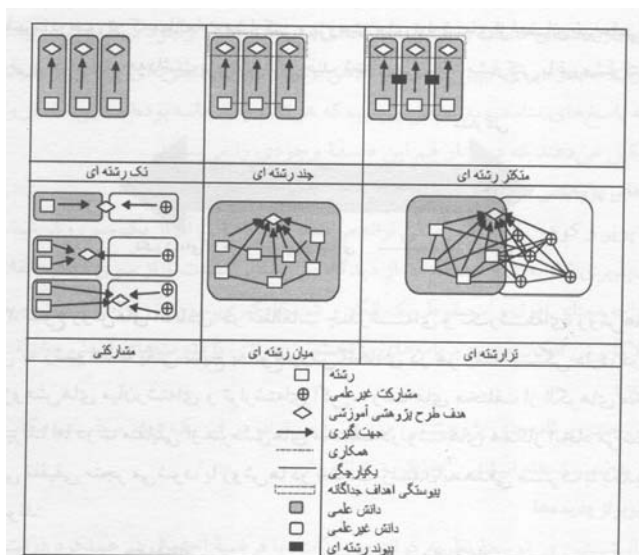
در ادامه این پژوهش با تدوین جزئیات مدل اجرایی و راهنمای تولید برنامه درسی، امکان تولید عناوین و محتوای دوره‌ها و درس‌ها برای مدرسانی با تخصص کمتر در حوزه تولید مدل - پایه دروس فراهم می‌گردد.

۲- پیشینه

رشته‌ای است، بخشی از هویت خود را وامدار این گفتمان است. میان‌رشته‌ای به معنی گردهمایی و تلفیق مصنوعی و تحمیلی دانش، نظریه‌ها، مفاهیم، تخصص‌ها و روش‌های دو یا چندرشته علمی و یا جمع‌آوری و التقاط داده‌های مختلف با یکدیگر و یا آمیزش چشم‌اندازهای رشته‌های مختلف به منظور دستیابی به یک تلفیق استعاره‌ای نیست. بلکه به معنی ایجاد ارتباط و تلفیق هدفمند و روشمند دانش، مفاهیم، مهارت‌ها و روش‌های رشته‌های مرتبط، پیرامون یک مسئله و یا موضوع پیچیده و واقعی است که مستلزم شناخت، فهم و تحلیل همه جانبه آنست. میان‌رشته‌ای مدعی آن است که با عرضه چشم‌اندازی ترکیبی از رشته‌های مختلف و تلاش برای ایجاد همگرایی و تعاون میان آنها، درکی جامع‌تر و پایدارتر را به ما ارائه می‌کند [۴].

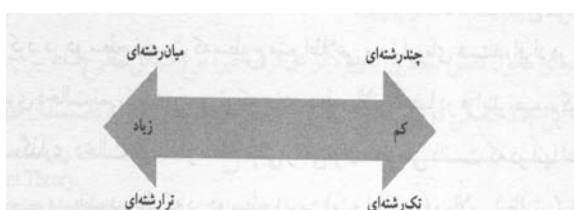
۳- مروری بر مطالعات پسرارشته‌ای

با توجه به تمایزات تقسیم‌بندی‌های پژوهش‌های میان‌رشته‌ای، یک تقسیم‌بندی هفت طبقه‌ای را برای این مطالعات می‌توان مناسب دانست: مطالعات تک رشته‌ای^۱، چندرشته‌ای^{۱۱}، متکثر رشته‌ای^{۱۲}، مشارکتی^{۱۳}، میان‌رشته‌ای^{۱۴}، ترا (فرا) رشته‌ای^{۱۵} و یکپارچه^{۱۶}. استفاده از واژه مطالعه برای تسری این طبقه‌بندی به حوزه آموزش استفاده می‌کنند که پژوهش در این زمینه‌ها پیش نیاز آن است که این تمایزات در شکل ۱ تصویر شده است [۵].



شکل ۱- تمایزهای میان‌رشته‌ای [۵]

تمایزهای مطالعات بین رشته‌ای را از چهار جنبه تولید علم یکپارچه، گونه مشارکت، گونه روش‌های اتخاذی و میزان مشارکت می‌توان بررسی کرد. در الگوی تولید علم یکپارچه، یکپارچگی رویکرد مشترک مطالعات است (شکل ۲). هدف مشترک، روش‌های همسو شده و نظریه‌ای واحد ولی چند جانبه ویژگی‌های این الگو است.



شکل ۲- الگوی تولید علم یکپارچه [۵]

جولی تامپسون کلاین (۲۰۰۴) می‌گوید: میان‌رشته‌ای نوعی فلسفه، روش شناسی، مفهوم، فرآیند، شیوه تفکر، طرز تفکر انعکاسی و یا ایدئولوژی واکنشی^۷ تعریف شده است. به منظور شناخت و فهم دقیق‌تر و علمی‌تر از میان‌رشته‌ای و ماهیت آن، ناگزیر از توصیف، تبیین و ارائه تعریف دقیق‌تر و جامع‌تری از رشته و ابعاد و خصلت‌های رشته^۸ هستیم. رشته‌ای، ساختار معرفتی حاکم بر سیاست‌ها، فرآیندها و شیوه‌های آموزشی و پژوهشی دو قرن اخیر است که به مثابه گفتمانی مسلط بر نهادهای سازمانی تولید دانش و عرضه آموزش نظیر دانشگاه حکمفرما گشته است. رشته‌های سنتی^۹، متشکل از ساختارهای منظم و سازمان یافته دانش هستند که به طور تاریخی و متعارف، حول مجموعه‌ای مشخص از مسائل و موضوعات مانند تاریخ، فلسفه، الهیات یا فیزیک متمرکزند [۱].

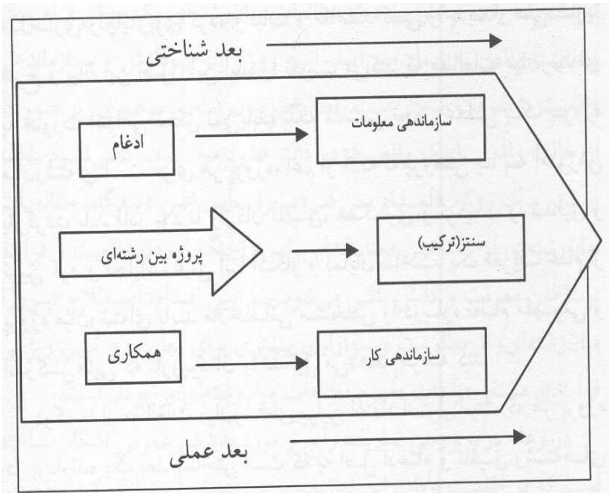
از عصر روشنگری تا اواسط قرن گذشته، گفتمان رشته‌ای و رویکردهای رشته‌ای، گفتمان غالب عصری بود که ارتباطات، تعاملات و تغییرات اجتماعی خطی و پیش‌بینی پذیر تصور می‌گردید. "لیوتار" می‌گوید: در شرایط پسادردن، مشروعیت زدائی از دانش نتیجه و محصول کارکرد گرائی بوده است. بر این اساس یک استاد شایسته‌تر از شبکه‌ای از بانک‌های حافظه فعال در زمینه انتقال دانش و نیز شایسته‌تر از گروه‌های بین رشته‌ای نیست. در اثر پاره پاره شدن دانش در جامعه پسا صنعتی و تبدیل آن اطلاعات، قدرت یکپارچه کننده آن از میان رفته است و به این ترتیب هیچ فضایی برای خودمختاری دانشگاه باقی نمانده است. "دریدا" می‌گوید تضاد میان تحقیقات بنیادی و کاربردی تخیلی بیش نیست، واقعیت تحقیقات چیز دیگری است. دیگر میان فناوری و علم تمایز و تفاوتی نمی‌توان قائل شد. امروزه، ما شاهد ظهور چیزی به نام فن- علم هستیم. این موید نظر "هایدگر" است که گفته است فناوری و اساسا ماهیت فناوری، صرفا فناورانه نیست [۲].

به گفته "گیبسون" الگوی جدیدی از دانش به نام الگوی دوم در حال جایگزینی الگوی اول (که مربوط به مدرنیته سازمان یافته است) می‌باشد. در الگوی دوم دانش با حجم انبوه تولیدکنندگان دانش مشخص می‌شود که بر اساس درخواست‌های ارائه شده به خود فعالیت می‌کنند. در این وضعیت، دانشگاه که با دسته‌ای از تولیدکنندگان جدید دانش مواجه است، جایگاه خود به منزله مکان اصلی و مهم تولید دانش را از دست می‌دهد. در شرایطی که چندرشته گرائی در حال تبدیل شدن به پدیده‌ای رایج و پدیده پسرارشته گرائی در حال مسلط شدن بر جهان دانش است، مرز میان‌رشته‌ها روز به روز تاریک و نامشخص‌تر می‌شود [۳]. مطالعات میان‌رشته‌ای، به سرعت در حال تبدیل شدن به نوعی هنجار و معیار برای بسیاری از دانشجویان است. این در حالی است که رشته‌های سنتی متقاضیان خود را روز به روز از دست می‌دهند و جای خود را به برنامه‌های جدید موضوع محور می‌دهند. موضوعات میان‌رشته‌ای همیشه به صورت خطی تحول نمی‌یابند، بلکه در حال تحول عمودی به سوی واحدهایی وسیع‌تر از قبیل قلمروها، گرایش‌ها و شاخه‌های تحصیلی نیز هستند که در اثر طبقه‌بندی دانش و شناخت، دیدگاه‌های آینده‌نگر، تحقیقات کاربردی، ساختارهای سازمانی و محافل علمی که در بردارنده تعاملاتی فراتر از یک رشته خاص علمی هستند، شکل می‌گیرند. بنابراین، میان‌رشته گرائی، منجر به توسعه رشته‌های جدید می‌شود. رشته‌های جدید غالبا بیش از رشته‌های پیشین اشتغال محور و متمرکز بر حرفه‌های شغلی هستند. ارتباط حرفه‌ای و پیوند شغلی نیروی محرکه عمده‌ای در جهت توسعه و گسترش برنامه‌ها و گروه‌های چندرشته‌ای و فرارشته‌ای محسوب می‌شود [۳].

میان‌رشته‌ای برای توصیف و تبیین سطح وسیعی از فعالیت‌های علمی، پژوهشی و آموزشی بکار برده می‌شود که کنشگران و متخصصان دو یا چندرشته علمی در ارتباط با شناخت، فهم و تحلیل یک پدیده، موضوع و یا مسئله پیچیده و واقعی با هم همکاری می‌کنند. گروهی معتقدند رشته‌ای بیش از حد فرسوده، کهنه و تک‌ساختی شده است و پاسخگوی تحلیل‌های چندوجهی نیست. اما گفتمان میان‌رشته‌ای که در ادامه گفتمان

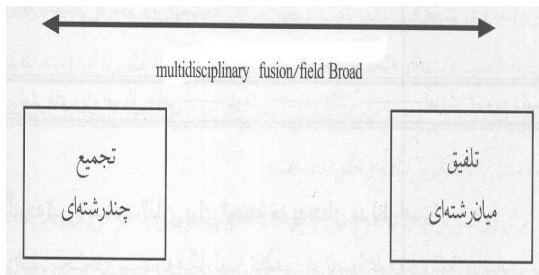
در تبیین بیشتر این گونه مدل‌های شبه فرآیندی می‌توان به روال ادغام و سازماندهی کار در قالب یک پروژه میان‌رشته‌ای پرداخت. وجه شناختی این پروژه شامل سازماندهی جدید معلومات و دانسته‌ها از طریق اتصال و تقابل چندین رشته علمی است.

سازماندهی دانش‌ها اشاره به تعامل شاخه‌های رشته‌ای دارد. این تعاملات و تبادلات بین مفاهیم، نظریه‌ها، روش‌ها و ابزارهای تحلیل صورت می‌گیرد. ایجاد شبکه مفهومی از اتصال مفاهیم، استفاده از تقارب و تجمیع راه‌حل‌های رشته‌ها و ایجاد یک چارچوب ترکیبی منسجم حالات متصور این ساختار مفهومی مطالعات میان‌رشته‌ای است (شکل ۶).



شکل ۶- ساختار مفهومی مطالعات میان‌رشته‌ای [۶]

نگاه طیفی به تاسیس رشته‌ها، از چندرشته‌ای تا میان‌رشته‌ای، می‌تواند از الگوهای تجمیع، تلفیق و همجوشی^{۱۷} بهره‌گیر (شکل ۷).



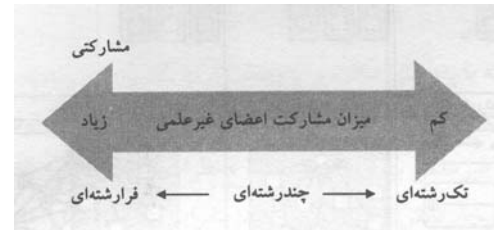
شکل ۷- میان‌رشته‌ای حاصل تجمیع، همجوشی و تلفیق رشته‌ها [۷]

تکرار این الگوها سطوح پسرشته‌ای را از طریق افزایش همکاری و هماهنگی بین رشته‌ها، در آموزش و نوآوری، بدست می‌دهد (شکل ۸).

کسب تخصصی ژرف در دانش‌آموختگان پسرشته‌ای هدفی ضروری است که "اشتاین" در قیفی تخصصی آنرا مصور کرده است (شکل ۹).

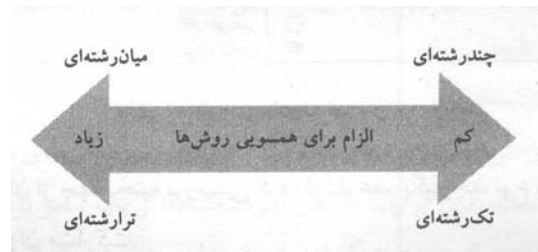
بر این مبنا با تکیه به انواع رویکردهای تلفیق (شکل ۱۰) می‌توان مجموعه‌ای از مفاهیم سیال دانش در کنش‌های علمی پرداخت (شکل ۱۱) و طبقه‌بندی میان بخشی سلسله مراتبی گونه‌های فعالیت‌های علمی (جدول ۱) را به انواع پسرشته‌ها نگاشت کرد.

در الگوی گونه مشارکت در مطالعات از اعضای هیات علمی و غیرهیات علمی اما پژوهشی بهره گرفته می‌شود (شکل ۳). هرچند در پژوهش‌های چندرشته‌ای می‌توان از الگوهای غیر مشارکتی هم استفاده کرد.



شکل ۳- الگوی گونه مشارکت [۵]

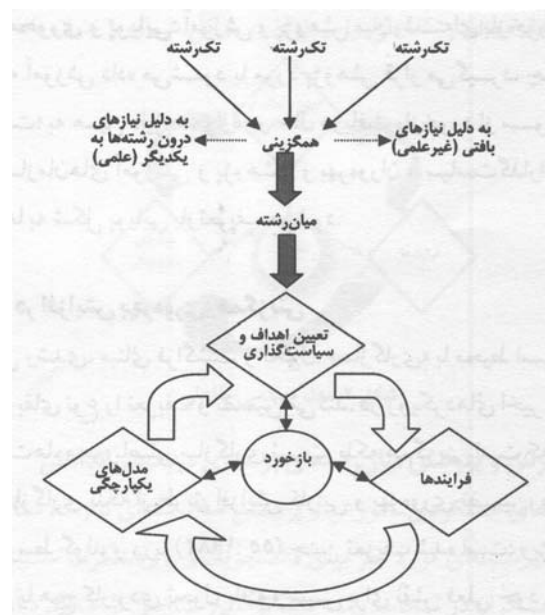
در الگوی روش‌های اتخاذی، درک متقابلی از بن‌انگاشت‌های متفاوت رشته‌های همکار ایجاد می‌شود و از روشی تلفیقی برای نیل به هدف مشترک استفاده می‌شود (شکل ۴).



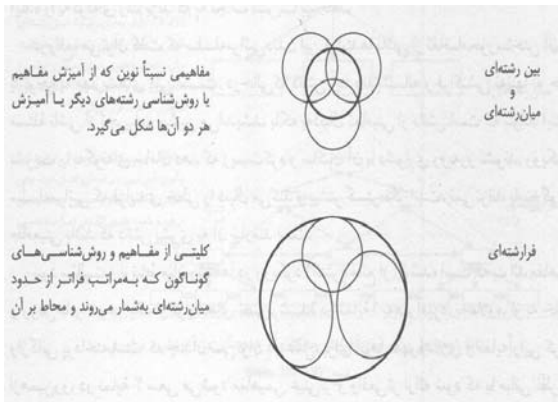
شکل ۴- الگوی روش‌های اتخاذی [۵]

در الگوی میزان مشارکت، اعضای غیرعلمی گروه، می‌توانند یکی از چهار نقش منبع اطلاعاتی، مشاوره، عامل یا ناظر را ایفا کنند.

از دیگر دستاوردهای این پژوهش‌ها به مدلی برای فراگشت میان‌رشته‌ای می‌توان اشاره کرد که بسیاری از الزامات برپائی دوره‌ها و دروس میان‌رشته‌ای را در بر دارد (شکل ۵).



شکل ۵- مدل فراگشت میان‌رشته‌ای [۵]



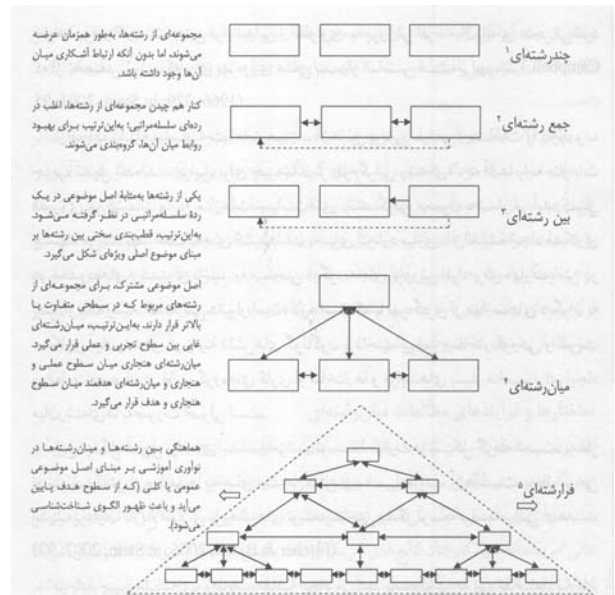
شکل ۱۱- مفاهیم سیال دانش در کنش‌های علمی [۸]

جدول ۱- طبقه‌بندی میانه بخشی سلسله مراتبی گونه‌های فعالیت علمی [۸]

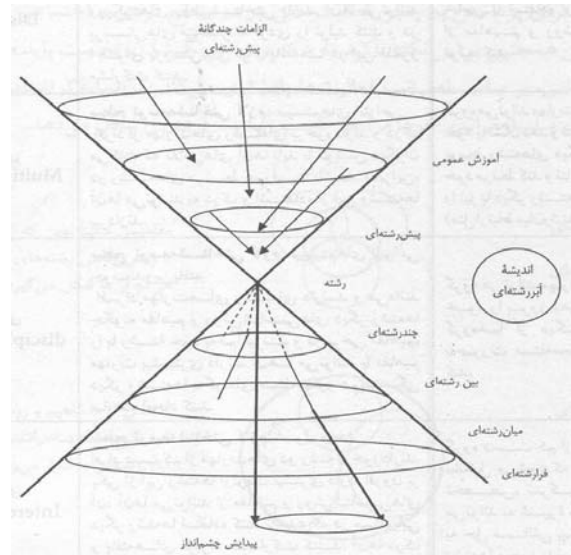
گونه‌شناسی	مهارت‌های فردی	مهارت‌های گروهی
رشته Disciplinary	سطح توسعه‌شناختی لازم: برنامه‌ریزی انتزاعی خوب تدوین یافته افراد درک درستی از مجموعه‌ای از مفاهیم و رویکردهای روش‌شناختی دارند. آن‌ها می‌توانند پرسش‌های منحصر به فردی را تولید کنند و در اجرای پژوهش‌های نو و یافته‌ها در این قلمرو مشارکت کنند.	گروه می‌تواند دانش نو (با تأیید دانش موجود) را در رشته‌ای خاص با استفاده از مجموعه‌ای از مفاهیم و روش‌شناسی‌ها تولید کند.
چندرشته‌ای Multidisciplinary	سطح توسعه‌شناختی لازم: سیستم‌های انتزاعی افراد از مهارت‌های رشته‌ای برخوردارند و درک می‌کنند که تلاش‌های آن‌ها باید با کوشش دیگران در رشته‌ای مرتبط هم‌راستا باشد. بنابراین، آن‌ها می‌توانند به درک و استفاده از این رشته‌ها بپردازند.	گروه می‌تواند مهارت‌های رشته‌ای خود را نشان دهد و نتایج تولید شده توسط رشته‌های دیگر را با رشته خود مرتبط کند و نتایج رشته خود را نیز با دیگر رشته‌ها پیوند دهد (مثل ارتباط میان‌رشته‌ها)
بین‌رشته‌ای Disciplinary Cross	سطح توسعه‌شناختی لازم: سیستم‌های انتزاعی خوب تدوین یافته افراد مهارت‌های رشته‌ای دارند و می‌دانند چگونه مفاهیم و روش‌شناسی‌های دیگر رشته‌ها را با رشته خود به هم آمیزند و در برخی مفاهیم، مهارت بیشتری دارند. آن‌ها می‌توانند با مفاهیم دیگر رشته‌ها به گونه‌ای مسئله‌محور هم‌پیوستگی سازنده ایجاد کنند.	گروه می‌تواند مهارت رشته‌ای خود را بروز دهد و با دیگر گروه‌ها از دیگر رشته‌ها به‌صورت مسئله محور همکاری کند.
میان‌رشته‌ای Interdisciplinary	سطح توسعه‌شناختی لازم: اصول واحد افراد دست‌کم از مهارت‌های دو رشته برخوردارند. یکی از این رشته‌ها اولویت بیشتری دارد. افزون بر آن، آن‌ها می‌توانند از مفاهیم و روش‌شناسی‌های دیگر رشته‌ها استفاده کنند تا حدی که در مسئله‌یابی و یافته‌های مربوط مشارکت کنند. آن‌ها درک جدیدی از رشته نخستین خود دارند.	گروه دست‌کم از دو زیر گروه تشکیل می‌شود که بر روی یک تخصص، تمرکز دارد. گروه می‌تواند به شیوه مسئله محور به حل مسائلی بپردازد که در رشته‌ها وجود ندارند.
فرارشته‌ای Transdisciplinary	سطح توسعه‌شناختی لازم: فراتر از اصول واحد افراد دست‌کم از مهارت در دو رشته برخوردارند. هیچ‌کدام از آن‌ها الویت خاصی ندارند. آن‌ها در هر دو قلمرو کار می‌کنند و در تولید یافته‌ها، مفاهیم و موضوعات به مثابه ظهور چشم‌انداز فرارشته‌ای مشارکت می‌کنند. آن‌ها می‌توانند با مفاهیم و روش‌های مجموعه‌ای از رشته‌ها به شکل اجرایی ارتباط برقرار کنند.	گروه دست‌کم از دو زیر گروه تشکیل می‌شود که هیچ‌کدام اولویت خاصی نسبت به دیگری ندارد و به تولید دانش مسئله‌محور و اجرایی می‌پردازد که نمی‌توان آن را تا حد یک رشته فروکاست. گروه می‌تواند سرمنشأ رشته‌های جدید و اصلاح رشته‌های موجود بر مبنای چشم‌انداز جدید باشد.

۴- مفروضات گذر از پیشا به پسرانته‌گی

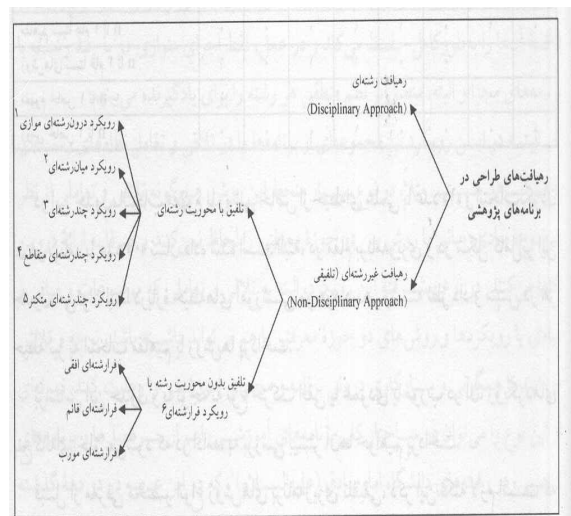
سیر تکوینی گفتمان رشتگی که از پیشارشته‌گی‌آغاز شد، پس از گذر از رشتگی، تا پسرانته‌گی امتداد می‌یابد. در دوران پیشا رشتگی، قوام موضوعات علمی به سختی



شکل ۸- مراحل پیاپی برای افزایش همکاری و هماهنگی در آموزش و نوآوری [۸]



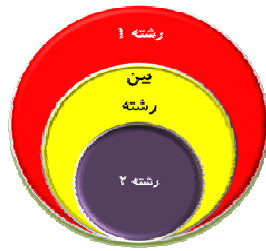
شکل ۹- کیف تخصصی [۸]



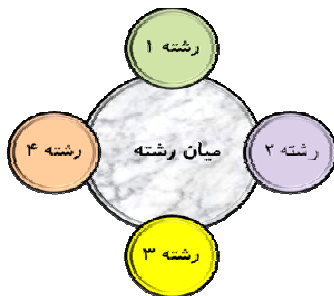
شکل ۱۰- انواع رویکردهای تلفیق [۸]



شکل ۱۶- پارسرشتگی - چندرشته‌ای



شکل ۱۷- پارسرشتگی - بین‌رشته‌ای



شکل ۱۸- پارسرشتگی - میان‌رشته‌ای

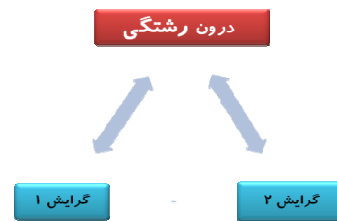


شکل ۱۹- پارسرشتگی - فرارشته

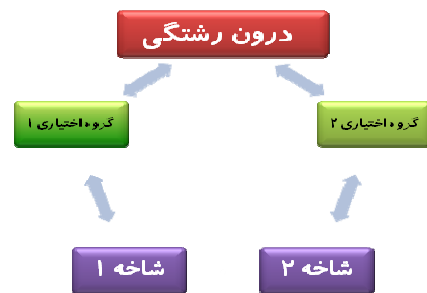
به حوزه‌های مستقل و یا کاملاً منسجم و سترگ، قابل تقسیم بود. در دوران رشتگی این انسجام حاصل شد و در دوران پارسرشتگی، نیاز به تحلیل‌های چندوجهی رشته‌ها را به همکاری فراخواند. دوران رشتگی، عصر رشته‌های مستقل در دانشکده‌هایی با اسامی رشته‌ها و گروه‌های درسی متولی رشته‌ها است. در عصر رشتگی، پس از مدتی ذیل گزینه درون رشتگی^{۱۸}، دوره‌های تحصیلی در قالب‌های گزایشی^{۱۹}، شاخه‌ای^{۲۰} و تمرکزی^{۲۱}، طراحی و بکار گرفته شدند (اشکال ۱۲ تا ۱۵). اما زمانه پارسرشته با ضرورت چند زمینه‌ای^{۲۲} و فرازمینه‌ای^{۲۳} شمردن مضامین و فرآیندهای آموزشی شکل گرفت. تکثیر رشتگی^{۲۴}، زنجیره‌ای از حوزه‌های چندرشتگی^{۲۵}، میان‌رشتگی^{۲۶}، بین‌رشتگی^{۲۷} تا فرارشتگی را در پی داشت (اشکال ۱۶ تا ۱۹) که پسا نوگرایان^{۲۸}، پارسرشتگی را گاه تا ضد رشتگی^{۲۹} و پاد رشتگی^{۳۰} امتداد دادند [۱۰].



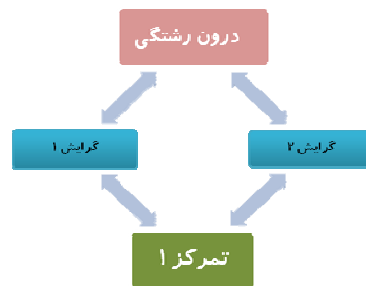
شکل ۱۲- از پیشا تا پارسرشته



شکل ۱۳- درون‌رشتگی گزایشی



شکل ۱۴- درون‌رشتگی شاخه‌ای



شکل ۱۵- درون‌رشتگی تمرکزی

۵- پیشینه، و نقاط قوت و ضعف اقدامات کنونی

پیشینه دوره‌های آموزش دانشگاهی علوم و مهندسی رایانش^{۳۱} (شامل مهندسی و علوم رایانه و فناوری اطلاعات) در ایران با عمری کمتر از نیم قرن (۱۳۹۱-۱۳۴۷) نشان از چهار دوره متفاوت دارد که از آنها با عناوین دوران‌های گرده برداری عالمانه (۵۱-۴۷)، نوآوری‌های بی‌سرانجام (۵۶-۵۲)، مناسب ساری محدود روزآمد (۹۰-۷۰)، پالایش و تحول خواهی کم فرصت (۹۰- تاکنون) می‌توان یاد کرد [۱۱]. اما در شرایط امروز که به گسترش بکارگیری فناوری اطلاعات و افزایش کثیر تعداد شاغلین به تحصیل در دوره‌های دانشگاهی شده، نیاز به این تحول احساس و اقدامات جسته و گریخته‌ای برای انجام آن شده است. در این میان دو اقدام مهم که نیاز به آسیب‌شناسی دارد، عبارت از شکل‌گیری گروه‌های کارشناسی محدود و موظف از خبرگان دانشگاهی برای بازبینی برنامه دوره‌ها، و تعیین جایگاه فناوری

احتراز از تک درسی شدن این دوره‌ها، هم برای دروس جبرانی و هم برای سایر دروس، باید لحاظ شود. واحد مجاز تعیین کمینه سواد لازم هر رشته و زنجیره دروس پیشنهادی مربوطه، باید از بین مجرب‌ترین و معتبرترین دانشگاه‌های عرضه کننده رشته‌های پیوندی، انتخاب شود. محل ارائه و متولی اولویت‌دار دروس (با غلبه محتوایی دانش و مهارت یک رشته در آن)، بهتر است از بین دانشکده‌های متولی آن رشته انتخاب شود.

این چارچوب، از یک مدل پایه، برای تولید برنامه‌های درسی دانشگاهی پسرارشته‌ای و توصیه ده رفتار اثربخش، برای پنج ذینفع اصلی (درس، دانشجو، استاد، دانشکده مجری، نحوه اجرا) تشکیل شده است. مدل پایه‌ی تولید محتوای دروس (قابل ارتقاء جهت تولید زنجیره‌های درسی، عناوین دوره‌ها و برنامه دوره‌های آموزشی) میان‌رشته‌ای پیشنهادی، که مدلی مناسب سازی شده است را زد-۴ می‌نامیم. این مدل پنج بعدی و مناسب شده‌ی مدل چهار بعدی زد-۳ است، که به کمک آن دروس جدید و اضافه شده به برنامه درسی دوره کارشناسی فاه، در دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، طراحی و تولید شده است [۱۸]. مدل زد-۳ بر پایه مدل حوزه‌ها واحدهای دانشی curricula طراحی شده است، که به آن سه بعد: زنجیره‌های درسی، ویژگی‌های مهارتی مورد نیاز دانشجو و الزامات اسناد بالادستی - شامل ظرفیت‌های آتی اشتغال جهت تعیین میزان پذیرش دانشجو - افزوده شده است [۱۹] و [۲۰]. در زد-۴ جهت دوره‌های میان‌رشته‌ای، بعد الزامات پسرارشته‌ای را به آن می‌افزاییم. الزامات پسرارشته‌ای در مدل زد-۴ با محوریت رعایت ویژگی‌های گونه‌ی پسرارشته‌ی منتخب، الزاماتی از جمله: الگو و فرآیند تعیین و تحقق سواد کمینه‌ی لازم رشته‌های پیوندی به عنوان پیش نیاز ورود دانشجو به این دوره‌ها، در قالب دروس جبرانی، تعیین می‌شود (شکل ۲۰) و این مدل قابل تعمیم به دوره‌های پسرارشته‌ای کارشناسی ارشد دیگر حوزه‌های مهندسی است (شکل ۲۱).



شکل ۲۰- اجزای مدل زد - ۴: چارچوبی برای تولید محتوای دروس دوره‌های پسرارشته‌ای کارشناسی ارشد فناوری اطلاعات



شکل ۲۱- تعمیم مدل زد - ۴ به دوره‌های پسرارشته‌ای کارشناسی ارشد حوزه‌های مهندسی

اطلاعات در گروه فناوری‌های نوین در وزارت علوم، تحقیقات و فناوری و اقدام برای تدوین برنامه‌های کارشناسی ارشد فناوری اطلاعات عمدتاً در قالب بین‌رشته‌ای است.

علیرغم اینکه فناوری اطلاعات از جمله شش فناوری برتر در کشور است که در سیاست های کلی سند چشم‌انداز و برنامه چهارم توسعه، بر توسعه آنها تاکید شده است اما در نقشه علمی کشور بهای لازم به این فناوری داده نشده است [۱۲] در این میان دیوانسالاری گسترده و بدون حافظه سازمانی جهت تکرار توفیقات و یادگیری از شکست ها، دشواری‌های بسیاری در این حوزه ببار آورده است که بسیاری از آنها ثمره اقدامات اداری نسنجیده و پر خطر است.

اقدام درج فناوری اطلاعات در گروه فناوری‌های نوین، با اقدام کمیته برنامه‌ریزی درسی مربوطه، از طریق اعلام و پذیرش پیوند شالوده‌ای فناوری اطلاعات با مهندسی رایانه در دوره کارشناسی و برنامه‌ریزی میان‌رشته‌ای برای دوره‌های کارشناسی ارشد فناوری اطلاعات در قالب فناوری‌های نوین، کم آسیب شد. اما آسیب شکل اجرای بازبینی دوره‌ها و الگوی انجام آن که ایرادات جدی دارد، به قوت خود باقی ماند. اولاً شیوه نامه تدوین و بازنگری برنامه درسی موجود در این گروه‌ها عمل نمی‌شود. زیرا این شیوه‌نامه غیرفرآیندی، توصیفی و عملاً مناسب دوره‌های آموزش پیش دانشگاهی است [۱۳]. ثانیاً روال‌های نظرسنجی محدود در تنگنای زمانی و بدون کار پژوهشی پشتیبان، در جلسات با الگوی تصمیم‌گیری رای گیرانه، که تناسبی با اجماع علمی، مورد نیاز ندارد، ایرادی پایه‌ای است که امروزه حتی در اجرائی تکثر گرا، تنها منجر به بازگشت به دیدگاه‌های درون رشتگی^{۳۲} متعلق به دوران ابتدائی رشتگی، در قالب گرایش‌ها، شاخه‌ها و تمرکزها گردیده است. بخش اقدامات تصویب، بازبینی دوره‌های بین رشته‌ای اینک به اضطرار با الگویی نامدون، به شکل موردی عمل می‌شود.

دیگر آسیب‌هایی که پیش از ارائه راه حل، می‌توان به فهرست آنها اشاره کرد عبارتند از: خطر تربیت بین رشته‌ای هائی با دانش کم، جابجائی مدرسین محدود موجود از حوزه‌های رشته‌ای به بین رشته‌ای با دلایلی به جز ضرورت انتقال خیره‌ترین اساتید رشته‌ها به دوره‌های بین رشته‌ای، تبدیل آموزش دانشگاهی به جمعیتی از تک درس‌ها، به جای زنجیره‌های درسی ارزش افزا و شکل‌گیری آغازین و در شروع بین رشته‌ای ها در واحدهای آموزشی مستقل، به جای اجرا در مکان مناسب مثلا دانشگاه‌های ما در رشته‌های پیوندی.

۶- ابعاد و اجزای چارچوب راه‌حل

تدوین برنامه‌های درسی دانشگاهی، نیازمند چارچوبی تحلیلی و گسترش‌پذیر است که متفاوت با مدل‌های پیش دانشگاهی، از تعیین اهداف رفتاری آغاز نمی‌شود، بلکه به علت بلوغ دانشجو، با تعیین نیازهای دانشی آغاز می‌شود. یکی از مناسب‌ترین گزینه‌های پیشنهادی برای استخراج حوزه‌ها و واحدهای دانشی، مدل curricula و توصیه‌های SIGITE کمیته مشترک ACM & IEEE (با سابقه‌ای بیش از بیست سال و بهره‌مند از بهترین تجارب مهم‌ترین دانشگاه‌های جهان)، برای تولید برنامه‌های درسی آموزش دانشگاهی رایانش است [۱۴]، [۱۵]، [۱۶] و [۱۷]. این مدل، چون در قالب چارچوب عرضه شده، قابل مناسب‌سازی موردی است و ابزاری مفید برای تولید برنامه‌های درسی دانشگاهی تلقی می‌شود.

بحث دوم، ضرورت ایجاد و حفظ محرک‌های انگیزشی در ارائه دوره‌های میان‌رشته‌ای از طریق فراهم آوردن فضای برای پذیرش و جذب نخبگان سایر رشته‌هاست. بنابراین برای این دوره‌ها هم از دانش‌آموختگان هر یک از رشته‌های پیوندی و هم از متقاضیان دانش‌آموخته سایر رشته‌ها، باید دانشجو پذیرش کرد. برای جلوگیری از ورود دانش‌آموختگان کم دانش به عنوان ورودی این دوره‌ها، باید احراز و جبران شرایط منجر به تحقق کمینه سواد لازم از رشته‌های پیوندی - حتی به قیمت طولانی‌تر کردن مدت دوره آموزشی - در اولویت قرار گیرد. طراحی زنجیره‌های درسی و

می‌یابد (جدول ۳) و با استفاده از چارچوب‌های مفهومی تدریس دروس میان‌رشته‌ای (شکل ۲۲) به شکل موثر به اجرا گذاشته می‌شود.

جدول ۲- برخی عناوین درسی رشته فرضی کارشناسی ارشد اقتصاد فناوری اطلاعات و ارتباطات [۸]

خصوصی‌سازی و مقررات در فناوری اطلاعات و ارتباطات
مدیریت و سازمان در فناوری اطلاعات و ارتباطات
سیاست‌های تجاری و بازارهای بین‌المللی فناوری اطلاعات و ارتباطات
مدیریت پروژه‌های فناوری اطلاعات و ارتباطات
حسابداری صنعتی در فناوری اطلاعات و ارتباطات
مسائل بانکی و بیمه در فناوری اطلاعات و ارتباطات
بازاریابی و تولید در فناوری اطلاعات و ارتباطات
مهندسی فناوری اطلاعات و ارتباطات
تأمین مالی و سرمایه‌گذاری خارجی در پروژه‌های فناوری اطلاعات و ارتباطات
نقش فناوری اطلاعات و ارتباطات در توسعه اقتصادی
اقتصاد فناوری اطلاعات و ارتباطات ایران
حقوق اطلاعات و ارتباطات

جدول ۳- سلسله مراتب به هم تنیدگی در برنامه درسی میان‌رشته‌ای [۹]

۱۱. فرارشته‌ای
۱۰. میان‌رشته‌ای
۹. چندرشته‌ای
۸. تکمیلی
۷. همبستگی
۶. اشتراک
۵. همبستگی
۴. آشیانه‌سازی
۲. آگاهی
۱. جدایی

دروس در این دوره‌ها لازمست به شکل زنجیره‌های دو یا سه درسی، طراحی و ارائه شود تا با تک درسی شدن دوره، ماهیت دانشگاهی آن، مخدوش نشود [۲۱]. کمینه‌ی سواد لازم، شامل یک زنجیره سه درسی از هر رشته پیوندی، باید تعیین و هر دانشجوی ورودی، ملزم به گذراندن آزمون تایید داناتی یا گذراندن این دروس، به عنوان درس جبرانی قبل از شروع دوره، شود. دانش‌آموخته هر رشته، از گذراندن زنجیره سواد رشته دانش‌آموخته خود، معاف است. دانشجوی خارج از رشته‌های پیوندی (اما همگن با آنها)، تنها پس از گذراندن همه زنجیره‌های درسی سواد رشته‌های پیوندی، می‌تواند وارد دوره پسرارشته‌ای شوند. برای این گروه، افزایش تعداد واحدها از سقف واحدهای دوره کارشناسی ارشد را، می‌توان با سه ساله کردن دوره (مشابه دوره های DESS فرانسه - دوره سه ساله بین کارشناسی ارشد و دکتری - با نمونه دوره بین رشته‌ای تله ماتیک در دانشگاه پاریس ۶)، جبران نمود.

دانشجویان این دوره‌ها را، از همه رشته‌های پیوندی و دیگر رشته‌های همگن، باید بتوان پذیرفت. این الزام، برای شکل‌گیری و گسترش کیفی این دوره‌ها ضروری است. در سه دوره اول پذیرش باید تواناترین دانشجویان را - هر چند به تعداد کم - برگزید و این دوره‌ها به شکل فقط آموزشی یعنی بدون پایان‌نامه نباید عرضه شود.

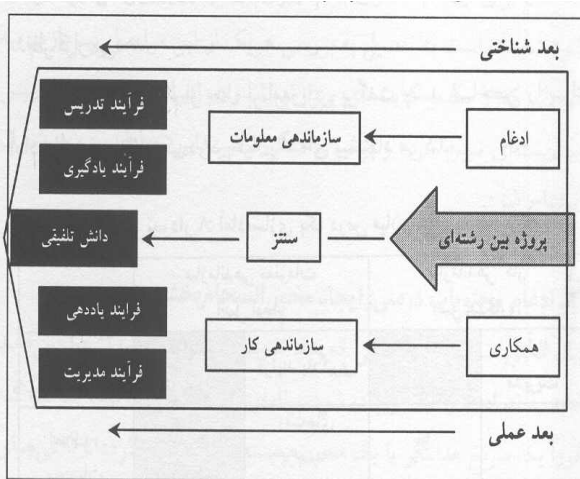
دانشگاه‌های مرتبط با این دوره‌ها، لازمست تعیین دروس مولد کمینه‌ی سواد رشته‌ای که در آن خیره هستند، به عهده بگیرند و متولی ارائه دروس رشته خود، شوند. در اجرا، برای ارتقاء سطح آموزش‌ها و تمام وقت بودن واقعی دانشجویان، مناسب است در سه دوره اول اجرا، از دانشجویان بورسیه - بدون دغدغه‌های معاش - استفاده شود. برای اجرا هم، جهت ارائه واجد کیفیت دوره‌ها، از بین دانشکده‌های مجری رشته‌های پیوندی، باید انتخاب محل مناسب تر با امکانات بیشتر، در اولویت قرار گیرد.

اساتید همکار در آموزش و پژوهش این دوره‌ها، لازمست از مجرب‌ترین و خبره‌ترین اساتید رشته‌ها، انتخاب شوند و به آنها امکانات لازم برای ترجمه و تالیف کتب دانشگاهی میان‌رشته‌ای، داده شود.

۷- مطالعه موردی کارشناسی ارشد رایانش

پسرارشته‌ای‌های متعددی در حوزه رایانش و دامنه جزئی فناوری اطلاعات، می‌توان تعریف کرد که ثمره پیوند تجارت، حقوق، هنر یا پزشکی و بسیاری رشته‌های دیگر است. امروزه حتی فرارشته‌ای با عنوان علوم و مهندسی شبکه اعم از شبکه‌های رایانه‌ای، اجتماعی، عصبی، مصوب و در دانشگاه تهران در دست راه‌اندازی و اجراء است. البته این اولین فرارشته‌ای عرضه شده در ایران نیست. چهار سال است که دانشگاه تهران، فرارشته علوم مهندسی را، ارائه می‌کند و هر چند تجربه دانشگاهی مثل تورنتو در کانادا، در ارائه این دوره، از دانشگاه تهران، حدود سی سال بیشتر است. اما بین رشته‌ای هائی هم در کشور ما، در دست اجراء است که حوزه رشته‌ای واحدهای مجری آموزش آنها، سنخیت کمی با حوزه گسترده و در مواردی فرارشته‌ای این دوره‌ها، دارد. که این امر می‌تواند به قلت کیفی این دوره‌ها، منجر شود. مثلا دوره‌های کارشناسی ارشد سنجش از راه دور یا سامانه‌های اطلاعات جغرافیائی در مواردی، با این دشواری مواجهند. در عین حال به علت هم وزنی، هم‌ارزی و عمق مطالب مورد نیاز، در هر دو رشته پیوندی، دوره‌های نظیر آموزش الکترونیکی، بانکداری الکترونیکی یا تجارت الکترونیکی، در قالب میان‌رشته یا فرارشته با کیفیت غنی و به شکل مستقل، هنوز طراحی و راه‌اندازی نشده‌اند. که راه کارهائی را هم در این موارد، باید اندیشید [۲۲] و [۲۳].

از این مطالعات و فرآیندهای پژوهشی دوره‌های پسرارشته‌ای غنی و موثر حاصل می‌شود (جدول ۲) که این درهم تنیدگی تا برنامه‌های درسی میان‌رشته‌ای امتداد



شکل ۲۲- چارچوب مفهومی تدریس میان‌رشته‌ای [۶]

۸- فضای توسعه پایدار مولد عصر پسرارشته‌ای

رعایت ملاحظات و کمک به توسعه پایدار^{۳۳}، از الزامات راهبردی^{۳۴} فعالیت در دنیای مدرن است تا در مواجهه با فناوری‌ها - نه با داوری ارزشی بلکه با پذیرش واقعیت غیر خنثی و اثر گذار آنها - به فرصت‌سازی و کاهش تهدیدات آنها پرداخت. عصر آموزش‌های پسرارشته‌ای، می‌تواند مولد بخشی از الزامات فضای

هم تکرار نمی‌کنیم. جای سمپوزیومی سالیانه در ایران با موضوع آموزش رایانش، کماکان خالی است. گسترش کمی بی‌رویه دوره‌های تحصیلات تکمیلی، کمتر فرصتی برای نوآوری‌های آموزشی و تکثیر آن به مدرسین دانشگاهی می‌دهد. توصیه این مقاله، به استفاده از چارچوب، در گسترش دوره‌های پسرارشته‌ای رایانش، می‌تواند با تحقیقات کمی و کیفی بیشتر، منجر به شناسائی حوزه‌های نیاز عاجل تر و موسسات بالقوه توانا تر، در اجرای این آموزش‌ها شود. این دستاورد احتمالی، برای آینده آموزش‌های دانشگاهی حوزه رایانش کلیدی است.

توسعه پایدار باشد. چنانچه در محتوای آموزشی به مفاهیم آدابی، فناوری‌های سبز، الزامات حرفه‌ای گری، کمک به کاهش فاصله رقمی و چاره اندیشی برای بازیافت ابزار مستهلک فناورانه، بپردازد. پسرارشته‌ها، در شکل‌گیری فضای همکاری (به جای فضای رقابت نا سالم) بین واحدهای متولی آموزش‌های دانشگاهی هم موثرند.

۹- هستان‌شناسی عنوان دوره‌های پسرارشته‌ای

برای انتخاب عناوین دوره‌های پسرارشته، با دارا بودن ویژگی‌های بیانگر کمیت و کیفیت محتوای دوره‌ها و پیشگیری از عناوین متفاوت برای محتواهای یکسان و بالعکس، دشواری‌هایی وجود دارد. اما اگر بتوان برای هر گونه پسرارشته، توالی از اسامی انتخاب کرد که نحوه پیوند چندرشته و محتوای آن را به گونه‌ای یکه، بازنمائی کند، بسیار مناسب است. استفاده از الگوهای هستان‌شناسی موردی نیز می‌تواند مفید باشد. اما دشواری‌های زبانی، مسئله‌ساز است. مثلا نامگذاری دو بین رشته‌ای حقوق انفورماتیکی و انفورماتیک حقوقی با الگوی هستان شناسانه به گونه‌ای که تفاوت‌های فاحش محتوایی این دو میان‌رشته را باز نمائی کنند، نیاز به پژوهش دارد [۲۴].

۱۰- نتیجه‌گیری

علیرغم دشواری‌های موجود، نیازهای روزافزون، ایجاب می‌کند، دوره‌های پسرارشته‌ای که با کیفیت راه‌اندازی شده‌اند، گسترش یابند. پسرارشته‌های جدیدی که متقاضی توانا دارند، طراحی شده و به اجرا درآیند تا اندکی از عقب‌ماندگی‌های ما در این زمینه‌ها، جبران شود. لازم به یادآوری است که در نیمه دوم دهه ۹۰ میلادی که دانشگاه موناخ استرالیا، دوره‌های متعدد بین رشته‌ای را راه‌اندازی کرد، دانشگاه‌های ما - علیرغم اطلاع و هشدار دلسوزان - به لزوم فعالیت در این زمینه توجه نکردند. وضع کنونی موناخ، توفیقات و گستره فعالیت‌های آن، نشان می‌دهد که ما به یک فرصت آموزشی حتی با منافع مالی، در زمان خود، کم توجه بوده‌ایم. اما راه گشوده شده بر گسترش پسرارشته‌ای‌های رایانش را اینک باید کم خطر کرد، زیرا پیشینه‌ی گونه مواجهه ما با پدیده‌های نو - که در ابتدا دفعی و عاقبت بر اثر نیاز، ایجابی و پس از ایجاب گاه، ولع آلود است - این امر را لازم می‌سازد. اساتید خبیره رشته‌های دانشگاهی می‌توانند با اساتید جوان، نوآور و پرتوان، هسته‌های اصلی راه‌اندازی و گسترش یر ثمر و فرصت، پسرارشته‌های رایانشی را تشکیل دهند. شرط کافی در این میان حساسیت نسبت به کیفیت آموزش هاست، در ابتدای برگزاری این دوره‌ها، با هدف انتقال دانش و تجربه فنی، از دانشگاه‌های مجرب خارجی به عنوان همکار، میتوان کمک گرفت. تجارب موفق دانشگاه‌های داخلی از جمله همکاری مشترک دانشگاه تربیت مدرس با دانشگاه لولای سوئد، در برگزاری مشترک و موفق بین رشته‌ای تجارت و بازاریابی الکترونیکی، می‌تواند راهنما باشد. ایرانیان پر شمار دانشگاهی موفق خارج از کشور را در دوره‌های کوتاه مدت، می‌توان در برگزاری این دوره‌ها شریک کرد و از تجارب مفید آنها بهره گرفت [۲۵]، [۲۶] و [۲۷].

۱۱- پژوهش‌های آینده

پژوهش انجام شده که این مقاله حاصل آنست، نشان می‌دهد آموزش دانشگاهی رایانش در ایران با قلت پژوهش‌های موردی و آینده نگارانه مواجه است [۲۸]. البته در غیاب حافظه بخشی برای این فعالیت‌ها، آسیب‌شناسی کمی، دشوار و تحلیل کیفی، تکراری است زیرا از تجارب ناموفق آموخته‌هایی نمی‌اندوریم و به همین ترتیب تجارب موفق را

مراجع

[1] J. T. Klein, *Creating Interdisciplinary Campus Cultures: A Model for Strength and Sustainability*, Jossey-Bass Publishing, 2010.

[۲] ج. دلانتی، دانش در چالش: دانشگاه در جامعه دانائی، ترجمه ع. بختیاری‌زاده، پژوهشکده مطالعات فرهنگی و اجتماعی، وزارت علوم، تحقیقات و فناوری، ۱۳۸۹.

[۳] ب. رامانیا، و م. موهان، یادگیری و تدریس میان‌رشته‌ای در آموزش عالی: نظریه و عمل، ترجمه م. ر. دهسیری، پژوهشکده مطالعات فرهنگی و اجتماعی، وزارت علوم، تحقیقات و فناوری، ۱۳۸۹.

[۴] س. م. علوی‌پور، و همکاران، "چالش‌ها و چشم‌اندازهای مطالعات میان‌رشته‌ای (مجموعه مقالات)،" پژوهشکده مطالعات فرهنگی و اجتماعی، وزارت علوم، تحقیقات و فناوری، ۱۳۸۷.

[۵] ک. زاهدی، "بررسی فراگشت آموزش و پژوهش میان‌رشته‌ای: نیاز دیروز، رویکرد امروز، زیر ساخت فردا،" فصلنامه علمی - پژوهشی مطالعات میان‌رشته‌ای، شماره چهارم، ۱۳۸۸.

[۶] ن. ر. کوله، آموزش دانشگاهی و مطالعات میان‌رشته‌ای، ترجمه م. ر. دهسیری، پژوهشکده مطالعات فرهنگی و اجتماعی، وزارت علوم، تحقیقات و فناوری، ۱۳۸۸.

[۷] م. مهر محمدی، "ملاحظات اساسی در باب سیاست‌گذاری توسعه برنامه درسی میان‌رشته‌ای در آموزش عالی از منظر فرآیند تکوین،" فصلنامه علمی - پژوهشی مطالعات میان‌رشته‌ای، شماره سوم، ۱۳۸۸.

[۸] ع. پیغمایی، و ج. نورانی، "گونه‌شناسی رهیافت‌های تلفیقی در طراحی برنامه‌های درسی و دلالت‌های کاربردی آن برای رشته اقتصاد،" فصلنامه علمی - پژوهشی مطالعات میان‌رشته‌ای، دور دوم، شماره یک، ۱۳۸۸.

[۹] ع. بازرگان، "چگونگی ارزشیابی برنامه‌های درسی میان‌رشته‌ای،" فصلنامه علمی - پژوهشی مطالعات میان‌رشته‌ای، شماره سوم، ۱۳۸۸.

[۱۰] ع. خورسندی طاسکوه، "گفتمان میان‌رشته‌ای دانش،" انتشارات پژوهشکده مطالعات فرهنگی و اجتماعی، وزارت علوم، تحقیقات و فناوری، ۱۳۸۷.

[۱۱] س. ا. ابطحي، "آموزش مهندسی فناوری اطلاعات تا سال ۱۴۰۴،" کنفرانس آموزش مهندسی در ۱۴۰۴، دانشگاه تهران، ۱۳۸۸.

[۱۲] م. ع. کی نژاد، و غ. ر. بصیرنیا، "تولید علم و فناوری در کشور: جریان‌شناسی، توانمندی‌ها و آسیب‌شناسی،" شورای عالی انقلاب فرهنگی، ۱۳۸۹.

[25] The Joint Task Force for Computing Curricula, *Computing Curricula*, 2005.

[26] H. Jeasani, and T. Marius Truta, "Development and Implementation of a MASTER Program in Computer Information Technology," *American Society for Engineering Education*, pp. 42-48, 2010.

[27] C. A. Perry, *Information Technology and the Curriculum: A Status Report*, 2004.

[28] C. Sugrue, *The Future of Educational Change, International Perspectives*, Routledge, 2008.



سید ابراهیم ابطی کارشناسی خود در رشته کاربرد کامپیوتر و آنالیز سیستم‌ها از مدرسه عالی برنامه‌ریزی و کاربرد کامپیوتر در سال ۱۳۵۶ و کارشناسی ارشد خود را در علوم رایانه از دانشکده ریاضی و علوم کامپیوتر دانشگاه صنعتی شریف در سال ۱۳۶۳ دریافت کرد. از سال ۱۳۶۶ به شکل حقالندرسی و از سال ۱۳۶۸ به شکل رسمی با درجه مربی‌گری به عضویت هیئت علمی دانشگاه صنعتی شریف درآمد که تاکنون به این سمت اشتغال دارد. وی ضمن عضویت در هیئت اجرائی انجمن انفورماتیک ایران و هیئت مدیره انجمن کامپیوتر ایران در دوره‌های گوناگون هم‌اکنون با سمت نایب رئیس انجمن انفورماتیک ایران و عضو هیئت مدیره در انجمن کامپیوتر ایران فعالیت دارد. زمینه‌های پژوهشی مورد علاقه او مهندسی نرم‌افزار، فناوری اطلاعات و آموزش علوم و مهندسی رایانش در سطوح پیش دانشگاهی و دانشگاهی است. ایشان هم‌اکنون نیز عضو کمیته تدوین برنامه‌های درسی دانشگاهی فناوری اطلاعات در وزارت علوم، تحقیقات و فناوری است. از ایشان در سال ۱۳۸۴ به عنوان یکی از پنج فرد برگزیده دانشگاهی، با اهدای لوح و جایزه، توسط ریاست جمهوری وقت تقدیر گردیده است. آدرس پست‌الکترونیکی ایشان عبارت است از:

abtahi@sharif.edu

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۲/۸/۱۲

تاریخ اصلاح: ۹۳/۶/۸

تاریخ قبول شدن: ۹۳/۶/۳۰

نویسنده مرتبط: مهندس سید ابراهیم ابطی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران.

[۱۳] س. قدیمی، و ب. محمدی‌نژاد، "شیوه‌نامه تدوین و بازنگری برنامه درسی"، دفتر برنامه‌ریزی و پشتیبانی آموزش عالی، معاونت آموزشی وزارت علوم، تحقیقات و فناوری، ۱۳۹۰.

[۱۴] س. ا. ابطی، "رایانش، عنوانی برای برنامه تحول در آموزش‌های دانشگاهی مهندسی رایانه و فناوری اطلاعات"، در مجموعه مقالات هفدهمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، ۱۳۹۰.

[۱۵] س. ا. ابطی، "بررسی پیش‌نویس برنامه درسی پیشنهادی ACM & IEEE برای دوره‌های دانشگاهی آموزش فناوری اطلاعات"، ماهنامه گزارش کامپیوتر نشریه انجمن انفورماتیک ایران، شماره ۱۶۴، ۱۳۸۴.

[16] B. M. Lunt, "Information Technology 2008, Curriculum Guidelines for Undergraduate Degree Program in Information Technology," *Journal of Information Technology Education*, vol. 3, no. 1, pp. 12-24, 2008.

[17] R. Kamali, "A Curriculum Model-based on the SIGITE Guidelines," *Journal of Information Technology Education*, vol. 5, no. 3, pp. 42-54, 2008.

[۱۸] س. ا. ابطی، "برنامه‌ای درسی برای دوره دانشگاهی کارشناسی فناوری اطلاعات"، در مجموعه مقالات دومین کنفرانس ملی مدیریت فناوری اطلاعات و ارتباطات، ۱۳۸۴.

[۱۹] س. ا. ابطی، "الگوئی پیشنهادی برای تولید چارچوب برنامه‌های درسی دوره‌های کارشناسی ارشد فناوری اطلاعات"، در مجموعه مقالات پانزدهمین کنفرانس ملی سالانه کامپیوتر انجمن کامپیوتر ایران، مرکز توسعه فناوری نیرو (متن)، ۱۳۸۸.

[۲۰] س. ا. ابطی، "زد (زنجیره‌های درسی): الگوئی کاربردی برای تهیه برنامه درسی در دوره کارشناسی مهندسی فناوری اطلاعات"، در مجموعه مقالات دهمین کنفرانس سالانه انجمن کامپیوتر ایران، پژوهشکده فناوری اطلاعات و ارتباطات ایران، ۱۳۸۳.

[۲۱] س. ا. ابطی، "زنجیره‌های درسی در دوره کارشناسی مهندسی فناوری اطلاعات"، ماهنامه گزارش کامپیوتر نشریه انجمن انفورماتیک ایران، شماره ۱۵۷، ۱۳۸۳.

[۲۲] س. ا. ابطی، "آموزش‌های دانشگاهی فناوری اطلاعات، علوم و مهندسی رایانه، نیازمند گذر از رشد کمی به توسعه کیفی"، ماهنامه گزارش کامپیوتر نشریه انجمن انفورماتیک ایران، شماره ۱۹۸، ۱۳۹۰.

[۲۳] س. ا. ابطی، "دوراندیشی، آینده‌نگری و بهبود مستمر، ضرورت‌های ارتقاء آموزش‌های فناوری اطلاعات و مهندسی رایانه"، ماهنامه گزارش کامپیوتر نشریه انجمن انفورماتیک ایران، شماره ۱۸۲، ۱۳۸۷.

[۲۴] س. ا. ابطی، "دانش افزایی مفاهیم میان‌رشته‌ای راهی برای برنامه‌ریزان درسی دانشگاهی جهت گذر از نگره‌گرایی به نگاه فرارشته‌ای"، ماهنامه گزارش کامپیوتر نشریه انجمن انفورماتیک ایران، شماره ۲۰۶، ۱۳۹۱.

¹ Computing

² Methodology

³ Postdisciplinary

⁴ Disciplinary

⁵ Predisciplinary

⁶ Interdisciplinary

⁷ Reflexive Ideology

⁸ Disciplinarity

⁹ Traditional Discipline

¹⁰ Monodisciplinary

¹¹ Multidisciplinary

¹² Pluridisciplinary

¹³ Participatory

¹⁴ Interdisciplinary

¹⁵ Transdisciplinary

¹⁶ Integrative

-
- 17 Fusion
 - 18 Intradisciplinarity
 - 19 Branch
 - 20 Track
 - 21 Concentration
 - 22 Multi-Contextual
 - 23 Trans-Contextual
 - 24 Pluridisciplinarity
 - 25 Multi-Disciplinarity
 - 26 Inter-Disciplinarity
 - 27 Cross-Disciplinarity
 - 28 Post-Moderns
 - 29 Adisciplinary
 - 30 Anti-Disciplinarity
 - 31 Computing
 - 32 Intradisciplinary
 - 33 Sustainable
 - 34 Strategic

آتوماتای بوخی رکوردی احتمالاتی: معناشناسی سیستم‌های هماهنگی با کانال‌هایی غیرقطعی

هاجر عابدی محمد ایزدی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

ریو زبانی برای توصیف روابط مولفه‌های موجود در سیستم‌های مولفه‌گرا با استفاده از مجموعه پیشینی از کانال‌ها است که در محیط‌های توزیع‌شده و موازی برای هماهنگ کردن فرایندهای همروند استفاده می‌شود. در مجموعه‌ای از مقالات قبلی، برای معناشناسی عملیاتی توصیفات ریو، مدل‌های آتوماتای بوخی رکوردی و ویرایش توسعه یافته آن را ارائه نموده‌ایم. در این مقاله مدل آتوماتای بوخی رکوردی را با پارامترهای احتمالاتی به عنوان معناشناسی برای مدارهای ریو با کانال‌هایی با رفتار غیرقطعی و احتمالاتی، یعنی کانال‌هایی که قابل اعتماد نیستند و امکان از دست رفتن یا خراب شدن پیام در آن‌ها وجود دارد، توسعه می‌دهیم. در این راستا دو نسخه احتمالاتی برای آتوماتای بوخی ارائه می‌دهیم. آتوماتای بوخی احتمالاتی از رکوردهای غیرقطعی ساده که احتمالات را تنها برای پیکربندی بعدی در نظر می‌گیرد. سپس آتوماتای بوخی احتمالاتی از رکوردها برای حالتی که گام‌های بعدی از طریق اعمال احتمالات بر عملگرهای ورودی و خروجی ممکن مشخص می‌شوند را بیان می‌کنیم. همچنین عملگر ضرب دو مدل که معادل پیوند کانال‌ها در ریو است و عملگر مخفی‌سازی پورت‌ها را برای هر دو مدل تعریف می‌کنیم. مقاله همچنین شامل چند مثال موردی و مقایسه قدرت بیان مدل‌های پیشنهاد شده با دیگر مدل‌های احتمالاتی ریو یعنی آتوماتای محدودیت احتمالاتی است.

کلمات کلیدی: زبان هماهنگ‌سازی ریو، سیستم مولفه‌گرا (مبتنی بر اجزا)، آتوماتای بوخی رکوردی، آتوماتای محدودیت، هماهنگ‌کننده، مدل احتمالی، آتوماتای بوخی احتمالاتی از رکوردها.

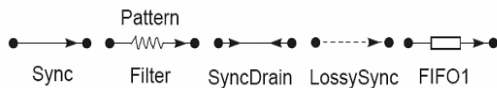
۱- مقدمه

دارد. در زبان‌های هماهنگی مبتنی بر کنترل بسترهای هماهنگی با هدف مشاهده تغییر حالت فرایند ایجاد شده‌اند [۳]. ریو زبان هماهنگی مبتنی بر کانال در این دسته قرار می‌گیرد [۱]. برای مطالعه بیشتر مدل‌های هماهنگی مرجع [۴] مطالعه شود. برای ریو معناشناسی‌های مختلفی تا کنون بیان شده است از جمله آن‌ها آتوماتای محدودیت [۵] و آتوماتای ریو [۶، ۷] و رنگ‌آمیزی رابطه‌ها [۸] و آتوماتای عمدی [۹] و آتوماتای بوخی از رکوردها [۱۰] را می‌توان نام برد. از بین این معناشناسی‌ها تنها برای آتوماتای محدودیت مدل آتوماتای محدودیت احتمالاتی [۱۱] ارائه شده است. با آتوماتای محدودیت احتمالاتی می‌توان کانال‌های ریو با رفتار غیر قابل اعتماد و تصادفی را مدل کرد، هدف توسعه مدل آتوماتای بوخی از رکوردها با اضافه کردن احتمال بر گذارهای موجود است تا بتوان کانال‌های معیوب با رفتار احتمالاتی و غیرقطعی را مدل کرد. این مدل همانند

بزرگتر شدن سیستم‌های نرم افزاری و پیچیده تر شدن آن‌ها باعث شده در طراحی سیستم‌های با اندازه‌های بزرگ مفاهیمی مانند سیستم‌های مبتنی بر مولفه مطرح شود [۱، ۲]. سیستم‌ها به مولفه‌های ساده‌تر تبدیل شده و طراحی می‌شوند و در نهایت مولفه‌ها با هماهنگی یکدیگر خدمتی را ارائه می‌دهند. مدل‌های هماهنگی^۲ وظیفه هماهنگی بین مولفه‌ها را بر عهده دارند و زبان‌های هماهنگی توصیفی برای این مدل‌ها هستند. مدل‌ها و زبان‌های هماهنگی به دو دسته مبتنی بر داده^۳ و مبتنی بر کنترل^۴ دسته‌بندی می‌شوند. تفاوت این دو مدل بیشتر بر جنبه‌های جداسازی محاسبات از هماهنگی است. در دسته مبتنی بر داده یک فضای داده‌ای مشترک وجود دارد. لیندا^۵ اولین روش مدل سازی هماهنگی در این دسته قرار

انواع کانال‌ها

- کانال‌های همگام دارای پایانه‌های مبدا و مقصد است، بافری ندارند و زوج‌هایی از عملیات ورودی / خروجی در صورتی که هر دو پایانه همزمان انجام دهند امکان پذیر است.
- کانال صافی دارای پایانه‌های مبدا و مقصد است اگر داده با الگو مطابقت داشت همانند کانال همگام عمل می‌کند و در غیر این صورت داده را از بین می‌برد.
- کانال‌های همگام اتلافی پایانه مبدا همیشه داده‌ها را می‌پذیرد در صورتی که در پایانه مقصد هیچ درخواستی برای پذیرش داده نباشد داده از بین می‌رود.
- کانال تخلیه همگام از دو پایانه مبدا تشکیل شده است یک عملیات ورودی/خروجی در صورتی انجام می‌شود که هر دو پایانه همزمان عملیات ورودی/خروجی را انجام دهد، داده‌های موجود در این کانال همه از بین می‌روند. کانال تخلیه ناهمگام در این کانال‌ها هر دو پایانه همزمان با هم، هرگز عملیات نوشتن ندارند. همه داده‌های نوشته شده در این کانال نیز از بین می‌روند.
- کانال‌های دهانه همگام^{۱۳} دارای دو پایانه مقصد می‌باشد و زوج عملیات ورودی/خروجی به صورت همزمان انجام می‌شود به این کانال‌ها، کانال‌های تخلیه می‌گویند، کانال‌های دهانه ناهمگام نیز همانند کانال دهانه همگام است ولی عملیات ورودی / خروجی^{۱۴} همزمان وجود ندارد.
- کانال‌های اول ورود / اول خروج^{۱۵} یک بافر نامحدود وجود دارد دارای دو پایانه مبدا و مقصد است از پایانه مبدا همیشه داده را می‌پذیرد و به ترتیب ورود به صف وارد کرده و سپس داده را به پایانه مقصد تحویل می‌دهد. بافر در این کانال امکان عملیات ناهمگام را فراهم می‌نماید.
- کانال‌های اول ورود/ اول خروج (FIFO n) همانند کانال اول ورود اول خروج ولی با ظرفیت محدود است و کانال اول ورود/ اول خروج انتقالی^{۱۶} دارای ظرفیت محدود n است و در صورت پر بودن بافر اگر داده‌ای وارد شود داده‌های قدیمی از ابتدای صف حذف می‌شود و داده جدید جایگزین آن می‌شود.



شکل ۱- نمایش گرافیکی کانال‌های ریو [۱۴]

از اتصال کانال‌ها می‌توان رابط‌های پیچیده‌تر بدست آورد برای اتصال کانال‌ها از عملیات پیوند در ریو استفاده می‌شود. عمل پیوند دو گره را از بین می‌برد و گره جدید ایجاد می‌شود و همه پایانه‌های منطبق بر دو گره قبلی بر گره جدید منطبق می‌شوند. در صورتی که یک گره مقصد باشد و چندین پایانه کانال مقصد بر آن منطبق باشد در صورتی که پیش از یک پایانه داده‌ی مناسب ارائه دهد یکی از آن‌ها به صورت غیر قطعی انتخاب می‌شود. در صورتی که گره‌ای منبع باشد و چندین پایانه منبع به آن متصل باشد در صورتی که داده را می‌پذیرد که همه پایانه‌های منبع آماده پذیرش داده باشد هر گره منبع به عنوان تکرار کننده عمل می‌کند و داده در همه پایانه‌های منبع تکثیر می‌شود. نمایش گرافیکی رابط‌های پیچیده‌تر را مدار ریو می‌گویند. برای مثال در شکل ۲ ترکیب چند کانال نشان داده شده است که یک مدار ریو ایجاد کرده‌اند [۱۵، ۱۶]. برای مطالعه نمونه‌هایی کاربردی بیان شده با استفاده از رابط‌های ریو به [۱۷، ۱۸، ۱۹] رجوع شود. معاشناسی عملیاتی ریو با استفاده از اتوماتای محدودیت و رنگ آمیزی رابط‌ها و اتوماتای بوخی از رکوردها بیان شده است. اتوماتای محدودیت، پذیرنده رشته‌های

مدل اتوماتای بوخی از رکوردها قدرت بیان بیشتری از اتوماتای محدودیت دارد و می‌تواند مفاهیمی چون انصاف را نیز مدل کند و همچنین از الگوریتم‌های واریسی مدل که برای اتوماتای بوخی استفاده می‌شود [۱۲، ۱۳] می‌توان برای آن نیز استفاده کرد. همچنین در این مدل برچسب‌های موجود برای گذارهای اتوماتای رکوردها هستند که نسبت به برچسب‌های اتوماتای محدودیت حامل اطلاعات بیشتری هستند. همان‌طور که خواهیم دید هر مدل اتوماتای محدودیت احتمالاتی قابل تبدیل به اتوماتای بوخی احتمالاتی از رکوردها است. در ادامه ابتدا در بخش ۲ به معرفی زبان ریو و زبان‌ها و رشته‌های زماندار^{۱۱} و اتوماتای محدودیت و نسخه احتمالاتی آن اتوماتای محدودیت احتمالاتی می‌پردازیم، سپس در بخش سوم اتوماتای بوخی از رکوردها را معرفی می‌کنیم و در بخش ۴ به معرفی اتوماتای بوخی احتمالاتی از رکوردها در نسخه احتمالاتی ساده و سپس در بخش ۵ به بیان احتمالات به صورت کلی‌تر می‌پردازیم و مدل اتوماتای بوخی احتمالاتی نه لزوماً ساده را ارائه می‌کنیم. در پایان مثالی از ارتباط دو مولفه با استفاده از کانال‌های غیرقطعی ریو با اتوماتای بوخی احتمالاتی از رکوردها مدل شده است، می‌پردازیم.

۲- ریو و اتوماتای محدودیت

در این بخش ابتدا تعاریفی از مفاهیم اولیه ریو را ارائه می‌دهیم و در ادامه اتوماتای محدودیت را برای معاشناسی عملیاتی آن ارائه می‌دهیم.

۲-۱- ریو

در طراحی پیمان‌های و ساخت نرم‌افزار شامل مولفه‌های نرم‌افزاری مستقل از یکدیگر نیاز به کد چسباننده^{۱۲} می‌باشد که مولفه‌ها بتوانند در کنار یکدیگر سیستم را تشکیل دهند. ریو را می‌توان به عنوان کد چسباننده در مهندسی نرم‌افزار مبتنی بر مولفه نام برد. مولفه‌های سیستم با استفاده از اتصالات ریو با یکدیگر تعامل دارند و همکاری می‌کنند. ریو مدل هماهنگی بیرونی است از این‌رو از محاسبات درون موجودیت‌هایی که هماهنگی را در آنها ایجاد می‌کند اطلاعی ندارد.

اتصال کننده‌ها از کانال‌ها و پایانه‌ها که به صورت گره و یال نمایش می‌دهند تشکیل شده‌اند. صفر یا چند پایانه می‌توانند به هر گره متصل باشند و هر پایانه کانال دقیقاً به یک گره متصل است. برای هر گره مجموعه پایانه‌های متصل به آن را می‌توان به دو مجموعه $Src(N)$ مجموعه کلیه پایانه‌های مبدا متصل به گره N و مجموعه $Snk(N)$ کلیه پایانه‌های مقصد متصل به N ، افراز کرد.

- یک گره N را یک گره منبع می‌گویند اگر $Src(N) \neq \emptyset$ و $Snk(N) = \emptyset$ باشد.
- یک گره N را یک گره مقصد می‌گویند اگر $Snk(N) \neq \emptyset$ و $Src(N) = \emptyset$ باشد.
- یک گره N را یک گره ترکیبی می‌گویند اگر $Snk(N) \neq \emptyset$ و $Src(N) \neq \emptyset$ باشد.

گره‌ها در ریو مشخص کننده مکان نیستند. مولفه‌های بیرونی امکان اتصال با گره‌های ترکیبی را ندارد. کانال‌ها شناسه‌ی مشخصی ندارند ولی پایانه‌های آنها دارای شناسه هستند. همان‌طور که بیان شد پایانه‌های کانال به دو صورت هستند پایانه مبدا که داده‌ها را وارد کانال می‌کند و پایانه مقصد، داده‌ها را از کانال خارج می‌کند. یک کانال می‌تواند دارای دو پایانه مبدا و یا دو پایانه مقصد و یا یک پایانه مبدا و یک پایانه مقصد باشد. در ریو انواع کانال‌ها از جمله کانال‌های سنکرون و آسنکرون و یا کانال پر اتلاف و کانال صافی (دارای الگو) و کانال تخلیه و ... وجود دارد. در شکل ۱ نمایش گرافیکی کانال‌ها نشان داده شده است.

و همچنین $\theta.N$ نیز دنباله‌ای از نام پورت‌ها و زیر مجموعه $2^{\mathcal{N}}$ است. در آن $\theta.N = \theta.N(0), \theta.N(1), \dots$ به صورت کلی به شکل زیر تعریف می‌شود:

$$\theta.N(k) = \{A_i \in \mathcal{N} | a_i(l) = \theta.time(k) \text{ for some } l \in \{0, 1, 2, \dots\}\} \quad (3)$$

مجموعه‌ای از نام پورت‌ها است که در لحظه $\theta.time(k)$ یک مقدار داده‌ای بر آن‌ها وارد یا خارج می‌شود.

اگر تابع انتساب داده $\delta: \mathcal{N} \rightarrow D$ به پورت‌ها را به صورت δ تعریف کنیم، به شرطی که \mathcal{N} تهی نبوده و زیر مجموعه‌ای از مجموعه اسامی پورت‌ها \mathcal{N} باشد، آن‌گاه یک $\theta.\delta = \theta.\delta(0), \theta.\delta(1), \dots$ دنباله‌ای از داده‌های منتسب شده به پورت‌هاست و $\theta.\delta(k)$ جریان داده‌ای در لحظه $\theta.time(k)$ بر روی همه‌ی پورت‌های که در این لحظه فعال هستند، منظور پورت‌های موجود در $\theta.N(k)$ می‌باشد. به صورت زیر تعریف می‌شود:

$$\theta.\delta(k) = [A_i \rightarrow a_i(l_i) : A_i \in \theta.N(k)] \quad (4)$$

اگر $\{0, 1, 2, \dots\}$ اها اندیس‌های یکتا باشد، $a_i(l_i) = \theta.time(k)$ است. [۲۲]

مثال ۱. زبانی که برای یک کانال اول ورود / اول خروج (FIFO) پذیرفته می‌شود برابر است با:

$$\{((\alpha, a), (\beta, b)) \in TDS \times TDS | \alpha = \beta \wedge a < b < a'\}$$

در این زبان چون ابتدا داده در بافر قرار می‌گیرد و سپس داده در زمان بعدی توسط پایانه دیگر کانال از بافر برداشته می‌شود، زمان دریافت داده توسط پایانه مبدا باید کوچکتر از زمان دریافت داده توسط پایانه مقصد باشد.

با مخفی کردن اطلاعات زمانی در هر دنباله رشته‌ای زماندار می‌توان جریانی از رکوردها ایجاد کرد. بدین صورت برای هر $\theta \in TDS^{\mathcal{N}}$ داریم $\gamma(\theta) \in Rec_{\mathcal{N}}(D)^{\theta}$ و بدین ترتیب برای هر $k \in \mathcal{N}$ رکورد $\gamma(\theta)(k)$ شامل همه‌ی پورت‌ها و داده‌هایی که در زمان $\theta.time(k)$ تغییر کرده‌اند. در واقع برای $n \in \theta.N(k)$ داریم:

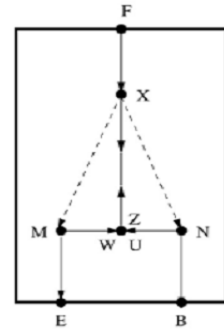
$$\gamma(\theta)(k).n = \theta.\delta(k)_n \quad (5)$$

در هر رکورد نیز $\text{dom}(\gamma(\theta)(k)) = \theta.N(k)$ برقرار است. می‌توان این قانون را برای مجموعه‌ها نیز بسط داد مثلاً برای هر $L_{TDS} \subseteq TDS^{\mathcal{N}}$ داریم: $\gamma(L_{TDS}) = U\{\gamma(\theta) | \theta \in L_{TDS}\}$ علاوه بر آن می‌توان هر جریانی از رکوردها $\rho \in Rec_{\mathcal{N}}(D)^{\theta}$ را با حدس زدن زمانی که داده تغییر کرده با توجه به ترتیب رکوردها به یک دنباله رشته زماندار $\Theta(\rho)$ تبدیل کرد. به صورتی که:

$$\Theta(\rho) = \{\theta | \forall k \geq 0 : (\theta.N(k) = \text{dom}(\rho(k)) \wedge \forall n \in \text{dom}(\rho(k)) : \theta.\delta(k)_n = \rho(k).n)\} \quad (6)$$

همین‌طور می‌توان Θ را به زبان‌هایی که زیر مجموعه از زبان‌های پذیرنده جریانی از رکوردها بسط داد در صورتی که $L \subseteq Rec_{\mathcal{N}}(D)^{\theta}$ داشته باشیم آن‌گاه $\Theta(L) = U\{\Theta(\rho) | \rho \in L\}$ برقرار است.

داده‌ای زماندار [۲۰] است از این‌رو ابتدا به معرفی این رشته‌های زماندار و سپس به بیان اتوماتای محدودیت و اتوماتای بوخی از رشته‌های نامتناهی از رکوردها و عملیات پیوند و مخفی‌سازی پورت بر روی اتوماتاها می‌پردازیم.



شکل ۲- نمایش گرافیکی مداری در ریو [۱۶]

۲-۲- رشته‌ها و زبان‌های زماندار

مجموعه S را در نظر بگیرید مجموعه‌ای از همه‌ی رشته‌های متناهی از S با S^* و مجموعه‌ی همه‌ی رشته‌های نامتناهی بر روی S را با S^{ω} بیان می‌کنیم.

هر رشته $\alpha \in S^{\omega}$ را می‌توان به صورت $\alpha = \alpha(0), \alpha(1), \alpha(2), \dots$ نوشت و مشتق آن را به این صورت $\alpha' = \alpha(1), \alpha(2), \dots$ نشان می‌دهیم. در واقع برای هر رشته α شرط $\alpha'(n) = \alpha(n+1)$ برقرار است.

تعریف ۱. مجموعه‌ی رشته‌های داده‌ای زماندار روی مجموعه داده‌ای D به صورت زیر تعریف می‌شود:

$$TDS = \{(\alpha, a) \in D^{\omega} \times \mathbb{R}_{+}^{\omega} | \forall n \geq 0 : a_n < a_{n+1} \text{ and } \lim_{n \rightarrow \infty} a_n = \infty\} \quad (1)$$

هر رشته زماندار (α, a) شامل یک دنباله داده α و یک دنباله صعودی زمانی a است، دنباله زمانی شامل اعداد حقیقی مثبت و بیانگر لحظه‌ای است که داده متناظر به یک پورت وارد یا خارج شده است.

اتوماتای محدودیت رامی‌توان پذیرنده‌ای برای رشته‌های داده‌ای زماندار که به عنوان ورودی یا خروجی بر روی پورت‌های A_1, A_2, \dots, A_n از مولفه‌ها یا رابط‌ها دانست.

تعریف ۲. $TDS^{\mathcal{N}}$ مجموعه‌ای از همه‌ی رشته‌های داده‌ای زماندار که به هر پورت نسبت داده می‌شود برای پورت‌ها بر اساس A_1, A_2, \dots, A_n ترتیب ثابتی در نظر گرفته شده است و i امین رشته‌ای زماندار متعلق به پورت A_i می‌باشد. اگر این ترتیب را در نظر بگیریم می‌توان رشته داده زماندار متعلق به پورت A_i را به صورت $\theta = (\theta|_i) A_i \in \mathcal{N}$ نشان داد. هر زبان رشته داده‌ای زماندار زیر مجموعه‌ای از $TDS^{\mathcal{N}}$ است.

تعریف ۳. اگر $\theta = ((\alpha_1, a_1), \dots, (\alpha_n, a_n)) \in TDS^{\mathcal{N}}$ باشد $\theta.time$ دنباله‌ای از زمان‌هاست که از ادغام زمان‌های موجود در TDS هر پورت بدست می‌آید. در واقع ترتیب صعودی دنباله‌های زماندار موجود در همه پورت‌هاست و به صورت زیر بدست می‌آید:

$$\begin{aligned} \theta.time(0) &= \min\{a_i(0) | i = 1, \dots, n\} \\ \theta.time(j) &= \min\{a_i(k) | a_i(k) > \theta.time(j-1), i = 1, \dots, n, k = 0, 1, \dots\} \end{aligned} \quad (2)$$

۲-۳- آتوماتای محدودیت

قسمت b کانال‌های همگام را با یک حالت مدل کرده‌ایم در این مدل لازم است هر دو پورت فعال باشد و مقدار داده‌ای هر دو پورت A و B یکسان است. کانال‌های اتلافی همگام را نیز می‌توان با یک حالت مدل کرد و با دو رابطه‌ی گذار در صورتی که هر دو پورت فعال باشند مقدار داده‌ای هر دو یکسان می‌شود و در صورتی که پورت B آماده پذیرش داده نباشد، داده دریافت شده در پورت A از بین می‌رود. کانال‌های تخلیه همگام با یک حالت مدل می‌شود، داده دریافتی در این دو کانال از بین می‌رود و مصرف می‌شوند پس هیچ محدودیت داده‌ای برای آن در نظر نمی‌گیریم و تنها فعال بودن دو پورت کافیست.

کانال‌های صافی با دو رابطه‌ی گذار نشان داده شده است در صورتی که مقدار داده‌ای منتسب به پورت A با الگو مطابقت داشته باشد مقدار داده‌ای B نیز برابر مقدار داده‌ای A خواهد شد و در غیر این صورت اگر تنها پورت A فعال باشد و یا مقدار داده‌ای با الگو اعلام شده مطابقت نداشته باشد گذار دوم انجام می‌شود و مقدار داده منتسب به A نیز از بین می‌رود.

تعریف ۶. محدودیت داده‌ای منتسب به حالت q در آتوماتای محدودیت A، در صورتیکه $N \subseteq Q$ و $P \subseteq Q$ باشد با استفاده از قانون زیر بدست می‌آید:

$$dc_A(q, N, P) = \bigvee \{g: q \xrightarrow{N, g} p \text{ for some } p \in P\} \quad (۸)$$

تعریف ۷. اگر $C = (Q, \mathcal{N}, \rightarrow, Q_0)$ آتوماتای محدودیت و Φ چندتایی از رشته‌های داده‌ای زماندار $\varphi \in TDS^M$ باشد، آنگاه یک اجرای نامتناهی از حالات $N = N.\varphi(0)$ وجود دارد و همچنین $q_0 \xrightarrow{N, g} q_1$ است که در آن گذار $r = q_0, q_1, \dots$ و $\varphi.\delta(0) = g$ و $r' = q_0, q_1, \dots$ اجرای نامتناهی برای φ' باشد. اجرای $r = q_0, q_1, \dots$ را یک اجرای آغازین می‌نامیم اگر $q_0 \in Q_0$ باشد. چندتایی رشته‌های داده‌ای زماندار Φ به وسیله یک آتوماتای محدودیت C پذیرفته می‌شود اگر تنها اگر یک اجرای نامتناهی برای Φ در C وجود داشته باشد.

زبانی که آتوماتای محدودیت C ارائه می‌دهد کلیه $\varphi \in TDS^M$ های است که توسط C پذیرفته می‌شود.

$$LTDS(C) = \{\varphi \in TDS^M \mid C \text{ accept } \varphi\} \quad (۹)$$

نکته: در یک آتوماتای محدودیت با حذف محدودیت‌های داده‌ای (gهایی) که ارضا شدنی نیستند تغییری در زبان قابل پذیرش توسط آن ایجاد نمی‌شود.

تعریف ۸. ضرب دو آتوماتای محدودیت $A_1 = (Q_1, \mathcal{N}_1, \rightarrow_1, Q_{01})$ و $A_2 = (Q_2, \mathcal{N}_2, \rightarrow_2, Q_{02})$ برابر با آتوماتای محدودیت $A_1 \times A_2 = (Q_1 \times Q_2, \mathcal{N}_1 \cup \mathcal{N}_2, \rightarrow, Q_{01} \times Q_{02})$ است.

که قوانین گذار در آتوماتای محدودیت نتیجه را می‌توان با استفاده از ۳ قانون زیر بدست آورد:

$$\begin{aligned} q \xrightarrow{N_1, g_1} q', p \xrightarrow{N_2, g_2} p', N_1 \cap N_2 &= N_2 \cap N_1 & (۱) \\ \langle q, p \rangle &\xrightarrow{N_1 \cup N_2, g_1 \wedge g_2} \langle q', p' \rangle \\ q \xrightarrow{N_1, g_1} q', N_1 \cap N_2 &= \emptyset & (۲) \\ \langle q, p \rangle &\xrightarrow{N_1, g_1} \langle q', p \rangle \\ p \xrightarrow{N_2, g_2} p', N_2 \cap N_1 &= \emptyset & (۳) \\ \langle q, p \rangle &\xrightarrow{N_2, g_2} \langle q, p' \rangle \end{aligned} \quad (۱۰)$$

همان‌طور که گفته شد در معناشناسی عملیاتی برای ریو می‌توان از آتوماتای محدودیت استفاده کرد در ادامه در تعریف مجموعه‌ی داده را مجموعه ثابت D در نظر می‌گیریم. آتوماتای محدودیت با زوجی از g از N برچسب گذاری شده است، N زیرمجموعه ناتهی از مجموعه اسامی پورت‌ها $\mathcal{N} = \{A_1, A_2, A_n, \dots\}$ و g را نگهبان^{۱۸} نامیده که محدودیت‌های داده‌ای^{۱۹} بر روی داده‌های موجود در پورت‌ها را بیان می‌کند.

تعریف ۴. محدودیت‌های داده‌ای برای مجموعه متناهی از پورت‌های N و داده‌های D به وسیله گرامر زیر تعریف می‌شود:

$$g ::= \text{false} \mid \text{true} \mid \text{data}(A) = d \mid g_1 \wedge g_2 \mid g_1 \vee g_2 \quad (۷)$$

در اینجا A نام پورت و $\text{data}(A)=d$ یعنی مقدار داده‌ای موجود بر پورت A برابر $d \in \text{Data}$ است.

DC مجموعه‌ای از همه‌ی محدودیت‌های داده‌ای تحت مجموعه پورت‌های \mathcal{N} و مجموع داده D است. اگر \mathcal{N} و D متناهی باشد DC نیز برای همه‌ی N و D متناهی است.

تعریف ۵. آتوماتای محدودیت چهارتایی $C = (Q, \mathcal{N}, \rightarrow, Q_0)$ تعریف می‌شود:

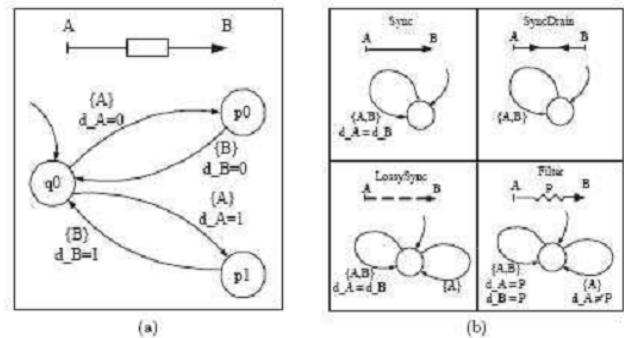
- Q مجموعه متناهی از حالات
- \mathcal{N} مجموعه متناهی از نام پورت‌ها
- \rightarrow زیر مجموعه‌ای متناهی گذار از $Q \times 2^{\mathcal{N}} \times DC \times Q$
- $Q_0 \subseteq Q$ مجموعه‌ی حالات اولیه است.

هر گذار به صورت $E \rightarrow (q, N, g, p)$ نشان داده می‌شود، N مجموعه پورت‌هایی که برای این گذار باید فعال باشند و g نگهبان گذار است که محدودیت داده‌ای برای انجام این گذار را بیان می‌کند.

نکته: دو شرط $N \neq \emptyset$ و $g \in DC(N, \text{Data})$ برای همه‌ی گذارها برقرار است. در صورتی که محدودیت داده‌ای true باشد، روی گذار نمی‌نویسیم. اگر $Q, \text{Data}, \mathcal{N}$ متناهی باشد یک آتوماتای محدودیت را متناهی می‌گویند.

در شکل ۳ کانال‌های ریو با آتوماتای محدودیت مدل شده است:

قسمت a مدلی از کانال اول ورود اول خروج نشان داده شده است. در این شکل فرض کرده‌ایم فضای داده $\{0, 1\}$ است در صورتی که مبدا داده صفر را وارد کند به حالت p_0 می‌رویم، با پذیرش داده توسط پورت B به حالت اولیه باز گشته و در صورتی که پورت A داده یک را بگیرد به حالت p_1 رفته و با پذیرش داده توسط B به حالت اولیه باز می‌گردیم.



شکل ۳- مدل کردن کانال ریو با آتوماتای محدودیت [۲۱]

می‌توان یک اتوماتای محدودیت را به یک اتوماتای محدودیت احتمالی تبدیل کرد بدین منظور به هر گذار آن احتمال ۱ را برای حالت بعدی نسبت می‌دهیم. **تعریف ۱۱.** ضرب دو اتوماتای محدودیت احتمالاتی ساده، $\mathcal{P}_1 = (Q_1, \mathcal{N}_1, \rightarrow_1, Q_{0,1})$ و $\mathcal{P}_2 = (Q_2, \mathcal{N}_2, \rightarrow_2, Q_{0,2})$ در شرایطی که $\mathcal{N}_1 \cap \mathcal{N}_2 \subseteq \mathcal{N}_1^{src} \cup \mathcal{N}_2^{src}$ برقرار است برابری $\mathcal{P}_1 \otimes \mathcal{P}_2 = (Q_1 \times Q_2, \mathcal{N}_1 \cup \mathcal{N}_2, \rightarrow, Q_{0,1} \times Q_{0,2})$ روابط گذار بر اساس قاعده‌های زیر بدست می‌آید:

• قاعده‌ی اول:

$$\frac{q_1 \xrightarrow{N_1, g_1} \pi_1, q_2 \xrightarrow{N_2, g_2} \pi_2, N_1 \cap N_2 = N_2 \cap N_1}{\langle q_1, q_2 \rangle \xrightarrow{N_1 \cup N_2, g_1 \wedge g_2} \pi_1 * \pi_2} \quad (13)$$

برای محاسبه توزیع احتمال از $(\pi_1 * \pi_2)((p_1, p_2)) = \pi_1(p_1) \cdot \pi_2(p_2)$ استفاده می‌کنیم. • قاعده‌ی دوم:

$$\frac{q_1 \xrightarrow{N, g} \pi_1, N \cap N_2 = \emptyset}{\langle q_1, q_2 \rangle \xrightarrow{N, g} \pi_1} \quad (14)$$

در این صورت مقدار توزیع احتمال اگر $q_2 = p_2$ باشد برابر است با $\pi_1(p_1)$ و در غیر این صورت توزیع احتمال برای این تابع گذار صفر می‌شود.

همینطور قانون سومی معادل با قانون دوم برای اتوماتای محدودیت احتمالاتی دوم، تعریف می‌شود.

تعریف ۱۲. مخفی سازی پورت‌های M در اتوماتای محدودیت $p = (Q, \mathcal{N}, \rightarrow, Q_0)$ در صورتی که $M \subset \mathcal{N}^{mix}$ است برابر با اتوماتای محدودیت $(Q, \mathcal{N}/M, \rightarrow_M, Q_{0,M})$ می‌شود و رابطه‌ی گذار به صورت زیر تعریف می‌شود.

$$\frac{q \xrightarrow{N, g} \pi, N' = N \setminus M, g' = \exists M[g]}{q \xrightarrow{N', g'} \pi} \quad (15)$$

$\exists M[g]$ و $Q_{0,M}$ در تعریف ۹ توضیح داده شده است.

۲-۴-۲- اتوماتای محدودیت احتمالاتی

اتوماتای محدودیت احتمالاتی ساده برای مدل کردن کانال‌های صف غیر قابل اعتماد استفاده می‌شود ولی مدل کردن کانال‌هایی که عملیات ورودی-خروجی همزمان با احتمالی ممکن است با شکست مواجه شود، قابل استفاده نیست. برای مثال یک کانال همگام ممکن است داده دریافت شده از پایانه مبدا را به مقصد تحویل ندهد یا پیام را خراب تحویل دهد، هم‌چنین فعالیت‌های که نیاز به پرتاب سکه است که در حل مسائل زیادی از جمله حل مسئله همروندی در سیستم‌های توزیع شده [۲۴]، الگوریتم‌های به کار برده شده در ریاتیک [۲۵] یا در پروتکل‌های امنیتی [۲۶] را نمی‌توان به وسیله اتوماتای محدودیت احتمالاتی ساده مدل کرد. برای مدل کردن این سیستم‌ها لازم است کانال‌هایی ریو که مقادیر خروجی تصادفی ایجاد می‌کنند، داشته باشیم.

تعریف ۹. مخفی سازی پورت $C \in \mathcal{N}$ در اتوماتای محدودیت $A = (Q, \mathcal{N}, \rightarrow, Q_0)$ را به صورت $\exists C[A] = (Q, \mathcal{N} \setminus \{C\}, \rightarrow_C, Q_{0,C})$ نشان داده و قوانین گذار در اتوماتا نتیجه با استفاده از قاعده زیر محاسبه می‌شود:

$$\frac{q \xrightarrow{*} p, p \xrightarrow{N, g} r, N' = N \setminus \{C\} \neq \emptyset, g' = \exists C[g]}{q \xrightarrow{N', g'} r} \quad (11)$$

در اینجا $q \xrightarrow{*} p$ یک مسیر متناهی مانند $q_1 \xrightarrow{\{c\}, g_1} q_2 \xrightarrow{\{c\}, g_2} \dots \xrightarrow{\{c\}, g_n} q_n$ از q به p می‌باشد در آن همه‌ی g_i ها ارضا شدنی و تنها به C وابسته هستند. مجموعه‌ی حالات اولیه $Q_{0,C}$ برای اتوماتا نتیجه برابر است با $Q_{0,C} = Q_0 \cup \{p \in Q \mid \exists q_0 \in Q_0, q_0 \xrightarrow{*} p\}$ شده، به صورت $\forall d \in \text{Data } g[\text{Data}(C)/d]$ بیان کننده این است که به ازای هر عضو از مجموعه داده در صورتی که در گزاره مقدار داده‌ای برای C برابر با آن داده باشد به جای آن مقدار راست q_0 قرار داده و در غیر این صورت مقدار غلط q_1 قرار می‌دهیم.

۲-۴- اتوماتای محدودیت احتمالاتی

کانال‌ها ممکن است رفتار غیر قابل اعتماد داشته باشند و پیام‌ها را خراب کنند یا از بین ببرند از این رو لازم است، مدل اتوماتای محدودیت احتمالاتی را برای بیان احتمال از بین رفتن پیام به وسیله کانال را بیان کنیم، بدین منظور اتوماتای محدودیت به دو گونه احتمالاتی شده است در گونه اول یک مدل احتمال ساده می‌باشد. در آن به ازای هر حالت ورودی و هر گذار ممکن احتمال وقوع حالت بعدی مشخص می‌شود. در گونه عمومی‌تر احتمالات به صورت کلی‌تری ارائه می‌گردد که در هر حالت احتمالات بر مجموعه‌ی نام پورت‌ها و محدودیت‌های داده‌ای اعمال می‌شود. برای مطالعه بیشتر درباره‌ی مدل اتوماتای محدودیت احتمالاتی و کانال‌های معیوب به [۱۱، ۲۲، ۲۳] رجوع شود.

۲-۴-۱- اتوماتای محدودیت احتمالاتی ساده

تعریف ۱۰. اتوماتای محدودیت احتمالاتی ساده چندتایی $\mathcal{P} = (Q, \mathcal{N}, \rightarrow, Q_0)$ است که در آن مقادیر Q_0, \mathcal{N}, Q همانند اتوماتای محدودیت است ولی رابطه‌ی گذار به صورت زیر تعریف می‌شود:

$$\rightarrow \subseteq \cup_{N \in \mathcal{N}} Q \times \{N\} \times DC(N) \times \text{Distr}(Q) \quad (12)$$

$\text{Distr}(Q)$ مجموعه توزیع‌های ممکن برای مجموعه Q است، هر توزیع به شکل تابع $\pi: Q \rightarrow [0,1]$ است، چنان که $\sum_{q \in Q} \pi(q) = 1$ برقرار باشد. در اتوماتای محدودیت احتمالاتی ساده، گزاره‌ها را به صورت $q \xrightarrow{N, g} \pi$ نمایش می‌دهیم. اتوماتای محدودیت احتمالاتی ساده نیز همانند فرایند تصمیم مارکوف برای حل مسئله عدم قطعیت از زمانبندی‌های تعریف شده در تئوری فرایند تصمیم مارکوف استفاده می‌کند. هر زمانبند عدم قطعیت را حل می‌کند و رفتار یک اتوماتای محدودیت احتمالاتی ساده را به عنوان یک زنجیر مارکوف زمان گسسته شرح می‌دهد، در واقع هر زمانبند یک زنجیره مارکوف زمان گسسته نامتناهی را مدل می‌کند و برچسب هر حالت پیشوند متناهی از اجرا در اتوماتای محدودیت است.

تعریف ۱۳. اتوماتای محدودیت احتمالاتی یک چهارتایی $(Q, \mathcal{N}, \rightarrow, Q_0)$ است: $DA(N_2) \times Q_2$ وجود دارد و قابل ترکیب با اتوماتای اول است چنان که یکی از دو شرط زیر برقرار باشد:

- مجموعه قابل شمارش از حالات است. $Q_0 \subseteq Q$
- مجموعه اسامی پورت‌ها است که در سه مجموعه $\mathcal{N}^{src}, \mathcal{N}^{snk}, \mathcal{N}^{mix}$ تقسیم بندی می‌شود.
- رابطه‌ی گذار به صورت $\rightarrow \subseteq Q \times \text{Distr}(U_{N \in \mathcal{N}} \{N\}) \times DA(N) \times DA(N)$ تعریف می‌شود.
- DA محدودیت‌های داده‌ای بر روی اسامی پورت‌های N است. برای هر حالت در تمام رابطه‌های گذار تعریف شده در یک توزیع پورت‌های مبدا، مشترک هستند. بنابراین برای هر گذار $q \rightarrow \Pi$ در صورتیکه $\Pi(N, \delta, p) > 0$ و $\Pi(M, \sigma, r) > 0$ داریم:

$$P_1; P_2 = (Q_1 \times Q_2, \mathcal{N}_1 \cup \mathcal{N}_2, \rightarrow, Q_{0,1} \times Q_{0,2}) \quad (17)$$

رابطه‌ی گذار برای الحاق دو اتوماتای محدودیت احتمالاتی با استفاده از قوانین زیر به دست می‌آید:

- با استفاده از قانون ترکیبی در بالا بیان شد.
- و یا در صورتی که در اتوماتای دوم گذاری وجود داشته باشد که قابل ترکیب با هیچ گذاری از اتوماتای اول نباشد با استفاده از قانون زیر می‌توان این گذار را بدست آورد:

$$\frac{q_2 \rightarrow_2 \Pi_2 \wedge (\Pi_2(N_2, \delta_2, p_2) > 0 \Rightarrow N_2 \cap \mathcal{N}_1 = \emptyset)}{\langle q_1, q_2 \rangle \rightarrow \Pi'_2} \quad (18)$$

در این صورت احتمال توزیع گذار به صورت $\Pi'_2(N, \delta, \langle p_1, p_2 \rangle) = \Pi_2(N, \delta, p_2)$ است در صورتی که $N \subseteq \mathcal{N}_2 \setminus \mathcal{N}_1$ و $p_1 = q_1$ و در همه‌ی حالات دیگر توزیع احتمال برابر با صفر است.

پورت‌های مبدا مشترک در اتوماتای دوم در گذار $\Pi'_2(N, \delta, \langle p_1, p_2 \rangle)$ تعریف بالا، مقدارشان را از پورت مقصد یا ترکیبی در اتوماتای اول می‌گیرند. از این‌رو پورت‌های مبدا در اتوماتای دوم چون در اتوماتای اول مقصد یا ترکیبی می‌باشند در نهایت پورت ترکیبی در اتوماتای الحاقی می‌شوند. در الحاق دو اتوماتا گذاری در اتوماتای دوم را در صورت وجود پورت‌های مشترک با گذاری در اتوماتای اول و مقدار داده‌ای یکسان می‌توان این دو گذار را ترکیب کرد و در صورتی که پورت مشترک وجود داشته باشد ولی با هیچ گذاری از اتوماتای اول قابل ترکیب نباشد از آن گذار صرف‌نظر می‌کنیم. گذارهایی که در یک اتوماتا هستند و هیچ یک از پورت‌های مشترک در گذار نباشند بدون درگیر کردن اتوماتای دیگر و حفظ حالت برای اتوماتای دیگر، انجام می‌شود.

در ضرب دو اتوماتا پورت‌های مشترک، پورتی مبدا در اتوماتای اول یا دوم یا هر دو می‌باشد و نیاز است گاهی مواقع یک گذار از یک اتوماتا با چندین گذار از اتوماتای دیگر به صورت دوره‌ای ترکیب شوند.

در ادامه عملیات ضرب دو اتوماتای محدودیت را تعریف کرده که ترکیبی از عملیات الحاق در اتوماتای محدودیت احتمالاتی و ضرب در اتوماتای محدودیت احتمالاتی ساده می‌باشد. ابتدا چند تعریف برای عملیات ورودی و خروجی قطعی و ورودی مستقل را بیان کرده و سپس ضرب دو اتوماتای محدودیت احتمالاتی را بیان می‌کنیم.

تعریف ۱۶. عملیات ورودی و خروجی قطعی^{۲۳}، اگر P_1 و P_2 را دو اتوماتای محدودیت احتمالاتی در نظر بگیریم و هر نود مشترک یک نود مبدا در P_1 یا P_2 باشد. در صورتی که در اتوماتای P_1 برای حالت q_1 توزیع $\Pi_1 \rightarrow_1 q_1$ وجود داشته باشد، در این توزیع به شرط وجود دو گذار $\Pi_1(N_1, \delta_1, p_1) > 0$ و $\Pi_1(N_1, \delta_1, p_1) > 0$ داشته باشیم $\Pi_1(N_1, \delta_1, p_1) > 0$ و به ازای هر پورت $A \in N_1 \cap \mathcal{N}_2$ داشته باشیم $A \in N_1 \cap \mathcal{N}_2$ $\delta_1.A = \delta_1.A$ این‌گونه $\Pi_1 \rightarrow_1 q_1$ عملیات

هر اتوماتای محدودیت احتمالاتی ساده را می‌توان به اتوماتای محدودیت احتمالاتی تبدیل کرد. بدین منظور هر گذار اتوماتای محدودیت احتمالاتی ساده به صورت $q \xrightarrow{Ng} \pi$ را به $\Pi(N, \delta, \pi)$ تبدیل کرده به شرطی که $\delta \in DA(N)$ و $\delta \models g$ مقدار توزیع احتمال برای یک حالت خاص برابر است با $\Pi(N, \delta, \pi)(N, \delta, p) = \pi(p)$. برای تبدیل اتوماتای محدودیت به اتوماتای محدودیت احتمالاتی لازم است ابتدا اتوماتای محدودیت را به مدل اتوماتای محدودیت احتمالاتی ساده تبدیل کرد سپس آن را به اتوماتای محدودیت احتمالاتی تبدیل می‌کنیم.

تعریف ۱۴. عملیات مخفی سازی پورت‌ها، فرض کنید در اتوماتای محدودیت احتمالاتی $P = (Q, \mathcal{N}, \rightarrow, Q_0)$ پورت‌های ترکیبی $M \subseteq \mathcal{N}^{mix}$ که ارتباطی با مولفه‌های بیرونی ندارند را مخفی کنیم نتیجه اتوماتا $hide(P, M) = (Q, \mathcal{N} \setminus M, \rightarrow_M, Q_{0,M})$ است که در آن رابطه‌ی گذار $q \rightarrow_M \Pi'$ در صورتی وجود دارد که $q \rightarrow \Pi$ در P وجود داشته باشد و مقدار احتمال برای هر حالت q در صورتی که شرط $N \setminus M = N'$ و $\delta|_{N'} = \delta'$ برقرار باشد برابر است با $\Pi'(N', \delta', p) = \sum_{N, \delta} \Pi(N, \delta, p)$

برای ضرب دو اتوماتای محدودیت احتمالاتی باید حالتی را در نظر بگیریم که عملیات ورودی - خروجی به صورت همزمان برای پورت‌های مشترک هر دو اتوماتای محدودیت احتمالاتی انجام شود. فرض می‌کنیم پورت‌های مشترک در دو اتوماتای محدودیت احتمالاتی پورت‌های مبدا در یکی از دو اتوماتا یا در هر دو باشد، ابتدا عملیات الحاق را تعریف می‌کنیم حالتی که اتوماتای دوم ورودی خود را در پورت‌های مشترک از اتوماتای اول می‌گیرد.

تعریف ۱۵. الحاقی^{۲۴} از دو اتوماتای محدودیت احتمالاتی $P_i = (Q_i, \mathcal{N}_i, \rightarrow_i, Q_{0,i})$ $i = 1, 2$ فرض می‌کنیم اشتراک اسامی پورت‌ها در دو اتوماتای محدودیت احتمالاتی زیر مجموعه پورت‌های مبدا در اتوماتای محدودیت احتمالاتی دوم است و توزیع گذار برای حالت $\langle q_1, q_2 \rangle$ تابع Π خواهد بود که رابطه‌ی گذار بدین صورت به دست می‌آید:

$$\Pi(N, \delta, \langle p_1, p_2 \rangle) = \Pi_1(N_1, \delta_1, p_1) \cdot \Pi_2^{N_1 \cap \mathcal{N}_2^{src}, \delta|_{N_1 \cap \mathcal{N}_2^{src}}} (N \cap \mathcal{N}_2, \delta|_{N \cap \mathcal{N}_2}, p_2) \quad (16)$$

در رابطه‌ی بالا در صورتی که گذار $\Pi_1(N_1, \delta_1, p_1)$ در اتوماتای اول وجود داشته باشد، و برای هر زوج $(M, \sigma) = (N_1 \cap \mathcal{N}_2, \delta|_{N_1 \cap \mathcal{N}_2})$ تساوی $\Pi_1(N_1, \delta_1, p_1) > 0$ داشته باشیم، در اتوماتای دوم گذاری $\Pi_2^{M, \sigma}$ با توزیع $\Pi_2^{M, \sigma}$ $U_{N_2 \cap \mathcal{N}_2} \{N_2\} \times$

(تعریف ۱۶) و ورودی مستقل (تعریف ۱۷) بدست می‌آید. برای مطالعه بیشتر درباره اتوماتای محدودیت احتمالاتی به [۱۱] مراجعه شود.

۳- اتوماتای بوخی از رکوردها

اتوماتای دیگری که به عنوان معناشناسی سیستم‌های هماهنگی ارائه شده است و پذیرنده رکوردهای زماندار می‌باشد اتوماتای بوخی از رکوردهاست. البته در اینجا نیز ترتیب منطقی زمان‌ها مهم است و از زمان دقیق رویدادها که در دنباله زماندار وجود دارد استفاده‌ای نمی‌شود. در این اتوماتا ساختار رکوردها برای بیان محدودیت‌های همزمان بر روی پورت‌ها استفاده شده است.

تعریف ۱۹. فرض کنید \mathcal{N} مجموعه متناهی از نام پورت‌ها و D مجموعه متناهی از داده‌ها، $\text{Rec}_{\mathcal{N}}(D) = \mathcal{N} \rightarrow D$ مشخص کننده مجموعه‌ای از رکوردها می‌باشد هر رکورد تابع جزئی از \mathcal{N} به D است. دامنه یک رکورد $r \in \text{Rec}_{\mathcal{N}}(D)$ را با $\text{dom}(r)$ نمایش می‌دهند. رکوردها دارای بار اطلاعات مثبت و منفی هستند، یعنی در هر رکورد r تنها پورت‌های موجود در دامنه رکورد در تبادل داده‌ای توسط آن رکورد شرکت می‌کنند و پورت‌های $n \in \mathcal{N} \setminus \text{dom}(r)$ هیچ تبادل داده‌ای انجام نمی‌دهند. محتوای رکورد r به صورت $r = [n_1 = d_1, \dots, n_k = d_k]$ قابل نمایش است. در این صورت $\text{dom}(r) = \{n_1, \dots, n_k\}$ و برای $1 \leq i \leq k$ $r(n_i) = d_i$ است. با $r.n$ می‌توان به مقدار داده‌ای پورت n دسترسی پیدا کرد، در هر رکورد تنها مقادیر داده‌ای منتسب به پورت‌های آن می‌توانند تغییر کنند.

رکوردی که دامنه تهی دارد $(\text{dom}(r) = \emptyset)$ را با τ نشان می‌دهیم، ترتیب عناصر در رکوردها مهم نیست. می‌توان اطلاعات موجود در یک رکورد را با اضافه کردن یک پورت دیگر و داده منتسب به آن افزایش داد برای این کار می‌نویسیم $\text{dom}(r) \cup \{n\}$ را $r[n:=d]$ در این صورت دامنه رکورد r تغییر کرده و برابر با $\text{dom}(r) \cup \{n\}$ می‌شود و مقدار d به پورت n منتسب می‌شود و مقادیر داده‌ای منتسب به پورت‌های دیگر بدون تغییر می‌ماند. با مخفی‌کردن پورت n از رکورد r ، این پورت از دامنه r ، بدین صورت $\text{dom}(r \setminus n) = \text{dom}(r) \setminus \{n\}$ حذف می‌شود هم‌چنین مقادیر داده‌ای برای پورت‌های دیگر هیچ تغییری نمی‌کند.

تعریف ۲۰. دو رکورد $r_1 \in \text{Rec}_{\mathcal{N}_1}(D)$ و $r_2 \in \text{Rec}_{\mathcal{N}_2}(D)$ با هم سازگارند اگر $\forall n \in \text{dom}(r_1) \cap \text{dom}(r_2) = \text{dom}(r_1) \cap \mathcal{N}_2 = \text{dom}(r_2) \cap \mathcal{N}_1$ شرط $\text{dom}(r_2) : r_1.n = r_2.n$ برقرار باشد. در این صورت می‌توان دو رکورد را با هم الحاق کرده و در این صورت نتیجه $r_1 \cup r_2$ می‌شود. مجموعه پورت‌ها برابر است با $\mathcal{N}_1 \cup \mathcal{N}_2$ و مقدار داده‌ای برای هر پورت برابر است با مقدار داده‌ای منتسب به آن پورت در رکوردی که به آن تعلق دارد [۲۷].

تعریف ۲۱. ابتدا فرض می‌کنیم $\theta \in \text{TDS}^M$ است و جریانی از رکوردها را با حذف اطلاعات زمانی از رکوردها ایجاد می‌کنیم، جریانی نامتناهی از رکوردها بدین صورت $\gamma(\theta) \in \text{Rec}_{\mathcal{N}}(D)^\omega$ ایجاد می‌شود. در حالی که k امین رکورد موجود در $\gamma(\theta)$ شامل همه پورت‌هایی است که در زمان $\theta.time(k)$ داده‌هایشان تغییر کرده است. به بیانی دیگر برای هر $k \in \mathbb{N}$ فرض زیر برقرار است:

$$\text{dom}(\gamma(\theta(k))) = \theta.N(k) \quad (۲۲)$$

$$\gamma(\theta)(k).n = \theta.\delta(k)_n$$

ابتدا با $\theta.time$ ترتیب زمانی دنباله را مشخص می‌کنیم، سپس بر اساس این دنباله $\theta.N$ محاسبه می‌شود، که ترتیب رکوردها را نشان می‌دهد، سپس برای هر $n \in \theta.N(k)$ مقدار داده منتسب به آن را می‌توان مشخص کرد [۲۸].

ورودی و خروجی قطعی برای اتوماتای دوم است. در این صورت گذار قابل ترکیب برای $q_1 \rightarrow_1 \Pi_1$ در Π_2 گذاری است که در صورت وجود گذار گذار $\Pi_1(N_1, \delta_1, p_1) > 0$ در اتوماتای اول و گذار $\Pi_2(N_2, \delta_2, p_2) > 0$ در اتوماتای دوم، پورت مشترک در گذار موجود در اتوماتای اول با اسامی پورت‌های اتوماتای دوم برابر با پورت‌های مشترکی که در گذار اتوماتای دوم با اسامی پورت‌های اتوماتای اول با مقادیر یکسان باشد. به عبارتی $N_1 \cap N_2 = N_2 \cap N_1$ و $\delta_1.A = \delta_2.A$ برای همه پورت‌های $A \in N_1 \cap N_2$ که می‌باشد در این صورت اگر $q_1 \rightarrow_1 \Pi_1$ یک ورودی و خروجی قطعی برای $q_2 \rightarrow_2 \Pi_2$ باشد آن‌گاه این دو گذار قابل ترکیب $\Pi = \Pi(q_1, q_2)$ و توزیع احتمالی به صورت زیر می‌توان تعریف کرد:

$$\Pi(N, \delta, \langle p_1, p_2 \rangle) = \Pi_1(N \cap \mathcal{N}_1, \delta|_{N \cap \mathcal{N}_1}, p_1) \cdot \Pi_2(N \cap \mathcal{N}_2, \delta|_{N \cap \mathcal{N}_2}, p_2) \quad (۱۹)$$

به این گذار ترکیبی ساده می‌گویند. برای هر $q_2 \rightarrow_2 \Pi_2$ در اتوماتای P_2 نیز می‌توان یک ورودی و خروجی قطعی نسبت به P_1 به همین صورت تعریف کرد.

تعریف ۱۷. عملیات ورودی مستقل^{۲۴}، اگر P_1 و P_2 را دو اتوماتای محدودیت احتمالاتی در نظر بگیریم و فرض کنیم هر نود مشترک یک نود مبدا در P_1 یا P_2 در هر دو است. در صورتی که در اتوماتای P_1 گذار $q_1 \rightarrow_1 \Pi_1$ وجود دارد این گذار $q_1 \rightarrow_1 \Pi_1$ ورودی مستقل نسبت به P_2 است اگر $\Pi_1(N_1, \delta_1, p_1) > 0$ و $N_1 \cap \mathcal{N}_1^{\text{src}} \cap (\mathcal{N}_2^{\text{snk}} \cup \mathcal{N}_2^{\text{mix}}) = \emptyset$ و ورودی مستقل روی P_2 و $q_2 \in Q_2$ است و علاوه بر آن فرض کنیم برای هر $\eta \in DA(L)$ و $L \subseteq \mathcal{N}_2^{\text{src}} \setminus \mathcal{N}_1$ $(M, \sigma) = (N_1 \cap \mathcal{N}_2, \delta_1|_{N_1 \cap \mathcal{N}_2})$ داریم. در این صورت تابع گذار قابل ترکیب در P_2 بر اساس یکی از دو حالت زیر انتخاب می‌شود:

- $q_2 \rightarrow_2 \Pi_2^{M, \sigma}$ یک گذار قابل ترکیب در P_2 است چنان‌که دو شرط زیر برقرار باشد:

$$N_2 \cap \mathcal{N}_2^{\text{src}} = (N_1 \cap \mathcal{N}_2^{\text{src}}) \cup L \quad \text{و} \quad \forall A \in L \delta_2.A = \eta.A \quad (۲۰)$$

$$N_2 \cap \mathcal{M}_1 = M \quad \text{و} \quad \forall A \in M \delta_2.A = \eta.A$$

- در صورتی که $M = L = \emptyset$ است توزیع $\text{Idle}_{q_2} = \Pi_2^{M, \sigma}$ برقرار است (در تعریف ۱۵ توضیح داده شده است).

گذار ترکیبی دو اتوماتا برابر است با $\Pi = \Pi(q_1, q_2)$ گذاری از حالت (q_1, q_2) که از حالت q_1 آغاز شده است با فرض $\delta \in N \subseteq \mathcal{N}_1 \cup \mathcal{N}_2$ ، مقدار احتمال برابر است با:

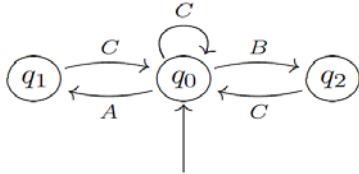
$$\Pi(N, \delta, \langle p_1, p_2 \rangle) = \Pi_1(N_1, \delta_1, p_1) \cdot \Pi_2^{N_1 \cap \mathcal{N}_2, \delta_1|_{N_1 \cap \mathcal{N}_2}}(N_2, \delta_2, p_2) \quad (۲۱)$$

اگر $q_1 \rightarrow_1 \Pi_1$ یا $L \neq \emptyset$ یک ورودی و خروجی قطعی برای P_2 نباشد آن‌گاه ترکیب گذارها به این صورت را غیر ساده گویند. به طور مشابه گذار ترکیبی غیر ساده از (q_1, q_2) آغاز شوند از q_2 نیز به همین صورت تعریف می‌شود.

تعریف ۱۸. ضرب دو اتوماتای محدودیت احتمالاتی از P_1 و P_2 برابر است با اتوماتای محدودیت احتمالاتی $Q_0, 1 \times P_1 \times P_2 = (Q_1 \times Q_2, \mathcal{N}_1 \cup \mathcal{N}_2, \rightarrow, Q_0, 1 \times Q_0, 2)$ که گذارهای قابل ترکیب از روابط تعریف شده در ورودی و خروجی قطعی

مجموعه‌ی حالات پایانی مجموعه‌ای از مجموعه‌ها به صورت $\mathcal{F} \subseteq 2^Q$ است، شرط پذیرش رشته ω در آتوماتای بوخی تعمیم یافته اگر و تنها اگر یک مسیر محاسبه مانند π وجود داشته باشد چنانکه برای هر $F \in \mathcal{F}$ حداقل یکی از حالات موجود در F در مسیر π نامتناهی بار اتفاق بیافتد [۳۰].

مثال ۳. در آتوماتای بوخی تعمیم یافته در شکل ۵ مجموعه الفبا $\{A, B, C\}$ است و دو مجموعه پذیرش $F_1 = \{q_1\}$ و $F_2 = \{q_2\}$ داریم، زبان قابل پذیرش توسط این آتوماتا کلیه رشته‌هایی است که هر دو حرف A, B نامتناهی بار دیده شوند.



شکل ۵- آتوماتای بوخی تعمیم یافته با مجموعه حالات پایانی $\{q_1, q_2\}$

نکته: در صورتی که در آتوماتای بوخی تعمیم یافته مجموعه حالات پذیرش خالی $\mathcal{F} = \emptyset$ باشد، در این صورت یک رشته ω در صورتی پذیرفته می‌باشد که یک پذیرش نامتناهی در آتوماتا وجود داشته باشد و زبان این آتوماتا کلیه رشته‌های است که یک مسیر پذیرش نامتناهی برای آن وجود داشته باشد ولی در آتوماتای بوخی معمولی اگر مجموعه حالات پایانی تهی باشد هیچ رشته‌ای پذیرفته نمی‌شود و زبان آن نیز تهی است.

مثال ۴. نمونه‌ای از کانال‌های ریو در شکل ۶ که با آتوماتای بوخی از رکوردها مدل شده است، نشان داده شده است. در اینجا فرض می‌کنیم همه کانال‌ها از پورت A به پورت B می‌باشد و مجموعه داده‌ای $D = \{d, d'\}$ است در شکل زیر برای سادگی گذارهای با رکوردهای متفاوت که مبدا و مقصد یکی است از یک کمان استفاده شده است و برچسب‌ها با کاما از یکدیگر جدا شده‌اند در حالی که کانال صافی داریم فرض شده است مقدار الگو برابر d است.

تعریف ۲۵. تبدیل آتوماتای محدودیت به آتوماتای بوخی از رکوردها [۲۸]، ابتدا باید همه‌ی برچسب‌های گذار آتوماتای محدودیت را به رکوردها تبدیل کنیم از این‌رو برای هر محدودیت داده g مجموعه‌ای از انتساب داده‌ای از N به D وجود دارد و هر عضو این مجموعه در واقع رکوردی در $Rec_{\mathcal{N}}(D)$ می‌باشد. بنابراین اگر آتوماتای محدودیت $(Q, \mathcal{N}, \rightarrow, Q_0)$ را در شرایطی که \mathcal{N}, D هر دو متناهی باشند به آتوماتای بوخی از رکوردها تبدیل کنیم آنگاه $F = Q$ و گذارهای موجود در آتوماتای بوخی از رکوردها بدین صورت بدست می‌آید:

$$\Delta = \left\{ (q, r, q') \mid \exists q \xrightarrow{(N,g)} q', \exists \delta: N \rightarrow D: \delta \models g, \text{dom}(r) = N \text{ and } \forall n \in N: r.n = \delta(n) \right\} \quad (24)$$

نکته. بر اساس قضیه اثبات شده در [۳۰] برای آتوماتای محدودیت $A = \langle Q, \mathcal{N}, \rightarrow, Q_0 \rangle$ ویژگی زیر برقرار است:

$$\gamma(L_{TDS}(A)) = L(B(A)) \text{ و } \Theta(L(B(A))) = L_{TDS}(A) \quad (25)$$

مثال ۵. در شکل ۷ نمونه‌ای از یک آتوماتای محدودیت (a) با مجموعه پورت $\{A, B\}$ و مجموعه داده‌ی $D = \{d, d'\}$ به آتوماتای بوخی از رکوردها (b) تبدیل شده است.

از آنجایی که آتوماتای بوخی از رکوردها توسعه‌ای بر آتوماتای بوخی است ابتدا معرفی از آتوماتای بوخی و پذیرنده رشته‌ی نامتناهی در آتوماتای بوخی و زبان آنرا می‌دهیم [۲۹].

تعریف ۲۲. یک آتوماتای بوخی چندتایی $(Q, \Sigma, Q_0, \Delta, F)$ است:

- Q مجموعه‌ای از حالات است.
- Σ مجموعه نامتهی و متناهی از سیمبل‌ها، الفبا گفته می‌شود.
- $\Delta \subseteq (Q \times Rec_{\mathcal{N}}(D) \times Q)$ تابع گذار گفته می‌شود.
- Q_0 زیر مجموعه نامتهی از Q است.
- $F \subseteq Q$ حالات پذیرنده^{۲۵} می‌باشد.

یک رشته بی‌انتهای w از الفبای Σ توسط آتوماتای بوخی پذیرفته می‌شود اگر حداقل یکی از وضعیت‌های پایانی (پذیرنده) نامتناهی بار در اجرای رشته w اتفاق بیافتد.

تعریف ۲۳. در آتوماتای بوخی از رکوردها، مجموعه الفبا رکوردها است که زیر مجموعه‌ای از $Rec_{\mathcal{N}}(D)$ است و هر رابطه گذار به صورت $(q, r, p) \in \Delta$ است. هر رابطه‌ی گذار را می‌توان به صورت $q \xrightarrow{r} p$ نیز نوشت.

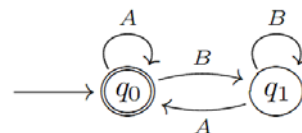
یک آتوماتای بوخی از رکوردها می‌تواند شامل گذار τ نیز باشد این گذارها می‌توانند به عنوان فعالیت‌های داخلی سیستم باشند که هیچ پورتهی در آن مشارکت ندارند و می‌توان از آنها صرف‌نظر کرد و تنها گذارهای قابل نمایش را مشاهده کرد.

زبان یک آتوماتای بوخی از رکوردها مجموعه‌ی از همه‌ی رشته‌های بی‌انتهای به صورت $\omega \in \Sigma^\omega$ است چنان‌که اجرایی برای پیمایش رشته به صورت نامتناهی در آتوماتای بوخی از رکوردها وجود داشته باشد و حداقل یکی از وضعیت‌های پایانی نامتناهی بار ملاقات شود. زبان آتوماتای B را با $L(B)$ نمایش می‌دهند که مجموعه‌ای از همه‌ی رشته‌های بی‌انتهای پذیرفته شده توسط آن است. B_1, B_2 را معادل (هم ارز) گویند اگر زبان‌های هر دو برابر باشد. **زبان قابل رویت^{۲۶}** برای یک آتوماتای بوخی از رکوردها B برابر است با:

$$L_{vis}(B) = \{ \rho \in Rec_{\mathcal{N}}(D)^\omega \mid \exists \omega \in L(B): \rho = vis(\omega) \} \quad (23)$$

$vis(\omega)$ دنباله‌ای است که از حذف τ ها از ω بدست می‌آید [۲۷]. آتوماتای بوخی تعریف شده در بالا غیرقطعی می‌باشد، در صورتی که تمایل به داشتن آتوماتای بوخی قطعی داشته باشیم باید $|Q_0| \leq 1$ برقرار باشد و رابطه‌ی گذار را به تابع گذار بدین صورت $\Delta: (Q \times \Sigma) \rightarrow Q$ تبدیل کنیم.

مثال ۲. در آتوماتای بوخی شکل ۴ حالت اولیه q_0 است و تنها حالت مورد پذیرش نیز q_0 می‌باشد. پس زبان مورد پذیرش این آتوماتا زبانی است که رشته‌هایی را بپذیرد که حالت q_0 را نامتناهی بار مشاهده کند یا به عبارتی کلیه رشته‌هایی که حرف A در آن نامتناهی بار مشاهده شود.



شکل ۴- آتوماتای بوخی از رکوردها

تعریف ۲۴. آتوماتای بوخی تعمیم یافته^{۲۷} نوعی دیگر از آتوماتای بوخی می‌باشد که شرط پذیرش آن نیازمند مشاهده کردن نامتناهی بار از چندین مجموعه از حالات پایانی F_1, \dots, F_k است، به صورت رسمی‌تر می‌توان گفت یک آتوماتای بوخی تعمیم یافته با چندتایی $(Q, \Sigma, Q_0, \Delta, \mathcal{F})$ $B = \langle Q, \Sigma, Q_0, \Delta, \mathcal{F} \rangle$ می‌شود که

تعریف ۲۷. حاصل ضرب دو اتوماتای بوخی $B_1 = (Q_1, \Sigma, \Delta_1, Q_{01}, F_1)$ و $B_2 = (Q_2, \Sigma, \Delta_2, Q_{02}, F_2)$ با مجموعه الفبای یکسان یک اتوماتای بوخی تعمیم یافته می‌شود:

$$B_1 \bowtie B_2 = (Q_1 \times Q_2, \Sigma, \Delta, Q_{01} \times Q_{02}, \{F_1 \times Q_2, F_2 \times Q_1\}) \quad (26)$$

رابطه‌ی گذار بدین صورت بدست می‌آید:

$$\frac{q \xrightarrow{a} q' \quad p \xrightarrow{a} p'}{\langle q, p \rangle \xrightarrow{a} \langle q', p' \rangle} \quad (27)$$

زبان مورد قبول برای ضرب دو اتوماتای بوخی برابر با اشتراک زبان متناظر به هر یک از دو اتوماتا است.

در ادامه ضرب دو اتوماتای بوخی از رکوردها با مجموعه اسامی پورت‌های متفاوت را بیان می‌کنیم نتیجه ضرب این دو اتوماتا نیز اتوماتای بوخی از رکوردهای تعمیم یافته می‌شود.

تعریف ۲۸. حاصل ضرب دو اتوماتای بوخی از رکوردها $B_1 = (Q_1, \text{Rec}_{\mathcal{N}_1}(D), \Delta_1, Q_{01}, F_1)$ و $B_2 = (Q_2, \text{Rec}_{\mathcal{N}_2}(D), \Delta_2, Q_{02}, F_2)$ یک اتوماتای بوخی تعمیم یافته از رکوردها می‌باشد:

$$B_1 \bowtie B_2 = (Q_1 \times Q_2, \text{Rec}_{\mathcal{N}_1 \cup \mathcal{N}_2}(D), \Delta, Q_{01} \times Q_{02}, \{F_1 \times Q_2, F_2 \times Q_1\}) \quad (28)$$

- در صورتی که r_1, r_2 سازگار باشند رابطه‌ی گذار این گونه تعیین می‌شود:

$$\frac{q \xrightarrow{r_1} q' \quad p \xrightarrow{r_2} p' \quad \text{comp}(r_1, r_2)}{\langle q, p \rangle \xrightarrow{r_1 \cup r_2} \langle q', p' \rangle} \quad (29)$$

و در صورتی که r_1, r_2 با هیچ رکوردی سازگار نباشند برای محاسبه گذار از دو قاعده زیر استفاده می‌کنیم:

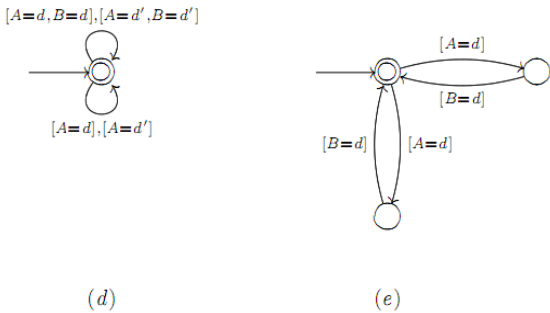
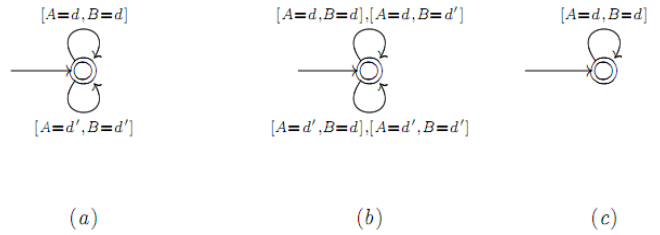
- در صورتی که گذار با رکورد r_1 موجود در اتوماتای اول با هیچ گذاری در اتوماتای دوم سازگار نباشد و مجموعه اسامی پورت‌های موجود در دامنه رکورد r_1 با مجموعه اسامی پورت‌های اتوماتای دوم اشتراکی نداشته باشد:

$$\frac{q \xrightarrow{r_1} q' \quad \text{dom}(r_1) \cap \mathcal{N}_2 = \emptyset}{\langle q, p \rangle \xrightarrow{r_1} \langle q', p \rangle} \quad (30)$$

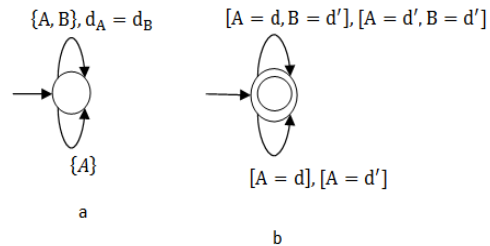
- به صورت معادل می‌توان برای اتوماتای دوم نیز فرمول زیر را تعریف کرد:

$$\frac{p \xrightarrow{r_2} p' \quad \text{dom}(r_2) \cap \mathcal{N}_1 = \emptyset}{\langle q, p \rangle \xrightarrow{r_2} \langle q, p' \rangle} \quad (31)$$

مثال ۶. در شکل ۹ اتوماتای بوخی از رکوردها قسمت a کانال صف FIFO از پورت A به B را مدلی می‌کند و قسمت b کانال صف از پورت B به C را می‌بینیم، مجموعه داده $D = \{d\}$ است با استفاده از عملیات پیوند دو صف به هم متصل شده‌اند و صف با ظرفیت بافر ۲ ایجاد می‌شود:

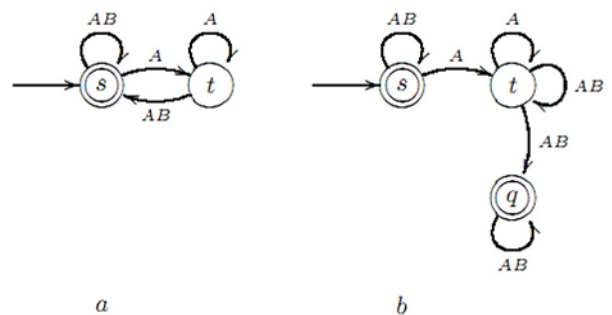


شکل ۶- مدل‌هایی از کانال‌های ریو (a) کانال همگام (b) کانال دهانه همگام (c) کانال صافی (d) کانال همگام اتلافی (e) کانال صف اول ورود اول خروج [۱۴]



شکل ۷- تبدیل اتوماتای محدودیت به اتوماتای بوخی از رکوردها [۱۴]

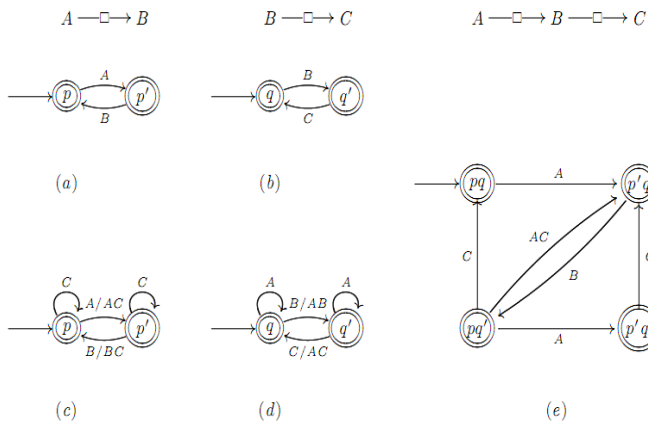
تعریف ۲۶. رابطه‌های ریو منصف، در اتوماتای محدودیت نمی‌توان رابطه‌های منصف را مدل کرد ولی با وجود حالات پایانی در اتوماتای بوخی از رکوردها و شرط پذیرش یک رشته می‌توان رابطه‌های ریو منصف را نیز مدل کرد. برای مثال در شکل ۸.a یک کانال اتلافی همگام را مدل کرده‌ایم ولی نسبت به مدل شکل ۶ دارای این شرط می‌باشد که همه‌ی داده‌ها نمی‌توانند از بین بروند و در شکل ۸.b شرط قوی‌تری برای این کانال گذاشته‌ایم تعداد محدودی داده‌ها در کانال می‌توانند از بین بروند [۳۱، ۳۲].



شکل ۸- رابطه‌های ریو منصف با اتوماتای بوخی از رکوردها [۱۴]

می‌توان حدس زد را اضافه می‌کنیم سه حدس وجود دارد: یکی اینکه محیط از پورت جدید استفاده نمی‌کند و همان گذارهای قبلی را لحاظ کرده، دوم اینکه محیط فقط از این پورت جدید استفاده کرده بدون استفاده از پورت‌های دیگر و یا محیط پورت n با محدودیت‌های داده‌ای دیگر ترکیب کرده در واقع پورت جدید را به همه‌ی رکوردهای موجود بر روی گذارها به روشی که قبلاً بیان شده است اضافه می‌کنیم [۲۷].

مثال ۷. در شکل ۱۰ از ضرب دو آتوماتا با استفاده از روش جدید استفاده کرده قسمت c گسترش مدل صف اول a می‌باشد با رجوع به نام پورت C و قسمت d نیز گسترش مدل آتوماتای بوخی از رکوردهای صف دوم b می‌باشد.



شکل ۱۰- ضربی از دو آتوماتای بوخی از رکوردها به روش غیرمستقیم [۱۴]

عملیات گسترش دو پورت در یک آتوماتای بوخی از رکوردها ترتیب تأثیری ندارد و برای دو پورت m و n هر دو آتوماتای زیر معادل هستند $(B \uparrow n) \uparrow m = (B \uparrow m) \uparrow n$ ، می‌توان یک آتوماتا را با رجوع به یک مجموعه‌ای از N نام پورت‌ها نیز گسترش داد و بدین منظور می‌نویسیم $B \uparrow N$ و تک تک پورت‌ها را به آن اضافه می‌کنیم.

تعریف ۳۰. دو آتوماتای B_1, B_2 دو آتوماتای بوخی از رکوردها تحت مجموعه الفبای $\text{Rec}_{\mathcal{N}_1}(D)$ و $\text{Rec}_{\mathcal{N}_2}(D)$ ابتدا باید هر یک از آتوماتاها را با رجوع به مجموعه اسمی پورت‌های دیگری گسترش داده و سپس با استفاده از ضرب استاندارد آتوماتاها آنها را ضرب کرده:

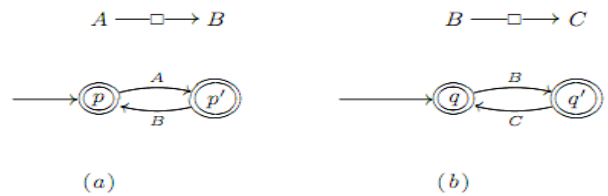
$$(B_1 \uparrow \mathcal{N}_2) \times_B (B_2 \uparrow \mathcal{N}_1) = B_1 \bowtie_B B_2 \quad (34)$$

در تصویر بالا در قسمت e ضرب استاندارد دو آتوماتای d و c می‌باشد که در نهایت صفتی با ظرفیت دو ایجاد می‌شود [۲۷].

تعریف ۳۱. برای مخفی کردن یک پورت $A \in \mathcal{N}$ در آتوماتای بوخی از رکوردها $B = \langle Q, \text{Rec}_{\mathcal{N}}(D), \Delta, Q_0, F \rangle$ آتوماتای $B \downarrow_A = \langle Q, \text{Rec}_{\mathcal{N} \setminus \{A\}}(D), \Delta', Q_0, F \rangle$ نتیجه می‌شود که در رابطه‌ی گذار در صورتی که از حالت q به p گذاری با رکورد r باشد در این آتوماتای بوخی نیز گذاری با حذف پورت A موجود در r از حالت q به p وجود دارند.

در عمل مخفی سازی دامنه یک گذار ممکن است تمام پورت‌هایی باشد که قرار است مخفی شود در این صورت پس از مخفی سازی دامنه تهی می‌شود و این گذار τ بوده که رفتار داخلی سیستم را نشان می‌دهد.

مثال ۸. در مثال قبلی عملیات ضرب بر روی دو آتوماتای بوخی از رکوردها را نشان دادیم در آتوماتای نتیجه پورت B را به عنوان یک پورت میانی می‌توانیم



شکل ۹- (a) صفتی از پورت A به پورت B (b) صفتی از پورت B به پورت C (c) اتصال دو کانال را نشان می‌دهد [۱۴]

لم ۱. در ضرب دو آتوماتای بوخی از رکوردها B_1, B_2 در صورت وجود مجموعه پورت و داده یکسان (الفبای یکسان $\Sigma = \text{Rec}_{\mathcal{N}}(D)$) زبان قابل نمایش (زبانی که پس از حذف رکوردهای τ بدست می‌آید) برای ضرب دو آتوماتا برابر با اشتراک زبان‌های قابل نمایش برای هر یک از آتوماتاها است.

$$L_{\text{vis}}(B_1 \bowtie B_2) = L_{\text{vis}}(B_1) \cap L_{\text{vis}}(B_2) \quad (32)$$

لم ۲. فرض کنید A_1 و A_2 دو آتوماتای محدودیت باشد در این صورت داریم:

$$B(A_1) \bowtie B(A_2) = B(A_1 \bowtie_c A_2) \quad (33)$$

اگر دو آتوماتا محدودیت را ابتدا به آتوماتای بوخی از رکوردها تبدیل کنیم و سپس ضرب دو آتوماتای بوخی را انجام دهیم برابر با ضرب دو آتوماتای محدودیت و سپس تبدیل آن به آتوماتای بوخی از رکوردها است. اثبات این قضیه در [۲۷] آورده شده است.

روش دیگری برای ضرب دو آتوماتای بوخی از رکوردها وجود دارد در این روش ابتدا الفبای دو آتوماتا را یکسان کرده سپس از روش ضرب استاندارد دو آتوماتا که الفبای یکسان دارند عمل می‌کنیم، ابتدا باید روش اضافه کردن اسمی پورت‌های جدید به هر آتوماتای بوخی از رکوردها را بیان کنیم.

تعریف ۲۹. در صورتی که $B = \langle Q, \text{Rec}_{\mathcal{N}}(D), \Delta, Q_0, F \rangle$ آتوماتای بوخی از رکوردها و n یک نام پورت باشد با گسترش آتوماتای بوخی از رکوردهای B با n آتوماتای بوخی از رکوردهای $\langle Q, \text{Rec}_{\mathcal{N} \cup \{n\}}(D), \Delta'', Q_0, F \rangle$ حاصل می‌شود:

- اگر $n \in \mathcal{N}$ باشد $\Delta'' = \Delta$ می‌شود.
- در غیر این صورت از اجتماع سه مجموعه $\Delta'' = \Delta \cup \Delta' \cup \Delta''$ رابطه‌ی گذار بدست می‌آید:

$$\Delta' = \{(q, [n = d], q) \mid q \in Q, d \in D\} \quad 1.$$

$$\Delta'' = \{(q, r[n := d], q') \mid (q, r, q') \in \Delta, d \in D\} \quad 2.$$

برای اضافه کردن یک پورت به آتوماتای بوخی رکوردی B از همان ساختار B استفاده کرده و چند رابطه‌ی گذار دیگر که رفتار آتوماتا را با رجوع به پورت n

- $\text{Rec}_{\mathcal{N}}(\text{Data})$ مجموعه رکوردها بر روی مجموعه داده‌ای Data می‌باشد و به آن الفبای اتوماتا گویند.
- $Q_0 \subseteq Q$ مجموعه حالات ابتدایی است.
- رابطه‌ی گذار بدین صورت تعریف می‌شود:

$$\Delta' \subseteq (Q \times \text{Rec}_{\mathcal{N}}(D) \times \text{Distr}(Q)) \quad (35)$$

$\text{Distr}(Q)$ مجموعه‌ای از توزیع‌ها برای مجموعه حالات می‌باشد و با تابع $\pi: Q \rightarrow [0,1]$ نشان داده می‌شود، به ازای هر حالت و هر رکورد مشخص در هر توزیع احتمال منتخب، مجموع احتمالات خروج از آن رکورد برابر با صفر یا یک می‌باشد:

$$\forall p \in Q, r \in \text{Rec}_{\mathcal{N}}(D) \Delta'(p \times r \times \text{Distr}(Q)) \quad (36)$$

$$: \sum_{q \in Q} \pi(q) = 0 \text{ یا } 1$$

- $F \subseteq Q$ مجموعه حالات پایانی (پذیرنده).

رشته بی‌انتهای $s = r_0 r_1 r_2 \dots$ به شرطی پذیرفته می‌شود که حداقل یک مسیر $q_0 r_1 q_1 r_2 q_2 r_3 \dots$ وجود داشته باشد که در آن $q_0 \in Q_0$ و همچنین احتمال گذار برای هر تابع گذار $\Delta(q_i, r_i, q_{i+1}) > 0$ باشد و اشتراک مجموعه حالات پایانی با مجموعه حالاتی که نامتناهی بار $\inf(\pi)$ در مسیر پذیرش رشته مشاهده می‌شود تهی نباشد.

نکته: یک اتوماتای بوخی را می‌توان با انتساب احتمال یک برای هر حالت بعدی به اتوماتای بوخی احتمالاتی از رکوردهای ساده تبدیل کرد.

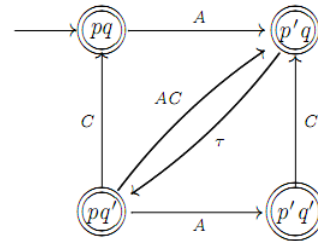
تعریف ۲۳. تبدیل اتوماتای محدودیت احتمالاتی ساده $\mathcal{P} = (Q, \mathcal{N}, \rightarrow, Q_0)$ به اتوماتای بوخی احتمالاتی ساده از رکوردها $B = (Q', \text{Rec}_{\mathcal{N}}(D), \Delta', Q'_0, F')$ رابطه‌ی گذار در اتوماتای محدودیت احتمالاتی ساده صورت $\rightarrow \subseteq Q \times \{N\} \times DC(N) \times \text{Distr}(Q)$ تعریف شده است. همان‌طور که در تبدیل مدل غیر احتمالاتی ذکر شد $Q'_0 = Q_0, Q' = Q, F' = F$ است. برای هر گذار با یک محدودیت داده‌ای مشخص شده همی انتسابات داده‌ای ممکن آن را بدست آورده، هر یک از این انتسابات داده‌ای معادل با یک رکورد است، ممکن است هر محدودیت داده‌ای معادل چندین رکورد باشد سپس برای همی رکوردهای معادل یک محدودیت داده‌ای گذاری تعریف می‌شود با همان توزیع احتمالی که برای آن گذار در اتوماتای محدودیت مشخص شده است.

$$\Delta = \left\{ (q, r, \pi) \mid \exists q \xrightarrow{(N,g)} \pi, \exists \delta: N \rightarrow D: \delta \models g, \text{dom}(r) = N \text{ and } \forall n \in N: r.n = \delta(n) \right\} \quad (37)$$

مثال ۱۰. اگر در اتوماتای محدودیت با مجموعه نام پورت‌ها \mathcal{N} و مجموعه داده‌ی $D = \{d, d'\}$ گذاری بدین صورت وجود داشته باشد $d_A = d_B, \pi$ $(q, \{A, B\}, d_A = d_B, \pi)$ آنگاه معادل آن دو رابطه‌ی گذار می‌توان در اتوماتای بوخی احتمالاتی بدین صورت $(q, [A = d, B = d], \pi)$ و $(q, [A = d', B = d'], \pi)$ نوشت.

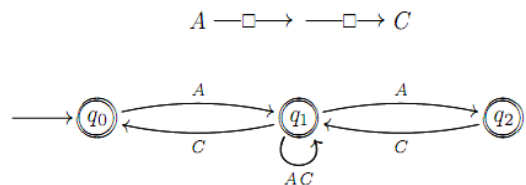
در قضیه زیر اثبات می‌کنیم رشته‌های داده‌ای زماندار که توسط اتوماتای محدودیت احتمالاتی ارائه شده از رشته‌های رکوردها که توسط اتوماتای بوخی احتمالاتی از رکوردها ارائه شده متفاوت نیست، نسخه غیر احتمالاتی این قضیه برای اتوماتای بوخی از رکوردها غیر احتمالاتی و اتوماتای محدودیت قبلا در [۲۱] ارائه شده است.

مخفی کنیم پس از مخفی سازی پورت B اتوماتای زیر بدست می‌آید و گذار $p'q$ به $p'q$ برچسب رکوردش تهی می‌شود و با برچسب τ نشان داده شده است.



شکل ۱۱- اتوماتای بوخی از رکوردهای نتیجه ضرب پس از مخفی سازی پورت [۱۴]

مثال ۹. در ادامه اتوماتای بوخی از رکوردها تعمیم یافته نتیجه بالا را به اتوماتای بوخی بدون گذار τ تبدیل کرده که در تصویر زیر می‌بینیم. در این مدل همان‌گونه که نشان داده شده است حالت q_1 حالتی را نشان می‌دهد که در بافر صف دو ظرفیتی FIFO یک داده قرار گرفته باشد و حالت q_2 حالتی است که هر دو ظرفیت بافر تکمیل باشد و دو داده در این صف باشد.



شکل ۱۲- نتیجه حذف گذاری با برچسب τ

برای مطالعه جزئیات اثبات کلیه قواعد بیان شده در بخش ۳ به [۱۴، ۲۱، ۲۷، ۲۸] رجوع شود.

به منظور مدل کردن کانال‌های احتمالاتی لازم است احتمالات را به گذارهای موجود در اتوماتا وارد کنیم. بدین منظور اتوماتای بوخی احتمالاتی از رکوردها را تعریف می‌کنیم، در این اتوماتا تابع گذار و عملیات ممکن در آن‌ها نسبت به مدل غیر احتمالی تغییر می‌کنند، یک اتوماتای بوخی از رکوردها را در دو سطح می‌توان احتمالاتی کرد. ابتدا اتوماتای بوخی احتمالاتی از رکوردهای ساده را بیان می‌کنیم. در سطح ساده احتمال را فقط برای پیکر بندی بعدی در نظر می‌گیریم و در صورت وجود یک حالت و رکوردی مشخص احتمال رفتن به حالت بعدی را مشخص می‌کنیم. در سطح کلی‌تر لازم است توزیع احتمال را بر عملیات ورودی و خروجی توسعه داده تا بتوان بسیاری از فعالیت‌هایی که برای حل مسئله همروندی در الگوریتم‌های توزیع شده وجود دارد و نیاز است کانال‌هایی با مقادیر خروجی تصادفی ایجاد کنند را مدل کرد، بدین منظور اتوماتای بوخی احتمالاتی از رکوردها کلی‌تری را بیان می‌کنیم، به طوری که در کانال‌ها رخداد عملیات ورودی و خروجی قابل مشاهده با احتمالی مشخص شود.

۴- اتوماتای بوخی احتمالاتی ساده از رکوردها

تعریف ۲۲. اتوماتای بوخی احتمالاتی ساده از رکوردها با چندتایی $B = (Q, \text{Rec}_{\mathcal{N}}(D), \Delta', Q_0, F)$ مشخص می‌شود:

- Q مجموعه‌ی حالات است.

احتمال بزرگتر از صفر برای حالت $\pi(q_{i+1}) > 0, q_{i+1}$ در $B(\mathcal{P})$ است. طبق تعریف ۳۳ بر اساس آنچه برای تبدیل اتوماتای محدودیت احتمالاتی ساده به اتوماتای بوخی احتمالاتی ساده از رکوردها گفته شد پس اجرایی به صورت $q_0, (N_0, g_0), q_1, (N_1, g_1), q_2, (N_2, g_2) \dots$ وجود دارد چنان که برای $i \geq 0$ یک انتساب داده‌ای $\delta_i: N_i \rightarrow D$ وجود دارد به طوری که $\delta_i \neq g_i$ و $\forall n \in N_i, n = \delta_i(n)$ بنا بر این $\theta = \Theta(r)$ و $\theta \in \Theta(L_{TDS}(\mathcal{P}))$ است و داریم $\Theta(L(B(\mathcal{P}))) \subseteq L_{TDS}(\mathcal{P})$.

در این قسمت به محاسبه‌ی احتمال رفتار نامتناهی در اتوماتای بوخی از رکوردهای احتمالاتی ساده، که دنباله‌ای از گذارهاست و از یک حالت اولیه شروع می‌شود، می‌پردازیم. در اجرای نامتناهی باید ابتدا مسئله عدم قطعیت انتخاب یک رکورد از مجموعه رکوردهای قابل انتخاب در آن حالت حل شود. از طرف دیگر بدلیل غیر قطعی بودن اتوماتا، به انتخاب احتمال میان توزیع‌های احتمالی ممکن، نیاز است بنابراین یکی از توزیع‌های احتمالی ممکن توسط زمانبند ها به صورت تصادفی انتخاب می‌شود. در نهایت یک زنجیره مارکوف زمان گسسته ایجاد می‌شود. با انتخاب هر توزیع زنجیره‌ی مارکوف ایجاد شده با دنباله (رشته‌ای) متناهی از رکوردها، برچسب‌دهی می‌شود.

تعریف ۳۴. ضرب دو اتوماتای بوخی احتمالاتی از رکوردهای ساده و $S_1 = (Q_1, Rec_{\mathcal{N}_1}(D), \Delta'_1, Q_{01}, F_1)$ و $S_2 = (Q_2, Rec_{\mathcal{N}_2}(D), \Delta'_2, Q_{02}, F_2)$ برابر است با:

$$S_1 \bowtie S_2 = (Q_1 \times Q_2, Rec_{\mathcal{N}_1 \cup \mathcal{N}_2}(D), \Delta', Q_{01}, F_1 \times F_2, F_2 \times F_1) \quad (39)$$

رابطه‌ی گذار از ۳ قاعده زیر بدست می‌آید:

- در صورت وجود گذارهای قابل ترکیب در دو اتوماتا:

$$\frac{q \xrightarrow{r_1} \pi_1 \quad p \xrightarrow{r_2} \pi_2 \quad \text{comp}(r_1, r_2)}{(q, p) \xrightarrow{r_1 \cup r_2} \pi_1 * \pi_2} \quad (40)$$

در این صورت احتمال برابر است با $(\pi_1 * \pi_2)((q', p')) = \pi_1(q') \cdot \pi_2(p')$ و در صورتی که رابطه‌ی گذار در یک اتوماتا قابل ترکیب نباشد، رابطه‌ی گذار از دو قاعده ۴۱ و ۴۲ به دست می‌آید:

$$\frac{q \xrightarrow{r_1} \pi_1 \quad \text{dom}(r_1) \cap \mathcal{N}_2 = \emptyset}{(q, p) \xrightarrow{r_1} \pi'_1} \quad (41)$$

- به شرطی که $p = p'$ باشد، احتمال برابر است با $\pi'_1((q', p')) = \pi_1(q')$ و در غیر این صورت $\pi'_1((q', p')) = 0$

$$\frac{p \xrightarrow{r_2} \pi_2 \quad \text{dom}(r_2) \cap \mathcal{N}_1 = \emptyset}{(q, p) \xrightarrow{r_2} \pi'_2} \quad (42)$$

- به شرطی که $q = q'$ باشد، احتمال گذار برابر است با $\pi'_2((q', p')) = \pi_2(p')$ و در غیر این صورت $\pi'_2((q', p')) = 0$
- مجموعه حالات پایانی به صورت $\{F_1 \times Q_2, F_2 \times Q_1\}$ است، پس از ضرب دو اتوماتای بوخی احتمالاتی ساده از رکوردهای نتیجه اتوماتای بوخی تعمیم یافته احتمالاتی ساده از رکوردهای می‌شود.

قضیه ۱. در صورتی که فرض کنیم اتوماتای محدودیت احتمالاتی ساده $\mathcal{P} = (Q, \mathcal{N}, \rightarrow, Q_0)$ در این صورت داریم:

$$\gamma(L_{TDS}(\mathcal{P})) = L(B(\mathcal{P})) \cup \Theta(L(B(\mathcal{P}))) = L_{TDS}(\mathcal{P}) \quad (38)$$

اثبات. ابتدا سمت چپ را اثبات می‌کنیم اگر فرض کنیم جریانی از رکوردها $\Gamma = r_0 r_1 r_2 \dots$ که با احتمالی بزرگتر از صفر در $L(B(\mathcal{P}))$ است. از طرفی $B(\mathcal{P})$ طبق تعریف ۳۳ یک اتوماتای بوخی احتمالاتی ساده از رکوردها که همه‌ی حالات پایانی هستند پس اجرای نامتناهی $q_0, r_1, q_1, r_2, q_2, r_3 \dots$ وجود دارد که از یک حالت اولیه q_0 شروع می‌شود و از طرفی هر $\Delta(q_i, r_i, \pi)$ گذاری با احتمال بزرگتر از صفر برای حالت $\pi(q_{i+1}) > 0, q_{i+1}$ است. از این رو بر اساس تعریف ۳۳ گذاری به صورت $\pi(q_i, N_i, g_i, \pi)$ یا $\pi(q_{i+1}) > 0$ با انتساب داده‌ای $\delta_i: N_i \rightarrow D$ وجود دارد چنانکه $\delta_i \neq g_i$ و $\forall n \in N_i, n = \delta(n)$ برقرار است. در این صورت $q_0, (N_0, g_0), q_1, (N_1, g_1), q_2, (N_2, g_2) \dots$ اجرای نامتناهی در \mathcal{P} است و برای هر دنباله زماندار $\theta \in TDS^{\mathcal{N}}$ که $\theta = \gamma(\theta)$ است آنگاه $N(i) = N_i$ و $\delta(i) \neq g_i \forall i \geq 0$ بنا بر این $\theta \in \gamma(L_{TDS}(A))$ و $L(B(A)) \subseteq \gamma(L_{TDS}(A))$.

اگر فرض کنیم $\Gamma = r_0 r_1 r_2 \dots$ جریانی از رکوردها در $L_{TDS}(\mathcal{P})$ پس $\theta \in L_{TDS}(\mathcal{P})$ داریم به طوری که $r = \gamma(\theta)$ برای هر $k \in N, n \in \theta.N(k)$ داریم $r(k) \cdot n = \theta.\delta(k)_n$ چون که $\theta \in L_{TDS}(\mathcal{P})$ پس در اتوماتای محدودیت \mathcal{P} یک اجرای نامتناهی با احتمالی بزرگتر از صفر به صورت $q_0, (N_0, g_0), q_1, (N_1, g_1), q_2, (N_2, g_2) \dots$ وجود دارد که از یک حالت اولیه q_0 شروع می‌شود به طوری که برای همه $\theta.N(i) = N(i)$ و $\delta_i \neq g_i$ و $i \geq 0$ پس بر اساس تعریف ۳۳ اجرایی با احتمال بزرگتر از صفر $q_0, r_1, q_1, r_2, q_2, r_3 \dots$ در $B(\mathcal{P})$ وجود دارد با انتساب داده‌ای به صورت $\delta_i: N_i \rightarrow D$ برای همه‌ی $i \geq 0$ و $\delta_i \neq g_i$ و $r_i \cdot n = \delta_i(n)$ و $\delta_i \neq g_i$ چونکه در $B(\mathcal{P})$ هر اجرایی که از یک حالت اولیه شروع شود پذیرفته می‌شود $r \in L(B(\mathcal{P}))$ بنا بر این $\gamma(L_{TDS}(\mathcal{P})) \subseteq L(B(\mathcal{P}))$.

اثبات سمت راست: اگر $\theta \in TDS^{\mathcal{N}}$ جریانی از رشته‌های زماندار قابل پذیرش در اتوماتای محدودیت \mathcal{P} باشد $\theta \in L_{TDS}(\mathcal{P})$ ، بنا بر این مسیر پذیرشی به صورت $q_0, (N_0, g_0), q_1, (N_1, g_1), q_2, (N_2, g_2) \dots$ همه‌ی گذارها وجود دارد و $q_0 \in Q_0$ می‌باشد. پس برای همه‌ی $i \geq 0$ گذار $\theta.\delta_i \neq \theta.N(i) = N(i)$ به طوری که $\pi(q_{i+1}) > 0$ با (q_i, N_i, g_i, π) برقرار است. بر اساس تعریف ۳۳ اجرای نامتناهی $q_0, r_1, q_1, r_2, q_2, r_3 \dots$ احتمال بزرگتر از صفر در بوخی اتوماتای احتمالاتی ساده از رکوردها وجود دارد. برای همه‌ی $i \geq 0$ انتساب داده‌ی $\delta_i: N_i \rightarrow D$ چنانکه $\delta_i \neq g_i$ و $\forall n \in N_i, n = \delta_i(n)$ برقرار است. بنا بر این $r = r_0, r_1, r_2 \dots \in L(B(\mathcal{P}))$ و $\theta = \Theta(r)$ بنا بر این $L_{TDS}(\mathcal{P}) \subseteq \Theta(L(B(\mathcal{P})))$.

فرض کنید $\theta \in TDS^{\mathcal{N}}$ چنان که $\theta \in \Theta(L(B(\mathcal{P})))$ پس جریانی از رکوردها $r = r_0, r_1, r_2 \dots \in L(B(\mathcal{P}))$ به طوری که $\theta = \Theta(r)$ و به ازای همه‌ی $k \geq 0$ داریم $\theta.N(k) = \text{dom}(r_k)$ و $\theta.\delta(k)_n = \text{dom}(r_k)$ است. از طرفی چون که $r \in L(B(\mathcal{P}))$ پس یک اجرای نامتناهی با احتمال بزرگتر از صفر به صورت $q_0, r_1, q_1, r_2, q_2, r_3 \dots$ وجود دارد به طوری که $q_0 \in Q_0$ و برای همه‌ی $i \geq 0$ گذارهای $\Delta(q_i, r_i, \pi)$ گذاری با

۵- آتوماتای بوخی احتمالاتی از رکوردها

در سطح بعدی احتمالات را بر عملیات ورودی و خروجی اعمال کرده و رابطه توزیع احتمال تغییر می‌کند. با این مدل می‌توان کانال‌های تصادفی را در ریو را مدل کرد. در ادامه ابتدا آتوماتای بوخی احتمالاتی از رکوردها را تعریف کرده سپس عملیات ممکن بر این آتوماتا را بیان می‌کنیم.

تعریف ۳۶. آتوماتای بوخی احتمالاتی از رکوردها (نه لزوماً ساده) با چندتایی $B = \langle Q, \text{Rec}_{\mathcal{N}}(D), \Delta, Q_0, F \rangle$ مشخص می‌شود:

- Q مجموعه‌ی حالات است.
- $\text{Rec}_{\mathcal{N}}(\text{Data})$ مجموعه رکوردها بر روی مجموعه داده‌ای Data می‌باشد و الفبای آتوماتا می‌باشد.
- $Q_0 \subseteq Q$ مجموعه حالات ابتدایی است.
- $F \subseteq Q$ مجموعه حالات پایانی (پذیرنده).

• رابطه گذار نسبت به آتوماتای بوخی از رکوردها تغییر کرده و به صورت $\Delta \subseteq (Q \times \text{Distr}(\text{Rec}_{\mathcal{N}}(D), Q))$ تعریف می‌شود، می‌توان به ازای هر حالت و دیدن رکوردی مشخص با احتمالی که عددی بین $[0, 1]$ می‌باشد به حالتی دیگر رفت، رابطه‌ی گذار بدین صورت نمایش داده می‌شود $q \rightarrow \Pi$. به دلایل تکنیکی در تعریف ضرب دو آتوماتای بوخی احتمالاتی از رکوردها در این مدل لازم است فرض کنیم در صورتی که در یک حالت دو رابطه‌ی گذار $q \rightarrow \Pi(r_1, q_1) > 0$ و $q \rightarrow \Pi(r_2, q_2) > 0$ داشته باشیم آنگاه باید رابطه‌ی زیر برقرار باشد:

$$\text{dom}(r_1) \cap \mathcal{N}^{\text{src}} = \text{dom}(r_2) \cap \mathcal{N}^{\text{src}} \forall n \in (\text{dom}(r_1) \cap \mathcal{N}^{\text{src}}): r_1.n = r_2.n. \quad (۴۳)$$

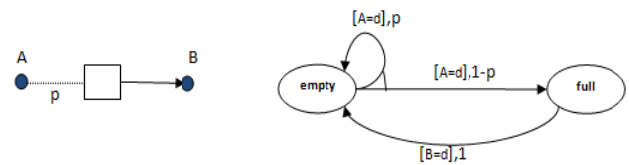
تعریف ۳۷. مخفی کردن پورت $A \in \mathcal{N}$ در آتوماتای بوخی احتمالاتی از رکوردها $B = \langle Q, \text{Rec}_{\mathcal{N}}(D), \Delta, Q_0, F \rangle$ نتیجه برابری است با $B \downarrow_A = \langle Q, \text{Rec}_{\mathcal{N} \setminus \{A\}}(D), \Delta', Q_0, F \rangle$ در هر رکوردی وجود دارد را حذف می‌کنیم و در رابطه‌ی گذار توزیع احتمال یک رکورد $r \setminus A$ برابر است با مجموع تمام توزیع‌های احتمال r به همان حالت مشخص، چون ممکن است با حذف یک پورت و مقدار داده‌ای متناسب به آن در رکوردها برای گذاری به یک حالت مشخص دو رکورد با مقادیر داده‌ای یکسان برای همه پورت‌های فعال در رکورد ایجاد شود بنابراین باید مجموع احتمالات را به دست آوریم. بنابراین در رابطه‌ی گذار $q \rightarrow \Pi'(r \setminus A, p)$ وجود دارد به شرطی که $q \rightarrow \Pi(r, p)$ در آتوماتای اولیه وجود داشته باشد و توزیع احتمال آن از رابطه‌ی زیر بدست می‌آید:

$$\Pi'(r \setminus A, p) = \sum_{r \in \text{Rec}_{\mathcal{N}}(D)} \Pi(r, p) \quad (۴۴)$$

در ادامه عملیات ضرب دو آتوماتای بوخی احتمالاتی از رکوردها را که معنانشناسی برای عملیات پیوند در ریو می‌باشد، را بیان می‌کنیم. در ضرب دو آتوماتای بوخی احتمالاتی از رکوردها پورت‌های مشترک پورتی در B_1 یا B_2 یا در هر دو می‌باشد. ضرب در این حالت پیچیده‌تر از ضرب دو آتوماتای بوخی احتمالاتی از رکوردها است. ممکن است برای کامل کردن گذارها با پورت مشترک لازم شود، چندین گذار با یکدیگر ترکیب شوند. ابتدا حالت ساده‌تر را که الحاق دو آتوماتا می‌باشد را تعریف می‌کنیم، در الحاق دو آتوماتا پورت مشترک، پورت مبدا در آتوماتای دوم و پورت مقصد یا ترکیبی در آتوماتای اول است.

تعریف ۳۵. برای مخفی کردن یک پورت $A \in \mathcal{N}$ در آتوماتای بوخی احتمالاتی از رکوردهای ساده $B \downarrow_A = \langle Q, \text{Rec}_{\mathcal{N} \setminus \{A\}}(D), \Delta', Q_0, F \rangle$ نتیجه می‌شود در آن رابطه‌ی گذار $(q, r \setminus A, \pi) \in \Delta'$ برقرار است اگر و تنها اگر $(q, r, \pi) \in \Delta$ برقرار باشد. مجموعه Q و Q_0 و F تغییر نمی‌کند. در مجموعه رکوردها نیز پورت A از دامنه همه رکوردها حذف می‌شود.

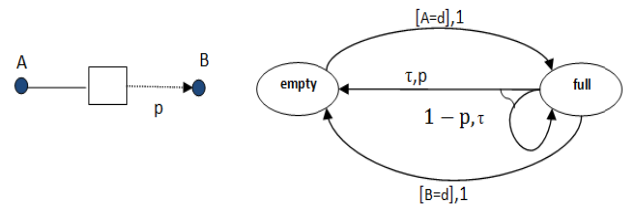
مثال ۱۱. در شکل ۱۳ کانال صفی را مشاهده می‌کنیم که با احتمالی پیام دریافت شده در پورت مبدا در بافر نوشته نشود و پیام گم شود، فرض می‌کنیم مجموعه $\text{Data} = \{d\}$ است. در این مثال با احتمال p داده در بافر نوشته نمی‌شود و با احتمال $1-p$ داده در بافر نوشته می‌شود. برای مدل کردن این صف لازم است دو حالت empty , full در نظر بگیریم مدل آتوماتای بوخی احتمالاتی ساده از رکوردهای معادل با آن در تصویر زیر دیده می‌شود، در این مدل empty و full هر دو حالت اولیه و پایانی هستند.



شکل ۱۳- آتوماتای بوخی احتمالاتی ساده از رکوردها برای صف با احتمال نوشتن پیام در بافر

در صورتی که در مجموعه داده بیشتر از یک عنصر وجود داشت برای هر عنصر مجموعه داده یک حالت در نظر می‌گیریم و احتمال رفتن به هر یک از حالات یکسان می‌باشد، بدین صورت که برای هر عنصر با احتمال p داده در بافر نوشته نمی‌شود، عملیات نوشتن موفقیت آمیز نیست و با احتمال $1-p$ عملیات نوشتن موفقیت آمیز است که متناسب با داده دریافت شده توسط A به حالت مناسب می‌رویم.

مثال ۱۲. می‌توان حالتی دیگر را در نظر گرفت که عملیات نوشتن در بافر با موفقیت انجام می‌شود ولی عملیات خواندن با احتمال p با موفقیت انجام نمی‌شود و داده در بافر گم می‌شود و با احتمال $1-p$ عملیات خواندن با موفقیت انجام می‌شود مدل ریو و مدل بوخی آتوماتای احتمالاتی قطعی از کوردها آن را در شکل ۱۴ می‌بینیم.



شکل ۱۴- مدل کردن کانال صف احتمال گم شدن پیام در بافر وجود دارد

در حالت empty در صورت فعال بودن پورت A با احتمال یک داده در بافر صف نوشته می‌شود و از حالت empty به حالت full می‌رویم در حالت full با احتمال p داده داخل بافر از بین می‌رود و با رکورد به حالت empty باز می‌گردیم و با احتمال $1-p$ داده در بافر می‌ماند و در حالت full منتظر می‌ماند تا زمانی که پورت B فعال شود و داده از بافر خوانده شود و دوباره به حالت empty باز گردیم.

در صورتی که گذاری در B_1 با توزیع احتمال بر روی رکورد Γ_1 وجود داشته باشد آنگاه در آتوماتای الحاقی $B_1; B_2$ گذاری با رکورد τ که دامنه آن زیر مجموعه‌ای از $\mathcal{N}_1 \cup \mathcal{N}_2$ و $\mathcal{N}_1 \cap \mathcal{N}_2 = \text{dom}(\tau)$ است، خواهیم داشت، در این صورت توزیع احتمال برای این گذار با توجه به قابل ترکیب شدن یا نشدن آن به دو صورت بدست می‌آید:

(۱) در صورتی که از حالت q_2 گذاری در B_2 با توزیع احتمال بر Γ_2 وجود داشته باشد به طوریکه $\text{dom}(\tau) = \text{dom}(\Gamma_1) \cap \mathcal{N}_2$ و پورتهایی که در دامنه \mathcal{N}_1 با \mathcal{N}_2 مشترک است برابر با پورتهایی که دامنه Γ_1 با فضای پورت \mathcal{N}_2 مشترک هستند و در دو رکورد پورتهای مشترک دارای مقادیر داده‌ای متناسب یکسان هستند (Γ_1, Γ_2 سازگار هستند) در این صورت توزیع احتمال برای رکورد τ در دو آتوماتای الحاقی برابر با ضرب احتمال هر توزیع در آتوماتا متناظرش است.

(۲) در صورتی که در B_1 گذاری با رکورد مشخص Γ_1 وجود داشته باشد که در دامنه آن اسامی پورتهای مشترک وجود ندارد این گذار قابل ترکیب با هیچ گذاری از B_2 نیست. برای آتوماتای دوم با احتمال یک با رکورد τ در همان حالت q_2 باقی مانده و تغییر حالت فقط برای آتوماتای B_1 داریم و برای حالت‌های دیگر توزیع احتمال صفر می‌شود. توزیع احتمال در آتوماتای نتیجه برای حالت‌هایی که در آتوماتای دوم تغییر حالتی نداریم برابر با احتمال گذار برای آن رکورد در آتوماتای اول است.

و در صورتی که گذاری در B_2 با رکورد Γ_2 داشته باشیم که دامنه این رکورد با اسامی پورت \mathcal{N}_1 اشتراکی ندارد، آنگاه مقدار احتمال در آتوماتای الحاقی در صورتی که تغییر حالت برای آتوماتای اول نداشته $q_1 = p_1$ باشد، برابر با همان توزیع احتمال در آتوماتای B_2 است و برای حالات دیگر نیز احتمال صفر می‌شود.

در ضرب دو آتوماتای بوخی احتمالاتی از رکوردها ممکن است یک گذار از یک آتوماتا با هیچ گذاری از آتوماتای دیگر ترکیب نشود یا ممکن است یک گذار با چندین گذار از آتوماتای دیگری ترکیب بشود. برای تعریف ضرب دو آتوماتای بوخی احتمالاتی از رکوردها ابتدا دو نوع گذار که یکی نشان دهنده عملیات ورودی و خروجی قطعی در یک آتوماتا که معادل با آن گذاری در آتوماتای دوم وجود دارد، دیگری عملیات ورودی مستقل در یک آتوماتا از آتوماتای دیگر است، را تعریف می‌کنیم.

در ادامه حالت کلی‌تر را در نظر می‌گیریم که پورتهای مشترک مبدا، پورتهای B_1 یا B_2 و یا هر دو می‌باشد و ضرب دو آتوماتا را بیان می‌کنیم. در ضرب گاهی یک گذار از یک آتوماتا لازم می‌باشد با چندین گذار از آتوماتای دیگر ترکیب شود و یا در صورتی که پورت مشترک در یکی مبدا و در دیگری مقصد باشد ممکن است به صورت دوره‌ای گذارها ترکیب شوند، ضرب دو آتوماتای بوخی احتمالاتی از رکوردها در واقع ترکیبی از عملیات ممکن برای ضرب دو آتوماتای بوخی احتمالاتی ساده از رکوردها و یا عملیات لازم برای الحاق دو آتوماتا می‌باشد.

۵-۱- عملیات ورودی و خروجی قطعی

برای هر پورت مشترک $A \in \mathcal{N}_1 \cap \mathcal{N}_2$ یک پورت مبدا در B_1 یا B_2 یا در هر دو است، اگر $\Pi_1 \rightarrow q_1$ گذاری در B_1 و دو توزیع $\Pi_1(r_1, p_1) > 0$ و $\Pi_1(r'_1, p'_1) > 0$ وجود داشته باشد در صورت برقراری شرط زیر:

$$\text{dom}(r_1) \cap \mathcal{N}_2 = \text{dom}(r'_1) \cap \mathcal{N}_2 \quad \forall n \in \text{dom}(r_1) \cap \mathcal{N}_2: r_1.n = r'_1.n \quad (51)$$

این گذار را ورودی و خروجی قطعی برای B_2 می‌نامیم.

تعریف ۳۸. در الحاق دو آتوماتای بوخی احتمالاتی از رکوردها $B_1 = (Q_1, \text{Rec}_{\mathcal{N}_1}(D), \Delta, Q_{01}, F_1)$ و $B_2 = (Q_2, \text{Rec}_{\mathcal{N}_2}(D), \Delta_2, Q_{02}, F_2)$ لازم است شرط $\mathcal{N}_1 \cap \mathcal{N}_2 \subset \mathcal{N}_2^{\text{src}}$ برقرار باشد. پورتهای مشترک در دو آتوماتا پورتهای مبدا در آتوماتای دوم است. الحاق دو آتوماتا را بدین صورت نشان می‌دهیم:

$$B_1; B_2 = (Q_1 \times Q_2, \text{Rec}_{\mathcal{N}_1 \cup \mathcal{N}_2}(D), \Delta, Q_{01} \times Q_{02}, \{F_1 \times Q_2, F_2 \times Q_1\}) \quad (45)$$

فرض می‌کنیم $\text{dom}(\tau) \subseteq \mathcal{N}_1 \cup \mathcal{N}_2, q_1, p_1 \in Q_1, q_2, p_2 \in Q_2$ پس از الحاق رابطه‌ی گذار در (q_1, p_2) به صورت $\Pi(r, (p_1, p_2))$ وجود دارد، به طوری که توزیع احتمال برای ترکیب گذارها برابر است با:

$$\Pi(r, (p_1, p_2)) = \Pi_1(r_1, p_1) \times \Pi_2^+(r_2, p_2) \quad (46)$$

در این رابطه رکوردهای r^* و τ_2 به شکل زیر فرض شده است:

$$\text{dom}(r^*) = \text{dom}(r_1) \cap \mathcal{N}_2 \quad \forall n \in \text{dom}(r^*): r^*.n = r_1.n \quad (47)$$

$$\text{dom}(r_2) = \text{dom}(r) \cap \mathcal{N}_2 \quad \forall n \in \text{dom}(r_2): r_2.n = r.n$$

(۱) در رابطه‌ی ۲۳.۳ توزیع $\Pi_2^+(r_2, p_2)$ به شرطی وجود دارد که:

$$q_2 \rightarrow \Pi_2^+(r_2, p_2) > 0 \Rightarrow \text{dom}(r^*) = \text{dom}(r_2) \cap \mathcal{N}_1 \quad \forall n \in \text{dom}(r^*): r^*.n = r_2.n \quad (48)$$

در این صورت این گذار قابل ترکیب با گذار $\Pi_1(r_1, p_1)$ است.

(۲) در صورتی که $\text{dom}(r^*) = \emptyset$ باشد (هیچ پورت مشترکی در دامنه رکورد τ با اسامی پورتهای \mathcal{N}_2 وجود نداشته باشد) آنگاه توزیع $\Pi_2^+(r_2, p_2) = \text{Idle}_{q_2}$ برقرار است. Idle_{q_2} توزیعی است که برای رکورد τ (رکوردی که دامنه آن تهی است) و $p_2 = q_2$ احتمال برابر با ۱ است برای همه حالات و رکوردهای دیگر احتمال صفر است.

به صورت کلی رابطه‌ی گذار برای الحاق دو آتوماتای یا ترکیبی از دو گذار، بر اساس دو شرط ۱ و ۲ در بالا بدست می‌آید:

$$\frac{\Pi(r, (p_1, p_2))}{\Pi(r, (p_1, p_2))} = \frac{\Pi_1(r_1, p_1) \times \Pi_2^+(r_2, p_2)}{\Pi_1(r_1, p_1) \times \Pi_2^+(r_2, p_2)}$$

یا وقتی که گذاری در آتوماتا B_2 موجود است. به طوری که دامنه رکورد موجود در گذار، اشتراکی با اسامی پورتهای B_1 ندارد:

$$q_2 \rightarrow \Pi_2 \wedge \Pi_2(r_2, p_2) > 0 \Rightarrow \text{dom}(r_2) \cap \mathcal{N}_1 = \emptyset \quad \forall n \in \text{dom}(r_2) \cap \mathcal{N}_1: r_2.n = r_1.n \quad (49)$$

در این صورت توزیع احتمال در آتوماتا نتیجه، وقتی که حالت آتوماتای اول تغییر نکند به طوری که $p_1 = q_1$ برابر با توزیع این گذار در آتوماتای دوم است:

$$\Pi_2'(r, (p_1, p_2)) = \Pi_2(r_2, p_2) \quad (50)$$

و در غیر این صورت توزیع احتمال صفر می‌شود $\Pi_2'(r, (p_1, p_2)) = 0$

نکته: اگر پورتهای مبدا در B_2 وجود داشته باشد که در B_1 مقصد یا ترکیبی است، در آتوماتای حاصل از الحاق $B_1; B_2$ ، پورت ترکیبی می‌شود.

روش مشابه تعریف کرد. هر گذار ترکیبی در ضرب دو اتوماتا یک گذار ترکیبی غیر ساده آغاز شده از q_1 یا q_2 می‌باشد و یا یک گذار ترکیبی ساده است.
تعریف ۳۹. ضرب دو اتوماتای بوخی احتمالاتی از رکوردها B_1 و B_2 اتوماتای بوخی احتمالاتی از رکوردها که با چندتایی

$$B_1 \bowtie B_2 = (Q_1 \times Q_2, \text{Rec}_{\mathcal{N}_1 \cup \mathcal{N}_2}, \rightarrow, Q_{0,1} \times Q_{0,2}, \{F_1 \times Q_2, F_2 \times Q_1\}) \quad (۵۶)$$

نشان داده می‌شود و تابع گذار از ترکیب‌های گذار بیان شده در تعاریف بالا (ترکیب‌های گذار ساده و غیر ساده) بدست می‌آید.

شرط پذیرش یک رشته نامتناهی از رکوردها در اتوماتای بوخی احتمالاتی از رکوردها وجود یک مسیر نامتناهی است که رشته را پردازش کند و یکی از حالات پایانی با احتمال بزرگتر از صفر بی‌نهایت بار دیده شود.

نکته ۱. هر اتوماتای بوخی احتمالاتی ساده از رکوردها را می‌توان به یک اتوماتای بوخی احتمالاتی از رکوردها تبدیل کرد. تنها رابطه‌ی گذار را باید تغییر داد و به ازای هر رابطه‌ی گذار $\pi \xrightarrow{r} q$ باید تبدیل به گذار $q \rightarrow \Pi(r, p)$ کرد و $\Pi(r, p)$ برابر با $\pi(p)$ می‌باشد.

نکته ۲. هر اتوماتای بوخی از رکوردها را نیز می‌توان ابتدا به اتوماتای بوخی احتمالاتی ساده از رکوردها تبدیل کرد و سپس آن را به اتوماتای بوخی احتمالاتی تبدیل کرد.

از آنجایی که اتوماتای بوخی احتمالاتی از رکوردها برای بیان سمانتیک کانال‌های احتمالاتی ریو بیان شده است در ادامه چند مثال از کانال‌های ریو احتمالاتی را بیان می‌کنیم.

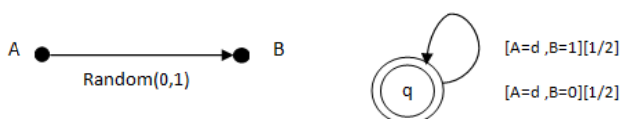
مثال ۱۳. مدل کردن کانالی همگام که با احتمال p داده در پایانه مقصد اشتباه دریافت می‌شود را در شکل ۱۵ ارائه می‌کنیم:



شکل ۱۵- کانال که با احتمال p داده در مقصد اشتباه دریافت می‌شود

رابطه‌ی گذار در این اتوماتا بدین صورت $q \rightarrow \Pi$ نشان داده می‌شود دارای دو توزیع احتمال $\Pi([A = d, B = d], q) = 1 - p$ و $\Pi([A = d, B = d'], q) = p$ است.

مثال ۱۴. در ریو می‌توان کانال‌های همگام با مقدار خروجی تصادفی صفر و یک داشت. در این کانال‌ها در صورتی که پایانه مبدا داده دلخواهی را بنویسد پایانه مقصد همزمان یک مقدار صفر یا یک را می‌پذیرد در این صورت $q \rightarrow \Pi$ تعریف می‌شود و $\Pi([A = d, B = 1], q) = \Pi([A = d, B = 0], q) = 1/2$ وجود دارد. در شکل ۱۶ این کانال مدل شده، این کانال را با استفاده از اتوماتای بوخی احتمالاتی از رکوردها ارائه می‌کنیم.



شکل ۱۶- کانال تصادفی و اتوماتای بوخی احتمالاتی از رکوردها

اگر $q_1 \rightarrow_1 \Pi_1$ یک I/O قطعی برای B_2 باشد، گذار مطابق با این گذار در B_2 به صورت $q_2 \rightarrow_2 \Pi_2$ وجود دارد، چنان‌که در دو گذار $\Pi_1(r_1, p_1) > 0$ و $\Pi_2(r_2, p_2) > 0$ را داریم:

$$\text{dom}(r_1) \cap \mathcal{N}_2 = \text{dom}(r_2) \cap \mathcal{N}_1 \text{ and } \forall n \in \text{dom}(r_1) \cap \mathcal{N}_2: r_1 \cdot n = r_2 \cdot n \quad (۵۲)$$

در ضرب دو اتوماتا می‌توان این دو گذار را با یکدیگر ترکیب کرد که آن را ترکیب ساده می‌نامیم و مقدار احتمال برای گذار ترکیبی از ضرب احتمال هر یک بدست می‌آید:

$$\Pi(r, (p_1, p_2)) = \Pi_1(r_1, p_1) \cdot \Pi_2(r_2, p_2) \quad (۵۳)$$

$$r = r_1 \cup r_2, r_1 \in \text{Rec}_{\mathcal{N}_1}, r_2 \in \text{Rec}_{\mathcal{N}_2}$$

اگر $q_1 \rightarrow_1 \Pi_1$ یک ورودی و خروجی قطعی برای B_2 باشد می‌توان گفت $q_2 \rightarrow_2 \Pi_2$ نیز یک ورودی و خروجی قطعی برای B_1 است.

۵-۲- عملیات ورودی مستقل

B_1 و B_2 دو اتوماتای بوخی احتمالاتی از رکوردها هستند. هر پورت مشترک $A \in \mathcal{N}_1 \cap \mathcal{N}_2$ یک پورت مبدا در B_1 یا B_2 یا در هر دو است. فرض می‌کنیم $q_1 \rightarrow_1 \Pi_1$ گذاری در اتوماتای اول است. اگر برای گذار $q_1 \rightarrow_1 \Pi_1$ پورت‌های مبدا موجود در دامنه r_1 اشتراکی با پورت‌های مقصد و ترکیبی در B_2 نداشته باشد $(\text{dom}(r_1) \cap \mathcal{N}_1^{\text{src}} \cap (\mathcal{N}_2^{\text{snk}} \cup \mathcal{N}_2^{\text{mix}})) = \emptyset$ این گذار را نسبت به B_2 مستقل می‌نامیم.

اگر فرض کنیم $q_1 \rightarrow_1 \Pi_1$ یک ورودی مستقل برای B_2 و $q_2 \in Q_2$ است و همچنین r' را رکوردی تعریف کنیم که دامنه آن زیر مجموعه پورت‌های مبدا که در \mathcal{N}_2 است ولی در مجموعه اسامی پورت‌های \mathcal{N}_1 وجود ندارد، هم‌چنین رکورد r^* را به صورت زیر در نظر بگیریم:

$$\text{dom}(r^*) = \text{dom}(r_1) \cap \mathcal{N}_2 \text{ و } \forall n \in \text{dom}(r^*) r^* \cdot n = r_1 \cdot n \quad (۳۱.۳)$$

گذار قابل ترکیب در B_2 برای گذار $q_1 \rightarrow_1 \Pi_1(r_1, p_1) > 0$ در B_1 برابر است با $q_2 \rightarrow_2 \Pi_2^*(r_2, p_2) > 0$ در صورتی که یکی از دو شرط زیر برقرار باشد:

- $\text{dom}(r_2) \cap \mathcal{N}_2^{\text{src}} = \text{dom}(r_1) \cap \mathcal{N}_2^{\text{src}} \cup \text{dom}(r')$ و $\forall n \in \text{dom}(r'): r' \cdot n = r_2 \cdot n$

$$\text{dom}(r_2) \cap \mathcal{N}_1 = \text{dom}(r^*) \text{ و } \forall m \in \text{dom}(r^*): r^* \cdot m = r_2 \cdot m \quad (۵۴)$$

• اگر $\text{dom}(r') = \text{dom}(r^*) = \emptyset$ در این صورت $\Pi_2^* = \text{Idle}_q$ هر گذار $(q_1, q_2) \rightarrow \Pi$ را یک گذار ترکیبی غیر ساده می‌نامیم که به وسیله $q_1 \in Q_1, p_2 \in Q_2$ و $\text{dom}(r) \subseteq \mathcal{N}_1 \cup \mathcal{N}_2$ آغاز می‌شود. اگر فرض کنیم $\text{dom}(r) \subseteq \mathcal{N}_1 \cup \mathcal{N}_2$ نگاه گذار ترکیبی از قاعده‌ی زیر بدست می‌آید:

$$\Pi(r, (p_1, p_2)) = \Pi_1(r_1, p_1) \cdot \Pi_2^*(r_2, p_2) \quad (۵۵)$$

$$\text{dom}(r^*) = \text{dom}(r_1) \cap \mathcal{N}_2 \text{ و } \forall n \in \text{dom}(r^*) r^* \cdot n = r_1 \cdot n$$

و همچنین دو شرط $\text{dom}(r_1) = \text{dom}(r) \cap \mathcal{N}_1$ و $\text{dom}(r_2) = \text{dom}(r) \cap \mathcal{N}_2$ نیز برقرار است، بنابراین اگر $\text{dom}(r') \neq \emptyset$ یا $q_1 \rightarrow_1 \Pi_1$ ورودی و خروجی قطعی برای اتوماتای دوم نباشد چنین گذار ترکیبی را غیر ساده گویند. برای هر (q_1, q_2) می‌توان گذار ترکیبی غیر ساده که از q_2 آغاز می‌شود به

مثال ۱۷. فرض کنید آتوماتای محدودیت احتمالاتی با مجموعه حالات ابتدایی $Q_0 = \{q\}$ و مجموعه پورت‌های $\mathcal{N} = \{A, B\}$ و مجموعه حالات $Q = \{p, q\}$ و توزیع گذاری به صورت $\Pi \rightarrow q$ با دو گذار:

$$\Pi(\{A, B\}, [d_A = \text{ و } \Pi(\{A, B\}, [d_A = d, d_B = d], p) = 1 - \tau$$

$$\Pi(\{A, B\}, [d_B = *], p) = \tau$$

رکوردها تبدیل کرد به طوری که $F' = \{p, q\}$, $Q'_0 = \{q\}$, $Q' = \{p, q\}$ و توزیع احتمال $\Pi' \rightarrow q$ با دو گذار:

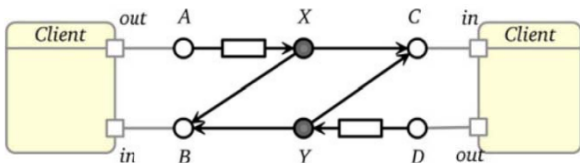
$$\Pi'([A = d, B = *], p) = \tau \quad \text{و} \quad \Pi'([A = d, B = d], p) = 1 - \tau$$

خواهیم داشت.

۶- مثال موردی

در این بخش مدار پیچیده‌تری از ریو با کانال‌های احتمالاتی را با آتوماتای بوخی احتمالاتی از رکوردها مدل می‌کنیم.

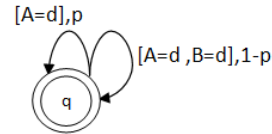
در سیستم نشان داده شده در شکل ۱۹ دو مولفه مشتری پیام‌ها را از طریق رابط ریو مبادله می‌کنند. پیام‌ها از طریق بافرها مبادله می‌شوند. وقتی پیامی در بافری نوشته می‌شود پیام‌ها به صورت همزمان توسط پورت‌های بعد از بافر برای هر دو مولفه فرستاده می‌شود و این تنها زمانی اتفاق می‌افتد که هر دو مولفه آماده پذیرش داده باشند. به عبارت دیگر در صورتی که مولفه‌ای داده‌ای را در پورت in دریافت کرده برابر با داده‌ای که خودش ارسال کرده بود، نشان دهنده این است که داده توسط مولفه‌ی دیگر نیز دریافت شده است. در مدل آتوماتای بوخی از رکوردهای احتمالاتی این سیستم نشان داده شده در شکل ۱۹ حالت‌ها نشان دهنده وجود داده در صف اول یا صف دوم و یا داده بر روی پورت‌های in می‌باشد. در این مدل فرض شده امکان دارد با احتمال p داده هنگام نوشته شدن در بافر صف اول از بین برود و در بافر آن نوشته نشود و با احتمال q داده هنگام نوشته شدن در بافر صف دوم از بین برود و در بافر آن نوشته نشود، حالت x- را حالت پایانی در نظر می‌گیریم چون در صورتی ارتباط دو مولفه موفقیت آمیز است که داده در پورت‌های in دریافت شود.



شکل ۱۹- مدل ریو ارتباط دو مولفه برای تبادل داده [۳۳]

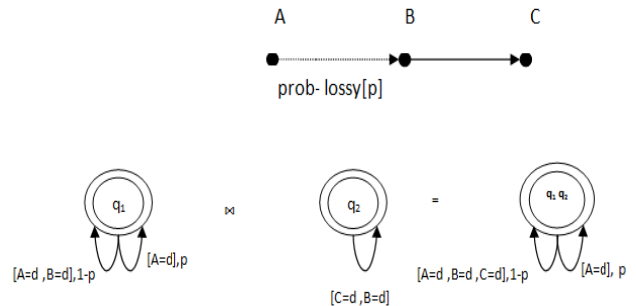
در آتوماتای بوخی احتمالاتی از رکوردها شکل ۲۰ هفت حالت وجود دارد مجموعه اسمی پورت‌های $\mathcal{N} = \{A, D, B, C, Y, X\}$ است و در مجموعه داده دو داده d و d' داریم، در حالت اول --- هر دو صف خالی و پورت in نیز داده‌ای وجود ندارد در صورتیکه مشتری اول داده d را در پورت out خود قرار دهد با احتمال p داده گم می‌شود در بافر صف بالایی نوشته نمی‌شود و پس با احتمال p در همین حالت اول باقی می‌ماند و با احتمال 1-p با گذار $[A=d]$ به حالت x- می‌رود و سپس در این حالت با گذار $[X=d, C=d, B=d]$ داده درون بافر در ورودی هر دو مولفه نوشته می‌شود و به حالت x- می‌رویم و در این حالت پس از مصرف داده توسط دو آتوماتا با گذار τ سیستم به حالت اول --- بر می‌گردد. در صورتی که در حالت اول پورت out از مولفه‌ی دوم داده‌ی d' قرار گیرد با احتمال q داده در بافر صف دوم نوشته نمی‌شود و در حالت اول می‌ماند و در غیر این صورت با احتمال 1-

مثال ۱۵. در دو مثال قبلی مجموعه دامنه دو رکورد در دو توزیع احتمال ثابت بود. مثالی برای یک آتوماتای بوخی احتمالاتی از رکوردها که از یک حالت با رکوردهای مختلف می‌توان توزیع احتمال را تعریف کرد کانال‌های اتلافی همگام‌است، در این کانال با احتمال p داده از بین می‌رود در حالی که با احتمال 1-p داده درست تحویل داده می‌شود. رابطه‌ی گذار در آن به صورت $\Pi \rightarrow q$ و دو توزیع احتمال برای آن به صورت $\Pi([A = d], q) = p$ و $\Pi([A = d, B = d], q) = 1 - p$ می‌توان تعریف کرد. این مدل در شکل ۱۷ ارائه می‌شود.



شکل ۱۷- کانال همگام اتلافی

مثال ۱۶. الحاق دو آتوماتای بوخی از رکوردهای احتمالاتی، آتوماتای اول مدلی از کانال همگام اتلافی و آتوماتای دوم کانال همگام می‌باشد. در این دو کانال پورت مشترک B، پورت مبدا در آتوماتای دوم است. در آتوماتای اول با احتمال p داده از بین می‌رود و به پایانه مقصد تحویل داده نمی‌شود و در آتوماتای دوم با احتمال ۱ داده پس از دریافت از پایانه مبدا به مقصد تحویل داده می‌شود. این مدل‌ها و الحاق آن‌ها در شکل ۱۸ نمایش داده می‌شود.



شکل ۱۸- الحاق دو سیستم گذار رکوردی احتمالاتی

نکته. عملیات پیوند در کانال‌های ریو معادل ضرب دو آتوماتای بوخی احتمالاتی از رکوردها می‌باشد، در ضرب دو آتوماتای بوخی از رکوردها ممکن است یک گذار از یک آتوماتا با هیچ گذاری از آتوماتای دیگر ترکیب نشود یا ممکن است یک گذار با چندین گذار از آتوماتای دیگری ترکیب شود.

تعریف ۴۰. تبدیل آتوماتای محدودیت احتمالاتی $\mathcal{P} = (Q, \mathcal{N}, \rightarrow, Q_0)$ به آتوماتای بوخی احتمالاتی از رکوردها $B = \langle Q', \text{Rec}_{\mathcal{N}}(D), \Delta', Q'_0, F' \rangle$ مانند تبدیل مدل غیر احتمالاتی داریم: $Q'_0 = Q_0$, $Q' = Q$, $F' = Q$ از طرفی رابطه‌ی گذار در آتوماتای محدودیت احتمالاتی به صورت

$\rightarrow \subseteq Q \times \text{Distr}(\cup_{N \in \mathcal{N}} \{N\} \times \text{DA}(N)) \times Q$

تعریف شده است از آنجاییکه در مدل آتوماتای محدودیت احتمالاتی در توزیع‌های احتمال برای هر گذار انتسابات داده‌ای مشخص می‌شود و هر انتساب داده‌ای معادل با یک رکورد است تبدیل گذارها به سادگی انجام شده و به ازای هر توزیع احتمال و گذارهای مربوط به آن رکورد مربوطه را به دست آورده با همان توزیع احتمال قرار می‌دهیم.

$$\Delta = \{q \rightarrow \Pi(r, p) \mid q \rightarrow \Pi(N, \sigma, p) > 0, \text{dom}(r) = N \text{ and } \forall n \in N: r.n = \sigma.n\} \quad (\Delta 7)$$

بوخی احتمالاتی از رکوردها نوعی از فرایند تصمیم مارکوف و مدلی برای بیان سیستم‌های احتمالاتی است. بنابراین در کارهای آتی می‌توان نشان داد برای مساله‌ی تحلیل دسترسی پذیری احتمالی می‌توان از الگوریتم‌های ارائه شده برای فرایند تصمیم مارکوف استفاده کرد. در ادامه می‌توان منطق زمانی (خطی و انشعابی) احتمالاتی تعبیر شده بر مدل‌های بوخی اتوماتای رکوردی که طبیعتاً همانند منطق زمانی خطی احتمالاتی (PLTL) و منطق CTL احتمالاتی (PCTL) را ارائه کرد. همچنین الگوریتم واریسی مدل و پیاده‌سازی ابزار برای مدل‌سازی و واریسی مدل را بیان کرد.

مراجع

[1] F. Arbab, "Reo: A Channel-based Coordination Model for Component Composition," *Journal of Mathematical Structures in Computer Science*, vol. 14, no. 3, pp. 329-366, 2003.

[2] C. Szyperski, D. Gruntz, and S. Murer, *Component Software: Beyond Object-oriented Programming*, Pearson Education, 2002.

[3] F. Arbab "What Do You Mean, Coordination," *Bulletin of the Dutch Association for Theoretical Computer Science*, pp. 1-18, 1998.

[4] G. Papadopoulos, and F. Arbab, "Coordination Models and Languages," *Journal of Computers and Engineering of Large Systems*, vol. 46, no. 4, pp. 329-400, 1998.

[5] F. Arbab, C. Baier, J. Rutten, and M. Sirjani, "Modeling Component Connectors in Reo by Constraint Automata," *Electronic Notes in Theoretical Computer Science*, vol. 97, no. 2, pp. 25-46, 2005.

[6] M. Bonsangue, D. Clarke, and A. Silva, "A Model of Context-dependent Component Connectors," *Science of Computer Programming Journal*, vol. 77, no. 6, pp. 685-706, 2009.

[7] M. Bonsangue, D. Clarke, and A. Silva, "A Model of Context-dependent Component Connectors," *Science of Computer Programming Journal*, vol. 77, no. 6, pp. 685-706, 2012.

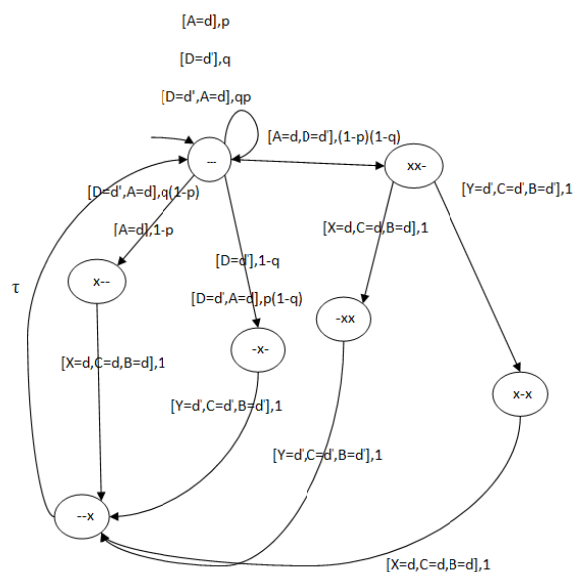
[8] D. Clarke, D. Costa, and F. Arbab, "Connector Colouring I: Synchronisation and Context Dependency," *Science of Computer Programming Journal*, vol. 66, no. 3, pp. 205-225, 2007.

[9] D. Costa, "Formal Model for Component Connector," Ph. D. Dissertation, Vrije University, Amsterdam, Netherlands, 2010.

[10] M. Izadi, M. Bonsangue, and D. Clarke, "Büchi Automata for Modeling Component Connectors," *Journal of Software and Systems Modeling*, vol. 10, no. 2, pp. 183-200, 2011.

[11] C. Baier, "Probabilistic Models for Reo Connector Circuits," *Journal of Universal Computer Science*, vol. 11, no. 2, pp. 1718-1748, 2005.

q داده در بافر صف دوم نوشته می‌شود و به حالت $x-$ می‌رود در این حالت با گذار $[Y=d', C=d', B=d']$ داده از بافر به پورت‌های ورودی دو مولفه انتقال می‌یابد و به حالت $--x$ می‌رود. در صورتیکه در حالت اول هر دو مولفه همزمان داده در بافر صف خود وارد کنند چهار احتمال می‌تواند اتفاق بیافتد، با احتمال q داده در بافر صف دوم نوشته نشود و داده در صف اول هم با احتمال p نوشته نشود پس با احتمال pq در حالت اول باقی می‌مانیم، حالت دیگر ممکن است داده در بافر اول نوشته شود و لی در بافر دوم نوشته نشود پس با احتمال $q(1-p)$ به حالت $x-$ می‌رویم، حالتی دیگر داده در بافر صف دوم نوشته شود ولی داده مولفه اول در بافر صف اول نوشته نشود پس با احتمال $p(1-q)$ به حالت $-x-$ می‌رود و یا امکان دارد داده درون هر دو بافر همزمان نوشته شود و با احتمال $(1-p)(1-q)$ به حالت $xx-$ برویم.



شکل ۲۰- اتوماتای بوخی احتمالاتی از رکوردهای شکل ۱۹

۷- جمع‌بندی و کارهای آتی

در این مقاله مدل اتوماتای بوخی احتمالاتی از رکوردها را به عنوان معناشناسی برای رابط‌های ریو احتمالاتی با رفتار غیر قطعی و تصادفی است ارائه دادیم. در معناشناسی‌های قبلی ارائه شده برای ریو تنها در مدل اتوماتای محدودیت احتمالاتی امکان مدل کردن مدارهای ریو با رفتار تصادفی و غیرقطعی وجود دارد. از طرفی اتوماتای بوخی احتمالاتی از رکوردها نسبت به اتوماتای محدودیت احتمالاتی قدرت بیان بالاتری دارد زیرا در این اتوماتا به علت وجود حالات پایانی می‌توان رابط‌های ریو منصف را مدل کرد در حالی که در اتوماتای محدودیت نمی‌توان مفاهیمی چون انصاف را مدل کرد و همچنین هر اتوماتای محدودیت احتمالاتی قابل تبدیل به اتوماتای بوخی احتمالاتی از رکوردها می‌باشد و در این مدل می‌توان از الگوریتم‌های واریسی موجود برای اتوماتا بوخی نیز استفاده کرد.

بدین منظور اتوماتای بوخی از رکوردها را در دو سطح احتمالاتی کرده در سطح اول اتوماتای بوخی احتمالاتی ساده از رکوردها رایبان کردیم که احتمالات را برای پیکر بندی بعدی در گذارهای موجود لحاظ کردیم و در ادامه در سطح کلی‌تر اتوماتای بوخی احتمالاتی از رکوردها نه لزوماً ساده بیان شد. احتمالات را بر عملیات ورودی و خروجی ممکن اعمال کردیم. در این مدل‌ها عملیات ضرب و مخفی سازی پورت‌ها را که معادل عملیات اتصال کانال‌ها در ریو می‌باشد تعریف کردیم. برای محاسبه احتمال رشته در مسیرهای نامتناهی اتوماتای بوخی احتمالاتی از رکوردها را به فرایند تصمیم مارکوف تبدیل کرده و نشان دادیم یک اتوماتای

- [26] M. K. Reiter, and A. D. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Trans. Information and System Security*, vol. 1, no. 1, pp. 66-92, 1998.
- [27] M. Izadi, M. M. Bonsangue, and D. Clarke, "Modeling Component Connector: Synchronisation and Context-dependency," *Proc. IEEE Intl Conf. Software Engineering and Formal Method*, pp. 303-312, 2008.
- [28] M. Izadi, and M. Bonsangue, "Recasting Constraint Automata into Büchi Automata," *Proc. IEEE Intl Conf. Theoretical Aspects of Computing*, pp.156-170, 2008.
- [29] W. Thomas, *Automata on Infinite Objects*, Handbook of Theoretical Computer Science, MIT Press, 1991.
- [30] CH. Baier, and J. P. Katoen, *Principles of Model Checking*, Handbook of Theoretical Computer Science, MIT Press, 2006.
- [31] O. Kupferman, and M. Vardi, "Verification of Fair Transition Systems," *Journal of Computer Aided Verification*, vol. 11, no. 2, pp. 372-382, 1996.
- [32] M. Vardi, "An Automata-theoretic Approach to Linear Temporal Logic," *Journal of Logics for Concurrency*, vol. 10, no. 4, pp. 238-266, 1996.
- [33] C. Verhoef, C. Krause, O. Kanters, and V. Mei, "Simulation-based Performance Analysis of Channel-based Coordination Models," *Journal of Coordination Models and Languages*, vol. 6, no. 2, pp. 187-201, 2011.
- [12] E. Clarke, O. Grumberg, D. Peled, *Model Checking*, MIT Press, 1999.
- [13] Z. Manna, and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer-Verlag, 1991.
- [14] M. Izadi, *Model Checking of Component Connector*, Ph. D. Dissertation, Leiden University, Leiden, Netherlands, 2011.
- [15] M. Mousavi, M. Sirjani, and F. Arbab, "Formal Semantics and Analysis of Component Connectors in Reo," *Electronic Notes in Theoretical Computer Science*, vol. 154, no. 1, pp. 83-99, 2006.
- [16] F. Arbab, C. Baier, F. De Boer, J. Rutten, and M. Sirjani, "Synthesis of Reo Circuits for Implementation of Component-connector Automata Specifications," *Journal of Coordination Languages and Models*, vol. 34, no. 5, pp. 236-25, 2005.
- [17] F. Arbab, "Abstract Behavior Types: A Foundation Model for Components and their Composition," *Science of Computer Programming Journal*, vol. 28, no. 2, pp. 33-70, 2003.
- [18] N. Diakov, and F. Arbab, "Compositional Construction of Web Services Using Reo," *Proc. IEEE Intl Workshop on Web Services: Modeling, Architecture and Infrastructure*, pp. 47-51, 2004.
- [19] D. Clarke, "Coordination: Reo, Nets, and Logic," *Journal of Formal Methods on Components and Objects*, vol. 5, no. 3, pp. 226-256, 2008.
- [20] F. Arbab, and J. J. M. M. Rutten, "A Coinductive Calculus of Component Connectors," *Lecture Notes in Computer Science*, vol. 2755, no. 5, pp. 35-56, 2003.
- [21] M. Izadi, M. Bonsangue, and D. Clarke, "Büchi Automata for Modeling Component Connectors," *Journal of Software and Systems Modeling*, vol. 10, no. 10, pp. 183-200, 2011.
- [22] A. Rabinovich, "Quantitative Analysis of Probabilistic Lossy Channel Systems," *Proc. IEEE Intl Conf. Automata, Languages and Programming*, pp. 189-189, 2003.
- [23] P. A. Abdulla, N. Bertrand, A. Rabinovich, and P. Schnoebelen, "Verification of Probabilistic Systems with Faulty Communication," *Proc. IEEE Intl Conf. Information and Computation*, pp. 141-165, 2005.
- [24] D. Lehmann, and M. O. Rabin, "On the Advantages of Free Choice: a Symmetric and Fully Distributed Solution to the Dining Philosophers Problem," *Proc. ACM Intl Symp. Principles of Programming Languages*, pp. 133-138, 1981.
- [25] Thrun S, "Probabilistic Algorithms in Robotics," *Artificial Intelligence Magazine*, vol. 21, no. 4, pp. 42-48, 2000.



هاجر عابدی مدرک کارشناسی در رشته مهندسی کامپیوتر (نرم افزار) را از دانشگاه پیام نور زرین شهر، اصفهان، ایران در سال ۱۳۸۹ و مدرک کارشناسی ارشد خود را در مهندسی کامپیوتر (نرم افزار) از دانشگاه صنعتی شریف، تهران، ایران در سال ۱۳۹۲ دریافت کرده است. علاقه مندی‌های تحقیقاتی او شامل نظریه زبان‌ها و ماشین، واری مدل و سیستم‌های توزیع شده است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

habedi@ce.sharif.edu



محمد ایزدی مدارک تحصیلی کارشناسی، کارشناسی ارشد و دکترای مهندسی کامپیوتر خود را از دانشگاه صنعتی شریف، تهران، ایران، دریافت نموده است. وی همچنین دارای مدرک کارشناسی ارشد دیگری در زمینه فلسفه علم از دانشگاه صنعتی شریف و درجه دکترا در علوم کامپیوتر از موسسه علوم کامپیوتر پیشرفته در دانشگاه لیدن، هلند است. از سال ۱۳۸۰ تا ۱۳۹۰ به مدت نزدیک به یازده سال عضو هیات علمی (و برای سه سال مدیر) گروه فلسفه علم پژوهشگاه علوم انسانی و مطالعات فرهنگی (و برای هشت سال) مامور به خدمت به عنوان عضو هیات علمی در گروه فلسفه علم دانشگاه صنعتی شریف بوده است و از مهرماه ۱۳۹۰ تاکنون عضو هیات علمی و استادیار دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف است. زمینه‌ها و علایق پژوهشی وی در حوزه‌های منطق در علوم و مهندسی کامپیوتر، روش‌های صوری در توصیف و

راستی آزمایی سیستم‌ها به ویژه سیستم‌های توزیع شده و هماهنگی و در نهایت نظریه محاسبات و نظریه پیچیدگی است.

آدرس پست الکترونیکی ایشان عبارت است از:

izadi@sharif.edu

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۲/۹/۱۱

تاریخ اصلاح: ۹۳/۶/۱۵

تاریخ قبول شدن: ۹۳/۷/۱۰

نویسنده مرتبط: دکتر محمد ایزدی، دانشکده مهندسی کامپیوتر، دانشگاه

صنعتی شریف، تهران، ایران.

-
- ¹ Component
 - ² Coordination Model
 - ³ Data-Driven
 - ⁴ Control-Driven
 - ⁵ Linda
 - ⁶ Semantics
 - ⁷ Connector Colouring
 - ⁸ Intentional Automata
 - ⁹ Constraint Automata
 - ¹⁰ Probabilistic Constraint Automata
 - ¹¹ Time Data Stream
 - ¹² Glue Code
 - ¹³ Spout Channel
 - ¹⁴ Input/Output
 - ¹⁵ First Input First Output
 - ¹⁶ Shift First Input First Output
 - ¹⁷ Data Assignment
 - ¹⁸ Guard
 - ¹⁹ Data Constraint
 - ²⁰ True
 - ²¹ False
 - ²² Concatenation
 - ²³ I/O Determinism
 - ²⁴ Input-Independence
 - ²⁵ Accept State
 - ²⁶ Visible Language
 - ²⁷ Generalized Büchi Automata

طراحی یک سیستم شناسایی چهره برخط با استفاده از پردازنده‌های گرافیکی

محسن بیگلری^۱ فائزه میرزائی^۱ حسین ابراهیم‌پور کومله^۲

^۱دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شاهرود، شاهرود، ایران
^۲دانشکده مهندسی کامپیوتر، دانشگاه کاشان، کاشان، ایران

چکیده

شناسایی چهره در سه دهه اخیر مورد توجه بسیاری از محققان قرار گرفته است. شناسایی چهره در محیط‌های کنترل شده و در تصاویر ثابت، پیشرفت چشمگیری داشته است. هر چند هنوز در کاربردهای برخط و در محیط‌های کنترل نشده، دارای چالش‌های بسیاری است.

در این مقاله، روشی جدید برای شناسایی چهره ارائه گردیده که تا حدود زیادی مقاوم به تغییرات حالت چهره و روشنایی در تصاویر است. از الگوریتم الگوی دودویی محلی (LBP) برای استخراج ویژگی پایدار به تغییرات یکنواخت روشنایی و حالت چهره و از الگوریتم SQI برای خنثی‌سازی تغییرات نور بهره برده شده است. برای حصول کارایی بالاتر، هر دوی این الگوریتم‌ها توسعه داده شدند. علاوه بر این، برای افزایش هر چه بیشتر سرعت سیستم، از پردازنده‌های گرافیکی (GPU) استفاده گردیده است؛ هر جا که استفاده از GPU مقرون به صرفه نبوده، موزایی‌سازی با CPU صورت گرفته است.

نتایج آزمایشات بر روی دو پایگاه داده YaleB و CMU-PIE نشان از برتری دقت روش پیشنهادی نسبت به سایر روش‌های مورد مقایسه دارد. از این جمله می‌توان به روش‌های SQI، QI، QIR، MQI، DMQI، DSFQI، PCA و LDA اشاره کرد. موزایی‌سازی موجب افزایش تقریبی ۱۹ برابری سرعت سیستم گشته است؛ به شکلی که پردازش یک تصویر با ابعاد 325×205 در زمان ۲۱۳ میلی ثانیه انجام می‌شود.

کلمات کلیدی: تغییرات حالت چهره، تغییرات روشنایی، GPU، الگوریتم موزایی، LBP، SQI.

۱- مقدمه

چالش‌هایی همچون تغییرات روشنایی، حالت و ژست، موجب کاهش کارایی قابل توجه آن‌ها می‌شود.

روش‌های بسیاری برای پشت سر گذاشتن این چالش‌ها ارائه شده است؛ برای نمونه در [۴-۹] چندین الگوریتم برای شناسایی چهره مقاوم به تغییرات روشنایی مورد مطالعه قرار گرفته است؛ روش‌های QI [۶]، SQI [۷] و MQI [۸] از این جمله بوده‌اند. دو روش SQI^۵ و MQI به دلیل موفقیت در نتایج ارائه شده، پایه و اساس بسیاری از روش‌های جدیدتر قرار گرفته‌اند. روش‌های DMQI [۱۰]، DSFQI [۴] نمونه‌هایی از این موارد می‌باشند. مراجع [۱۱، ۱۲] روش‌هایی برای غلبه بر مشکلات حاصل از تغییرات ژست ارائه کرده‌اند. تغییرات ژست از مواردی است که در محیط‌های کنترل نشده بسیار معمول است. به صورت کلی، روش‌های جزئی‌نگر^۶ مانند LBP [۱۳] و EBGm [۱۴] نسبت به روش‌های کلی‌نگر^۷ مانند

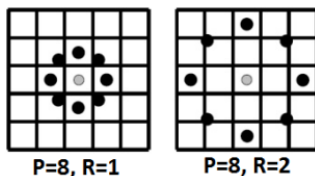
شناسایی چهره در سه دهه اخیر، در مرکز توجه بسیاری از محققان قرار گرفته است. در حال حاضر، سیستم‌های شناسایی چهره در محیط‌های کنترل شده، اغلب عملکرد مناسبی از خود نشان می‌دهند. اغلب این سیستم‌ها با تصاویر ثابت کار می‌کنند. قابلیت پردازش تصاویر متحرک، نیاز به سرعت پردازش برخط بالایی دارد. برخی از چالش‌های اصلی موجود در مسیر سیستم‌های شناسایی چهره عبارتند از: تغییرات حالت^۱ چهره، تغییرات روشنایی^۲، تغییرات ژست^۳ و انسداد^۴ بخشی از چهره.

برای نمونه می‌توان به الگوریتم‌های PCA [۱]، LDA [۲] و ICA [۳] اشاره کرد که در شرایط کنترل شده کارایی مناسبی دارند؛ اما در محیط‌های واقعی،

به دلیل ویژگی‌های ساختاری چهره انسان، عملگر پایه به شکلی توسعه داده شد که بتواند از همسایه‌های دایره‌ای شکل نیز بهره برد [۲۰]. نسخه‌ی توسعه یافته از تعداد نقاط نمونه‌برداری متفاوتی بر روی یک دایره به شعاع دلخواه استفاده می‌کند. نماد $LBP_{P,R}$ برای توصیف این عملگر با تعداد نقاط نمونه‌برداری P بر روی دایره‌ای به شعاع R اشاره دارد. شکل ۲ چند نمونه متفاوت از این عملگر را نشان می‌دهد. فرمول (۱) طرز کار عملگر LBP توسعه یافته را نشان داده است. در فرمول (۱)، $LBP_{P,R}(x,y)$ مقدار دهمی است که برای هر پیکسل (x,y) محاسبه می‌شود. g_c مقدار خاکستری پیکسل مرکزی و g_i مقدار خاکستری پیکسل همسایه است. روش محاسبه $s(x)$ نیز در فرمول (۲) آورده شده است.

$$LBP_{P,R}(x,y) = \sum_{i=0}^{P-1} s(g_i - g_c) 2^i \quad (1)$$

$$s(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (2)$$



شکل ۲- عملگر LBP توسعه یافته با P و R متفاوت

توسعه‌ی دیگری که بر روی عملگر LBP صورت گرفته، معرفی الگوهای یکنواخت^{۱۲} بوده است [۲۰]. یک الگوی دودویی، یکنواخت نامیده می‌شود اگر حداکثر دارای دو تغییر بیتی از صفر به یک یا برعکس باشد. عملگر توسعه یافته یکنواخت با نماد $LBP_{P,R}^{u2}$ نشان داده می‌شود. در مرجع [۲۰] به صورت تجربی نشان داده شده که کارایی عملگر $LBP_{8,1}^{u2}$ یکنواخت تقریباً برابر با عملگر $LBP_{8,1}$ است در صورتی که تعداد این الگوهای یکنواخت نسبت به تعداد کل الگوها برابر با $\frac{58}{256}$ است که کاهش ۷۷ درصدی را به همراه دارد. در محاسبه‌ی هیستوگرام الگوهای یکنواخت، معمولاً الگوهای غیریکنواخت همگی در یک ستون واحد قرار داده می‌شوند. در نتیجه در مجموع ۵۹ ستون خواهیم داشت.

۳- تصویر خود خارج قسمت

الگوریتم خود خارج قسمت یا SQI برای محاسبه تصویر مستقل از تغییرات روشنایی با استفاده از تنها خود تصویر ارائه شده است [۷]. فرمول (۳) روش کار این الگوریتم را نشان می‌دهد. $I(x,y)$ تصویر ورودی و $S(x,y)$ نسخه‌ی یکنواخت^{۱۴} شده‌ی آن می‌باشد.

$$Q(x,y) = \frac{I(x,y)}{S(x,y)} = \frac{I(x,y)}{F(x,y) * I(x,y)} \quad (3)$$

تابع F هسته‌ی یکنواخت کننده و * عملگر کانولوشن^{۱۵} است. معمولاً از فیلتر گوسین وزن دار برای یکنواخت‌سازی تصویر استفاده می‌شود. $Q(x,y)$ نسخه‌ی مستقل از روشنایی تصویر ورودی است؛ خروجی این روش بر روی یک تصویر نمونه در شکل ۳ نمایش داده شده است.

PCA و LDA عملکرد بهتری دارند [۱۵]. تغییرات حالت چهره، اهمیت ویژه‌ای در یک سیستم شناسایی چهره دارند؛ از این رو، تحقیقات گسترده‌ای در این زمینه نیز صورت گرفته است [۱۶-۱۸]. با وجود همه‌ی این مطالعات، شناسایی چهره در محیط‌های واقعی به دلیل وجود چالش‌های مورد اشاره، هنوز یک موضوع تحقیقاتی باز به شمار می‌رود.

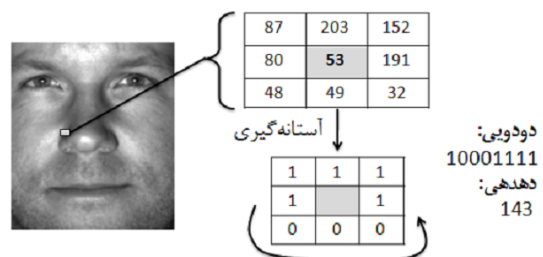
در این مقاله، روشی جدید برای شناسایی چهره پایدار به تغییرات حالت چهره و روشنایی ارائه شده است. این روش بر پایه‌ی دو الگوریتم SQI و LBP^A بنا نهاده شده است. از SQI برای از بین بردن تاثیر تغییرات روشنایی و از LBP برای استخراج ویژگی مقاوم به تغییرات حالت چهره و روشنایی یکنواخت خاکستری استفاده گردیده است. ساختار این دو الگوریتم به شکلی تغییر داده شده که بیشترین کارایی را برای سیستم پیشنهادی داشته باشند. استفاده از چند الگوریتم به صورت ترکیبی، هرچند موجب افزایش کارایی می‌شود ولی کاهش سرعت محسوسی را در پی خواهد داشت. به همین دلیل، با استفاده از پردازنده‌های گرافیکی^{۱۶} و CPU، سیستم پیشنهادی به صورت موازی پیاده‌سازی شده و ساختار مورد استفاده و افزایش سرعت حاصل از آن نیز گزارش گردیده است.

در پایان، دقت سیستم بر روی دو پایگاه داده شناخته شده‌ی YaleB و CMU-PIE مورد آزمایش قرار گرفته و کارایی آن با روش‌هایی چون SQI، QIR [۱۹]، MQI، DMQI، DSFQI، PCA و LDA مقایسه شده است. در ادامه، در بخش دوم و سوم، آشنایی مختصری با الگوریتم‌های LBP و SQI ایجاد شده است. در بخش چهارم، سیستم پیشنهادی ارائه گردیده است. موازی‌سازی سیستم پیشنهادی در بخش پنجم به تفصیل توضیح داده شده است؛ در ابتدای همین بخش، توضیح مختصری در مورد GPU و کودا داده شده است. نتیجه‌ی آزمایشات و نتیجه‌گیری نیز در دو بخش آخر ارائه گشته است.

۲- الگوی دودویی محلی

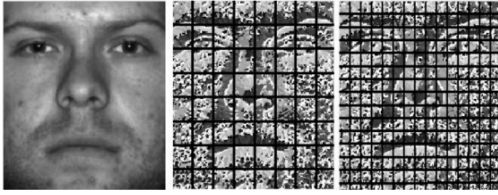
عملگر الگوی دودویی محلی یا LBP توسط أجالا^{۱۱} و سایرین برای توصیف بافت^{۱۱} ارائه شد [۱۳]. توصیف کننده‌های محلی چهره انسان به دلیل پایداریشان نسبت به تغییرات ژست و حالت، توجه زیادی را به خود جلب کرده‌اند. عملگر LBP یکی از بهترین توصیف‌گرهای محلی است. علاوه بر پایداری نسبت به تغییرات ژست و حالت، این عملگر نسبت به تغییرات یکنواخت خاکستری^{۱۲} نیز مقاوم است.

عملگر LBP پایه از یک پنجره 3×3 تشکیل شده است؛ مرکز این پنجره بر روی هر یک از پیکسل‌های تصویر قرار داده شده و اختلاف مقادیر پیکسل‌ها تا مرکز، برای هر یک از همسایه‌ها محاسبه می‌شود. نتیجه به شکل یک رشته‌ی دودویی در آورده شده و سپس به معادل دهمی آن تبدیل می‌شود. در نهایت، یک هیستوگرام به عنوان بردار ویژگی تولید خواهد شد؛ که در آن، هر یک از این مقادیر دهمی، یک ستون را تشکیل می‌دهند. شکل ۱ جزئیات بیشتری را نمایش داده است.



شکل ۱- عملگر LBP پایه

ویژگی‌های استخراج شده دو چندان خواهد شد. بنابراین، دو سطح ویژگی از تصویر ورودی استخراج گشته و سپس به شکل بردار ویژگی‌های نهایی ادغام می‌گردند (شکل ۴).



شکل ۴- استخراج ویژگی با استفاده از عملگر LBP به صورت چندبخشی 8×8 و 15×15



شکل ۳- تصویر مستقل از روشنایی بدست آمده توسط روش SQI

روش MQI [۸] که بعد از این روش ارائه گردیده، از عملگرهای مورفولوژیکی^{۱۶} برای محاسبه تصویر مستقل از روشنایی بهره می‌برد. این روش سرعت بالاتری دارد ولی دقت آن کمی پایین‌تر از روش SQI است.

۴-۳- ساخت بردار ویژگی نهایی و طبقه‌بندی وزن‌دار

برای تشکیل بردار ویژگی‌های نهایی، دو مرحله‌ی زیر به ترتیب انجام می‌شوند:

۱. هیستوگرام‌های استخراج شده از هر بخش، در کنار هم قرار داده می‌شوند.
۲. سپس هیستوگرام‌های بدست آمده از عملگر ۸ بخشی و ۱۵ بخشی (مرحله ۱)، در هم ادغام می‌شوند.

با داشتن ویژگی‌های مناسب، نکته‌ی حائز اهمیت بعدی، یک روش مناسب برای مقایسه‌ی ویژگی‌ها در مرحله‌ی طبقه‌بندی^{۱۷} است. اعمال وزن در مرحله‌ی مقایسه‌ی بردارهای ویژگی صورت می‌گیرد. برای مقایسه‌ی ویژگی‌ها از نسخه‌ی وزن‌دار روش مربع چای^{۱۸} (فرمول (۴)) استفاده شده است.

$$\chi_w^2(H1, H2) = \sum_{i,j} w_j \frac{(H1_{i,j} - H2_{i,j})^2}{(H1_{i,j} + H2_{i,j})} \quad (4)$$

$H1$ و $H2$ هیستوگرام ورودی و هیستوگرام ثبت شده در سیستم هستند. اندیس‌های i و j به ستون i ام از بردار ویژگی مربوط به بخش j اشاره دارند و w_j وزن بخش j ام می‌باشد. بخش‌های مختلف چهره، تاثیر یکسانی در شناسایی چهره ندارند؛ برای پررنگ‌تر کردن تاثیر بخش‌هایی چون بینی، چشم‌ها، دهان و ابروها، از وزن‌های نشان داده شده در شکل ۵ استفاده شده است. در این تصویر، وزن‌های مربوط به هر یک از $8 \times 8 = 64$ بخش که به صورت تجربی بدست آمده‌اند، نمایش داده شده است [۲۱].



شکل ۵- وزن‌های استفاده شده در مقایسه بردارهای ویژگی. رنگ تیره‌تر روشن‌تر به معنی وزن بیشتر است [۲۱]

۵- مقدمه‌ای بر پردازنده‌های گرافیکی و کودا

استفاده از پردازنده‌های چند هسته‌ای تلاشی در جهت رفع مشکل محدودیت زمانی در نرم‌افزارها بوده است [۲۲]؛ اما این پردازنده‌ها قیمت بالایی دارند و حداکثر افزایش راندمان توسط آن‌ها محدود به تعداد هسته‌های آن‌ها است. GPUها که اخیراً توجه بسیاری از محققان را به خود جلب کرده‌اند، ابزاری بسیار

۴- سیستم پیشنهادی

سیستم پیشنهادی از سه لایه تشکیل شده است. در لایه‌ی اول، با استفاده از SQI تاثیر روشنایی بر تصویر خنثی می‌شود. در لایه‌ی دوم، عملگر LBP یکنواخت به صورت چند بخشی به تصویر اعمال شده و ویژگی‌های مورد انتظار را از هر یک از بخش‌ها استخراج می‌کند. در لایه‌ی پایانی، ویژگی‌ها یا در واقع هیستوگرام‌های استخراج شده در کنار هم قرار داده شده و بردار ویژگی نهایی را تشکیل می‌دهند. در ادامه هر یک از لایه‌ها به اختصار توضیح داده شده‌اند.

۴-۱- پیش پردازش

دو فاکتور مهم در روش SQI اندازه پنجره و مقدار انحراف معیار (σ) مربوط به فیلتر گوسین مورد استفاده در آن است. بر اساس آزمایشات صورت گرفته در [۲۱]، اندازه پنجره 15×15 و مقدار انحراف معیار متغیر برای آموزش و آزمایش سیستم انتخاب گردید. در مرحله‌ی آموزش از مقدار انحراف معیار 0.7 و در مرحله‌ی آزمایش از مقدار 0.3 استفاده شد. مقادیر کوچک‌تر انحراف معیار با توجه به فرمول (۳)، تصویر مستقل از روشنایی یکنواخت‌تری را بدست می‌دهند. در نتیجه با استفاده از انحراف معیار بزرگ‌تر در مرحله‌ی آموزش، تصویری با جزئیات بیشتر در سیستم ثبت می‌شود؛ استفاده از تصویر یکنواخت‌تر در مرحله‌ی آزمایش، کارایی را افزایش خواهد داد.

۴-۲- استخراج ویژگی

برای استخراج ویژگی، از عملگر LBP توسعه یافته‌ی یکنواخت به صورت چندبخشی استفاده شده است ($LBP_{g,2}^{m2}$). اعمال این روش به صورت چندبخشی برای شناسایی چهره الزامی است؛ به این ترتیب، ویژگی‌های محلی از هر بخش استخراج می‌گردد. از طرفی تجمیع ویژگی‌های محلی در بردار ویژگی‌های نهایی، ویژگی‌های عمومی را تشکیل می‌دهد. یکی از مزایای دیگر استخراج ویژگی به صورت چندبخشی، امکان وزن‌دهی به بخش‌های مهم‌تر از قبیل چشم‌ها، دهان، بینی و ابروها است.

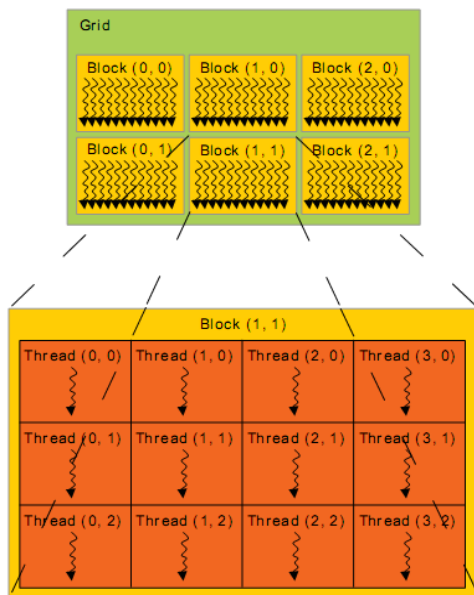
با توجه به تصاویر دو پایگاه داده‌ی مورد آزمایش، تعداد بخش‌های 15×15 انتخاب گردید. کارایی این تعداد بخش در [۲۱] به صورت تجربی اثبات شده است. استفاده از تنها همین تعداد بخش، بیشتر بر روی ویژگی‌های محلی تمرکز می‌کند. برای استخراج ویژگی‌های سطح بالاتر، از تعداد بخش‌های 8×8 نیز به صورت موازی بهره برده‌ایم. به دلیل بخش‌پذیر نبودن این دو مقدار (۱۵ و ۸)، کارایی

OS:	Windows 7
Processor:	Intel Core2 Quad CPU Q9550@2.83 GHz
RAM:	8 GB
Graphics:	GeForce GT 430
CUDA ComputeCapability:	2.1

زمان کل سیستم در حالت ترتیبی برای ابعاد اشاره شده، تقریباً برابر با ۴ ثانیه است؛ در این بخش، با ارائه‌ی ساختاری موازی و پیاده‌سازی آن، این زمان را کاهش داده و سیستم را برای کاربردهای برخط ایده‌آل ساخته‌ایم. در سکوی کودا، تعداد زیادی نخ^{۲۲} اجرایی در قالب مجموعه‌ای از بلوک‌ها در اختیار توسعه دهنده قرار می‌گیرد. برای اجرای یک تابع نوشته شده در کودا، می‌توان تعداد بلوک‌ها و تعداد نخ‌های موجود در هر بلوک را مشخص ساخت. شکل ۸ یک نمونه چیدمان بلوک‌ها و نخ‌های موجود در آن را در دو بعد نشان می‌دهد.

جدول ۱- بخش‌های مختلف سیستم پیشنهادی و زمان اجرای هر یک

مرحله	روش	مدت زمان (s)
۱-۱	پیش‌پردازش (تبدیل رنگی به خاکستری)	0.001
۲-۱	پیش‌پردازش (روش SQI)	3.800
۱	پیش‌پردازش (کل)	3.801
۱-۲	استخراج ویژگی‌ها (۸ LBP بخشی)	0.038
۲-۲	استخراج ویژگی‌ها (۱۵ LBP بخشی)	0.086
۲	استخراج ویژگی‌ها (کل)	0.124
۳	مقایسه ویژگی‌ها	0.016
مجموع	کل سیستم	3.951



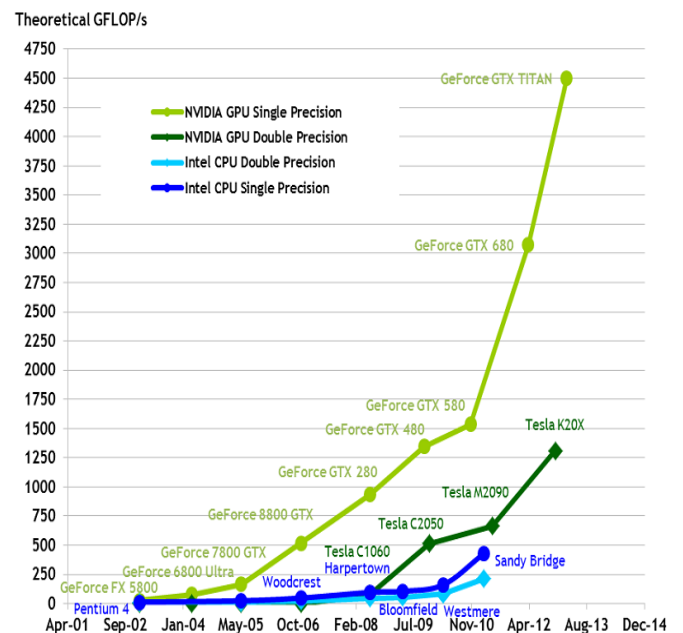
شکل ۸- یک نمونه چیدمان بلوک‌ها و نخ‌های GPU در دو بعد [۲۳]

برای افزایش سرعت سیستم، بخش‌های ۱ و ۳ از جدول ۱ با استفاده از GPU و بخش ۲ با استفاده از CPU موازی‌سازی شد. برای اجرای یک برنامه با استفاده از GPU باید ابتدا داده‌های موردنیاز را به حافظه‌ی GPU منتقل کرد و بعد از اتمام اجرا، داده‌های تولید شده را به حافظه‌ی CPU بازگرداند. بنابراین، در صورتی که زمان جابه‌جایی داده‌ها از سود حاصل از موازی‌سازی برنامه بیشتر باشد، موازی‌سازی به صرفه نخواهد بود. برای مثال موازی‌سازی LBP با GPU به دلیل ساده بودن عملیات انجام شده توسط آن نسبت به حجم داده‌های انتقالی، به صرفه نیست.

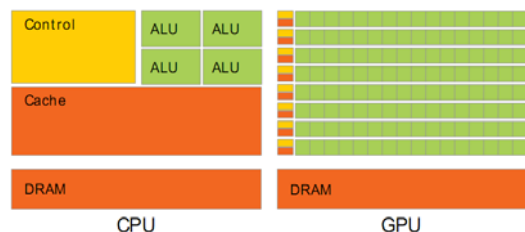
مناسب برای پیاده‌سازی الگوریتم‌های موازی می‌باشند. مزیت GPUها، کارایی بسیار بالا، قیمت مناسب و در دسترس بودن آنها است.

در شکل ۶ مقایسه‌ای بین مدل‌های مختلف CPU و GPU برای انجام عملیات ممیز شناور صورت گرفته است. همان‌طور که در این شکل قابل ملاحظه است، اختلاف عمده‌ای در توانایی انجام عملیات ممیز شناور بین CPU و GPU وجود دارد. علت برتری محسوس GPU، طراحی اختصاصی آن برای انجام عملیات محاسباتی به صورت موازی است؛ به شکلی که تعداد واحدهای پردازشی در GPU نسبت به واحدهای کش^{۱۹} و کنترل جریان^{۲۰} بسیار بیشتر هستند؛ در صورتی که در CPU چنین نیست. شکل ۷ گویای این مطلب است.

شرکت NVIDIA در سال ۲۰۰۶، سکوی کودا^{۲۱} را برای انجام محاسبات موازی حجیم و کارا بر روی GPUهای ساخت خود که در کارت‌های گرافیک این شرکت قرار داشتند، ارائه کرد. با استفاده از کودا، توسعه دهندگان قادر خواهند بود که برنامه‌ی خود را به زبان C نوشته و توسط GPUها اجرا کنند. برای توضیحات بیشتر در مورد این سکوی می‌توانید به [۲۳] مراجعه



شکل ۶- مقایسه‌ی توانایی انجام عملیات ممیز شناور در مدل‌های مختلف CPU و GPU [۲۳]



شکل ۷- مقایسه ساختار داخلی CPU و GPU [۲۳]

۶- موازی‌سازی سیستم پیشنهادی

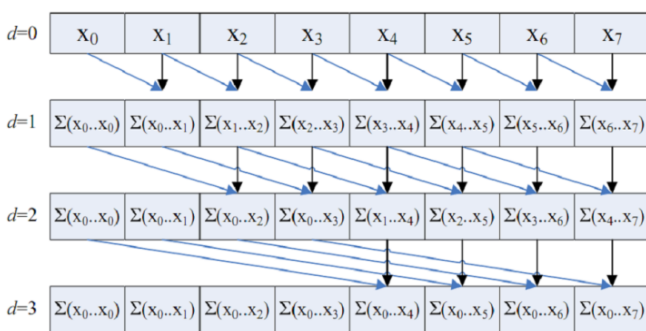
جدول ۱ بخش‌های تشکیل دهنده سیستم شناسایی چهره پیشنهادی و زمان اجرای هر یک را برای تصویری با ابعاد 325×205 نمایش می‌دهد. مشخصات سخت‌افزاری رایانه‌ی مورد استفاده به قرار زیر بوده است:

نگرانی برطرف می‌شود. شکل ۱۱ اجرای موازی این الگوریتم را برای آرایه‌ای به طول ۸ نمایش داده است.

```

for d = 1 to log n
  for all k in parallel
    if k ≥ 2k then
      x[out][k] = x[in][k - 2d-1] + x[in][k];
    else
      x[out][k] = x[in][k];
  swap(in,out)
    
```

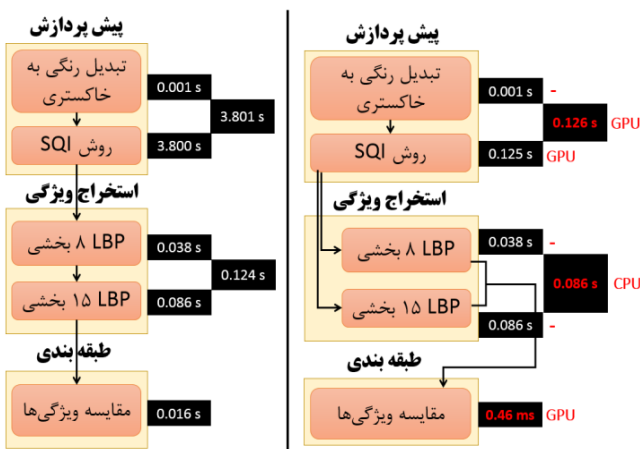
شکل ۱۰- شبه‌کد الگوریتم موازی دوباره برای محاسبه مجموع عناصر بردار



شکل ۱۱- الگوریتم دوباره برای محاسبه مجموع عناصر آرایه‌ای به طول ۸

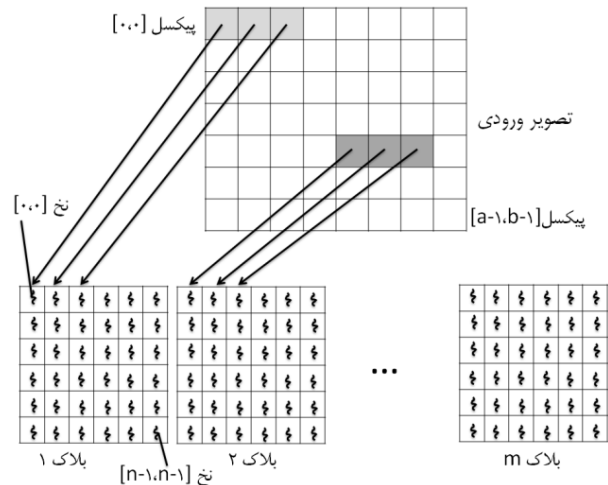
جدول ۲- زمان اجرای بخش‌های تشکیل دهنده سیستم موازی شده

مرحله	روش	مدت زمان (ms)
۱-۱	پیش‌پردازش (تبدیل رنگی به خاکستری)	1
۲-۱	پیش‌پردازش (روش SQI)	125
۱	پیش‌پردازش (کل)	126
۱-۲	استخراج ویژگی‌ها (LBP ۸ بخشی)	38
۲-۲	استخراج ویژگی‌ها (LBP ۱۵ بخشی)	86
۲	استخراج ویژگی (کل)	86
۳	مقایسه ویژگی‌ها	0.462
مجموع	کل سیستم	213.462



شکل ۱۲- خلاصه‌ای از مراحل موازی‌سازی صورت گرفته بر روی بخش‌های مختلف سیستم پیشنهادی

الگوریتم SQI برای محاسبه تصویر خارج قسمت، فیلتر گوسین را به تصویر اعمال می‌کند. برای این منظور، پنجره‌ای با اندازه‌ی 5×5 را بر روی تک‌تک پیکسل‌های تصویر با ابعاد $n \times n$ قرار می‌دهد. در نتیجه، پیچیدگی زمانی آن برابر با $O(s^2 n^2) \in O(n^2)$ خواهد بود. برای موازی‌سازی این الگوریتم، هر نخ را مسئول اعمال پنجره به یک پیکسل قرار می‌دهیم؛ در نتیجه پیچیدگی زمانی به $O(s^2) \in O(1)$ کاهش پیدا می‌کند. برای سپردن هر پیکسل به یک نخ، باید ابتدا نخ‌های تمامی بلوک‌های GPU و تمامی پیکسل‌های تصویر ورودی را به صورت خطی کنار هم قرار داده و سپس یک نگاشت ساده به یک بین آن‌ها برقرار کنیم؛ که البته این امر نیاز به تغییر چیدمان به صورت واقعی ندارد. شکل ۹ یک شمای کلی از توضیحات داده شده را نشان می‌دهد.



شکل ۹- نحوه‌ی توزیع کار بین نخ‌ها و بلوک‌های GPU

عملگر LBP به دو شکل ۸ بخشی و ۱۵ بخشی برای استخراج ویژگی به کار گرفته شد. از آنجا که این دو عملگر کاملاً نسبت به هم مستقل هستند، با استفاده از CPU آن‌ها به صورت موازی نسبت به هم پیاده‌سازی شد. در نتیجه زمان اجرای بخش ۲ از جدول ۱ به زمان عملگر ۱۵ بخشی (عملگر زمانبرتر) کاهش یافت.

مقایسه‌ی بردارهای ویژگی توسط روش مربع چای صورت می‌گیرد. واضح است که پیچیدگی زمانی این روش برابر با $O(n)$ است که n طول بردار ویژگی است. این الگوریتم از دو بخش زیر تشکیل می‌شود:

۱. محاسبه‌ی جمع، تفریق و تقسیم بردارهای ویژگی
۲. جمع نتایج حاصل

در بخش اول، خانه‌های معادل هم از دو بردار ویژگی به صورت ترتیبی مورد پردازش قرار می‌گیرند؛ در صورتی که پردازش هر دو خانه‌ی معادل را به یک نخ در GPU بسپاریم، پیچیدگی زمانی را از $O(n)$ به $O(1)$ کاهش داده‌ایم. البته در این حالت، خروجی الگوریتم یک بردار خواهد بود نه یک مقدار! در نتیجه، محاسبه‌ی مجموع عناصر بردار حاصل، ضروری است. برای محاسبه‌ی مجموع عناصر بردار از الگوریتم موازی دوباره^{۲۳} بهره برده‌ایم. شبه‌کد این روش که دارای پیچیدگی $O(\log n)$ می‌باشد در شکل ۱۰ نمایش داده شده است.

کودا دارای چند نوع حافظه است؛ حافظه‌ی سراسری بزرگترین و در عین حال کندترین آن‌هاست. حافظه‌ی اشتراکی دارای حجم محدود (بسته به نسخه کارت گرافیک) ولی سرعتی ۱۰۰ تا ۱۵۰ برابر حافظه‌ی سراسری است. از آنجا که دو بافر مورد استفاده در این الگوریتم به تعداد $O(n \log n)$ مرتبه خوانده می‌شوند، سرعت خواندن از آن‌ها بسیار با اهمیت است؛ با استفاده از حافظه‌ی اشتراکی، این

است؛ روش‌هایی مانند DSFQI که دقت نزدیکی به سیستم پیشنهادی کسب کرده‌اند، دارای سرعت مناسبی نیستند. در صورتی که سیستم پیشنهادی در کسری از ثانیه عمل می‌کند.



شکل ۱۴- چند نمونه از تصاویر CMU-PIE با روشنایی متفاوت

جدول ۳- مقایسه و آزمون سیستم پیشنهادی بر روی دو پایگاه داده YaleB و CMU-PIE

روش‌ها	دقت شناسایی (%)						
	CMU	YaleB					مجموع
		گروه ۱	گروه ۲	گروه ۳	گروه ۴	گروه ۵	
QI	84	100	98.3	62.5	34.2	23.6	63.72
HE	-	100	95	88.3	50.7	46.8	76.16
PCA	54	-	-	-	-	-	50
LDA	-	-	-	-	-	-	79
QIR	-	100	100	100	90	82.1	94.42
SQI	98.38	100	97.5	100	96.4	97.8	98.34
MQI	98.75	100	98.3	98.3	98.5	97.3	98.48
DMQI	-	100	100	98.3	98.5	97.8	98.92
DSFQI	-	100	99.1	99.1	98.5	98.4	99.02
روش پیشنهادی ۱	98.60	100	98.3	97.5	99.2	98.4	98.68
روش پیشنهادی ۲	99.11	100	100	97.5	100	98.9	99.28

۸- نتیجه‌گیری و کارهای آینده

در این مقاله، سیستمی برخط برای شناسایی چهره پایدار به تغییرات حالت چهره و روشنایی ارائه شد. این سیستم دارای سه بخش پیش‌پردازش، استخراج ویژگی و طبقه‌بندی بود که هر یک به تفصیل توضیح داده شدند. در بخش پیش‌پردازش، با استفاده از الگوریتم SQI تغییرات روشنایی در تصویر خنثی گردید. با استفاده از مقادیر متفاوت انحراف معیار در مرحله آموزش و آزمایش، کارایی سیستم افزایش پیدا کرد. سپس نسخه‌ی توسعه یافته‌ی عملگر LBP به صورت چندبخشی و در تعداد بخش‌های متفاوت برای استخراج ویژگی به تصویر اعمال گردید. در مرحله‌ی مقایسه، روش مربع چای وزن‌دار به کار گرفته شد؛ وزن‌های به کار گرفته شده، به صورت تجربی محاسبه شدند و همان‌طور که در نتایج مشخص است، تاثیر عمده‌ای در بهبود کارایی سیستم داشتند.

جدول ۲ زمان اجرای موازی بخش‌های تشکیل دهنده سیستم را برای یک تصویر با ابعاد 325×205 نمایش می‌دهد. افزایش سرعت حاصل از موازی‌سازی سیستم ۱۸.۵ برابر است که میزان قابل توجهی است. با استفاده از CPU به تنهایی، رسیدن به چنین سرعتی به سادگی امکان پذیر نیست. شکل ۱۲ مراحل موازی‌سازی انجام شده را خلاصه‌وار نمایش می‌دهد.

۷- نتیجه آزمایشات

برای آزمون سیستم پیشنهادی از دو پایگاه داده‌ی شناخته شده‌ی YaleB [۲۴] و CMU-PIE [۲۵] استفاده شده است. روش‌های QI، HE، SQI، QIR، MQI، DSFQI، DMQI، PCA و LDA با روش پیشنهادی، مورد مقایسه قرار گرفته‌اند. پایگاه داده YaleB شامل 10 شخص است که از هر یک ۶۴ تصویر با روشنایی متفاوت موجود می‌باشد. تصاویر به صورت دستی بریده شده و به اندازه‌ی 168×192 در آورده شده‌اند. مجموعه تصاویر بر اساس زاویه‌ی بین منبع نور و دوربین که منجر به سطح روشنایی متفاوتی گشته، به پنج دسته تقسیم شده‌اند (شکل ۱۳). در آزمایشات، یک تصویر از هر شخص که دارای روشنایی معمولی است، به عنوان تصویر آموزشی در نظر گرفته شده و سایر تصاویر مورد آزمون قرار گرفته‌اند.



شکل ۱۳- نمونه تصاویری از پنج گروه با سطح روشنایی متفاوت از YaleB

پایگاه داده CMU-PIE شامل ۶۴ شخص است که از هر شخص تصاویری با تغییرات در ژست، حالت و روشنایی وجود دارد. از آنجا که تمرکز سیستم پیشنهادی بر روی تغییرات روشنایی بوده است، تنها ۲۱ تصویر متعلق به گروه C27 برای هر شخص انتخاب شده‌اند. تصاویر به صورت دستی بریده شده و به اندازه‌ی 150×150 در آورده شده‌اند (شکل ۱۴). در آزمایشات، یک تصویر از هر شخص با روشنایی تقریباً معمولی (f8) به عنوان تصویر آموزشی در نظر گرفته شده و سایر تصاویر مورد آزمون قرار گرفتند. جدول ۳ نتایج آزمایشات صورت گرفته بر روی سیستم پیشنهادی را نمایش می‌دهد.

روش پیشنهادی ۱ در جدول ۳، نسخه‌ای از سیستم پیشنهادی است که از مقایسه‌ی وزن‌دار بهره نمی‌برد. با مقایسه روش پیشنهادی ۱ و ۲ می‌توان به کارایی مقایسه وزن‌دار به خوبی پی برد.

جدول ۳ به روشنی نشان می‌دهد که روش پیشنهادی از سایر روش‌های مقایسه شده عملکرد بهتری داشته است. گروه ۵ در پایگاه YaleB به دلیل تغییرات روشنایی بالای آن، دشوارترین گروه محسوب می‌شود. علت اصلی برتری روش پیشنهادی، اختلاف دقت کسب شده توسط آن بر روی همین گروه بوده است. نتیجه‌ی دیگری که از نتیجه‌ی آزمایشات می‌توان گرفت، کارایی بهتر روش‌های مبتنی بر تصویر خارج قسمت نسبت به سایر روش‌ها از قبیل LDA و PCA است. نکته‌ی مهمی که در این جدول قابل ملاحظه نیست، فاکتور سرعت

[11] X. Chai, S. Shan, X. Chen, and W. Gao, "Locally Linear Regression for Pose-invariant Face Recognition," *IEEE Trans. Image Processing*, vol. 16, no. 4, pp. 1716-1725, 2007.

[12] M. Wai Lee, and S. Ranganath, "Pose-invariant Face Recognition Using a 3D Deformable Model," *Journal of Pattern Recognition*, vol. 36, no. 6, pp. 1835-1846, 2003.

[13] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification based on Featured Distributions," *Journal of Pattern Recognition*, vol. 29, no. 2, pp. 51-59, 1996.

[14] L. Wiskott, J. M. Fellous, N. Kuiger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 775-779, 1997.

[15] X. Zhang, and Y. Gao, "Face Recognition Across Pose: A Review," *Journal of Pattern Recognition*, vol. 42, no. 2, pp. 2876-2896, 2009.

[16] B. Amberg, R. Knothe, and T. Vetter, "Expression Invariant 3D Face Recognition with a Morphable Model," *Proc. IEEE Intl Conf. Automatic Face and Gesture Recognition*, pp. 1-6, 2008.

[17] A. Bronstein, M. Bronstein, and R. Kimmel, "Robust Expression-invariant Face Recognition from Partially Missing Data," *Proc. IEEE Intl Conf. Computer Vision*, pp. 396-408, 2006.

[18] H. S. Lee, and D. Kim, "Expression-invariant Face Recognition by Facial Expression Transformations," *Journal of Pattern Recognition*, vol. 29, no. 2, pp. 1797-1805, 2008.

[19] S. Shan, W. Gao, B. Cao, and D. Zhao, "Illumination Normalization for Robust Face Recognition against Varying Lighting Conditions," *Proc. IEEE Intl Workshop on Analysis and Modeling of Faces and Gestures*, pp. 157-164, 2003.

[20] T. Ojala, M. Pietikäinen, and T. Maenpää, "Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 971-987, 2002.

[21] M. Biglari, F. Mirzaei, and H. Ebrahimpour, "Illumination Invariant Face Recognition Using SQI and Weighted LBP Histogram," *Journal of Electrical Engineering*, vol. 7, no. 4, pp. 47-54, 2013.

[22] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU Computing," *Journal of Electrical Engineering*, vol. 96, no. 6, pp. 879-899, 2008.

[23] CUDA C Programming Guide V5.0, <http://docs.nvidia.com/cuda/cuda-c-programming-guide/>, October 2012.

[24] A. S. Georghiadis, P. N. Belhumeur, and D. J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose,"

در پایان، سیستم نهایی با استفاده از CPU و GPU موازی‌سازی شد؛ به شکلی که سرعت آن تقریباً تا ۱۹ برابر افزایش پیدا کرد. زمان نهایی نسخه‌ی موازی سیستم شناسایی چهره برای پردازش یک تصویر با ابعاد 325×205 به ۲۱۳ میلی ثانیه کاهش پیدا کرد. این زمان برای استفاده از سیستم در یک کاربرد برخط، کاملاً ایده‌آل است. با استفاده از چنین زمان پردازشی می‌توان سیستم را برای پردازش تصاویر متحرک نیز به کار گرفت.

دقت بدست آمده از سیستم و نتایج مقایسه‌ی آن با سایر روش‌ها، قدرت و پایداری روش پیشنهادی را نشان می‌دهد. تکمیل این روش به شکلی که نسبت به فاکتورهای چالش برانگیزی چون تغییرات ژست و انسداد بخشی از چهره نیز پایدار شود، می‌تواند ایده‌ی مناسبی برای کارهای آینده باشد.

مراجع

[1] M. Turk, and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 2, pp. 71-86, 1991.

[2] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face Recognition Using LDA-based Algorithms," *IEEE Trans. Neural Networks*, vol. 14, no. 10, pp. 195-200, 2003.

[3] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face Recognition by Independent Component Analysis," *IEEE Trans. Neural Networks*, vol. 13, no. 3, pp. 1450-1464, 2002.

[4] Y. Cheng, Z. Jin, and C. Hao, "Illumination Normalization based on Different Smoothing Filters Quotient Image," *Proc. IEEE Intl Conf. Intelligent Networks and Intelligent Systems*, pp. 28-31, 2010.

[5] M. Nishiyama, T. Kozakaya, and O. Yamaguchi, "Illumination Normalization Using Quotient Image-based Techniques," *Proc. IEEE Intl Conf. Recent Advances in Face Recognition*, pp. 97-108, 2008.

[6] A. Shashua, and T. Riklin-Raviv, "The Quotient Image: Class-based Re-rendering and Recognition with Varying Illuminations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 129-139, 2001.

[7] H. Wang, S. Z. Li, Y. Wang, and J. Zhang, "Self Quotient Image for Face Recognition," *Proc. IEEE Intl Conf. Image Processing*, pp. 1397-1400, 2004.

[8] Y. Zhang, J. Tian, X. He, and X. Yang, "MQI-based Face Recognition under Uneven Illumination," *Proc. IEEE Intl Conf. Advances in Biometrics*, pp. 290-298, 2007.

[9] L. Zhichao, and E. M. Joo, "Face Recognition under Varying Illumination," *Proc. IEEE Intl Conf. New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems*, pp. 470-476, 2010.

[10] X. G. He, J. Tian, L. F. Wu, Y. Y. Zhang, and X. Yang, "Illumination Normalization with Morphological Quotient Image," *Journal of Software*, vol. 18, no. 1, pp. 2318-2325, 2007.

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۳/۲/۲۹

تاریخ اصلاح: ۹۳/۸/۴

تاریخ قبول شدن: ۹۳/۸/۱۱

نویسنده مرتبط: محسن بیگلری، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شاهرود، شاهرود، ایران.

IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 3, pp. 643-660, 2001.

[25] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) Database," *Proc. IEEE Intl Conf. Automatic Face and Gesture Recognition*, pp. 46-51, 2002.



محسن بیگلری مدرک کارشناسی خود را در گرایش نرم‌افزار از دانشگاه کاشان در سال ۸۹ دریافت کرد. سپس در سال ۹۱ موفق به اخذ مدرک کارشناسی ارشد در گرایش نرم‌افزار از دانشگاه کاشان گردید. در حال حاضر نیز، دانشجوی دکتری دانشگاه صنعتی شاهرود در گرایش هوش مصنوعی است. موضوع پایان‌نامه کارشناسی ارشد او، "تشخیص و شناسایی چهره با استفاده از الگوریتم‌های موازی، توسط کودا" می‌باشد. علایق پژوهشی او برنامه‌نویسی موازی و پردازش تصویر می‌باشند. آدرس پست‌الکترونیکی ایشان عبارت است از:

biglari@shahroodut.ac.ir



فائزه میرزائی مدرک کارشناسی و کارشناسی ارشد خود را در رشته‌ی مهندسی کامپیوتر در گرایش نرم‌افزار از دانشگاه کاشان در سال‌های ۱۳۸۹ و ۱۳۹۱ دریافت کرد. وی هم‌اکنون دانشجوی دکتری هوش مصنوعی در دانشگاه شاهرود می‌باشد. زمینه تحقیقات ایشان پردازش تصویر، بیومتریک و پردازش موازی است. آدرس پست‌الکترونیکی ایشان عبارت است از:

fmirzaei@shahroodut.ac.ir



حسین ابراهیم‌پور کومله دوره کارشناسی خود را در رشته مهندسی کامپیوتر (سخت‌افزار) با رتبه اول در دانشگاه صنعتی اصفهان در سال ۱۳۷۲ و دوره کارشناسی ارشد را با رتبه ممتاز در دانشگاه صنعتی امیرکبیر در گرایش هوش ماشین و رباتیک در سال ۱۳۷۵ به پایان رساند. سپس به عنوان عضو هیات علمی جذب دانشگاه کاشان شد و پس از قبولی در امتحان دکتری بورس و اعزام، از سال ۱۳۷۹ عازم کشور استرالیا گشت. او پروژه دکتری خود را در سال ۲۰۰۴ با عنوان "تکنیک‌های فکتالی برای شناسایی چهره" در دانشگاه صنعتی کوئینزلند با موفقیت به پایان رساند. دکتر ابراهیم‌پور پس از اتمام دوره پسادکتری در دانشگاه نیوکاسل استرالیا و همکاری با مرکز تحقیقات علمی استرالیا (csiro) به عنوان "Visiting Scientist" در سال ۱۳۸۶ به ایران مراجعت و همکاری خود را با دانشگاه‌های کشور از سر گرفته است. در حال حاضر ایشان عضو هیات علمی و معاون پژوهشی دانشکده مهندسی برق و کامپیوتر دانشگاه کاشان، همچنین عضو هیات تحریریه و مدیر داخلی مجله محاسبات نرم و عضو کمیته علمی و داور چندین ژورنال و کنفرانس معتبر داخلی و خارجی هستند و از ایشان تا کنون بیش از ۵۰ مقاله علمی در کنفرانس‌ها، مجلات و کتب معتبر علمی چاپ گشته است. آدرس پست‌الکترونیکی ایشان عبارت است از:

ebrahimpour@kashanu.ac.ir

- ¹ Expression
- ² Illumination
- ³ Pose
- ⁴ Occlusion
- ⁵ Self Quotient Image
- ⁶ Local Approaches
- ⁷ Holistic Approaches
- ⁸ Local Binary Pattern
- ⁹ Graphics Processing Unit (GPU)
- ¹⁰ Ojala
- ¹¹ Texture
- ¹² Monotonic Gray-scale
- ¹³ Uniform
- ¹⁴ Smooth
- ¹⁵ Convolution
- ¹⁶ Morphological Operators
- ¹⁷ Classification
- ¹⁸ Chi Square
- ¹⁹ Cache
- ²⁰ Flow Control
- ²¹ CUDA
- ²² Thread
- ²³ Double Buffer
- ²⁴ Histogram Equalization

معرفی الگوریتم بهبود یافته رقابت استعماری بر اساس رگرسیون چندمتغیره به کمک منطق فازی

محمد فیوضی^۱ سید کمال‌الدین موسوی مشهدی^۱ جواد حدادنیا^۲

^۱دانشکده مهندسی برق، دانشگاه علم و صنعت ایران، تهران، ایران
^۲دانشکده مهندسی برق و کامپیوتر، دانشگاه حکیم سبزواری، سبزوار، ایران

چکیده

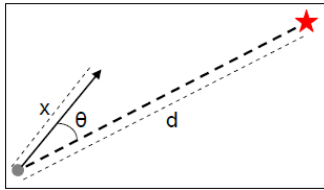
الگوریتم رقابت استعماری^۱ یکی از الگوریتم‌های بهینه‌سازی مبتنی بر جمعیت تصادفی است که از ایده تکامل در فرآیندهای سیاسی-اجتماعی بشر الهام گرفته شده است. در این الگوریتم بر پایه‌ی اطلاعات مربوط به بهترین کارایی قبل (کشورهای مستعمره و استعمارگر)، پارامترهای مسیر حرکت و اصل همگون‌سازی (جذب)، تنظیم می‌شوند. سپس تعداد کشورهای استعمارگر به همراه مستعمرات شان به صورت حرکت یک جمعیت از جواب‌ها، عمل جستجو را در فضای مساله انجام می‌دهند. در این روش هر یک از جواب‌های ممکن (کشورها) سعی می‌کنند به سمت بهترین تجربه بدست آمده (استعمارگر) همگرا شوند، بدنبال این فرایند نقطه بهینه تابع هدف توسط یک جستجوی هوشمند بدست می‌آید. تنظیم پارامترهای الگوریتم استاندارد نقش موثری در همگرایی به پاسخ بهینه در حل مسائل پیچیده با دامنه گسترده را دارد. سعی شد با استفاده از منطق فازی روشی برای کنترل پارامترهای ضریب جذب و هزینه مستعمرات به منظور افزایش دقت، سرعت و صحت الگوریتم رقابت استعماری ارائه شود، سپس توسط رگرسیون چندمتغیره، سیستم فازی در یک رابطه غیرخطی چندجمله‌ای خلاصه می‌شود که البته موجب کاهش حجم حافظه و محاسبه و افزایش سرعت اجرا را در پی دارد. بر این اساس در ابتدا با استفاده از دانش کافی و تسلط بر الگوریتم رقابت استعماری، سیستم فازی طراحی می‌شود، سپس رابطه چندجمله‌ای غیرخطی با استفاده از رگرسیون چندمتغیره استخراج می‌گردد. در هر مرحله از اجرای الگوریتم، موقعیت هر کشور (فضای جواب مساله) با توجه به ورودی‌های کنترلر فازی (نرمالیزه تکرار فرایند و تفاوت نقطه بهینه سراسری در هر تکرار) تعیین می‌شود تا در بدست آوردن موقعیت بعدی کشورها، توسط رابطه چندجمله‌ای غیرخطی، ضریب جذب و هزینه مستعمرات بدست آیند. روش پیشنهادی در حل مسائل معروف (مسائل محک) با الگوریتم‌های استاندارد^۳ ICA، GA^۳، PSO^۴ مقایسه شد و نتایج مقایسه، به منظور اثبات برتری روش پیشنهادی توسط شکل‌ها و جداولی ارائه می‌شود.

کلمات کلیدی: بهینه‌سازی، رقابت استعماری، بهینه سراسری و محلی، منطق فازی، رگرسیون چندمتغیره، جستجوی هوشمند.

۱- مقدمه

تکاملی^۵ (EA) زیادی برای حل مسائل بهینه‌سازی عمومی ارائه شده‌اند [۲، ۳]. این الگوریتم‌ها، با مدل کردن فرآیند تکامل در طبیعت، سعی در یافتن جواب بهینه برای مسئله را دارند [۴]. این موضوع از طریق به تکامل رساندن جمعیتی از راه حل‌های کاندید، مشابه فرآیند بیولوژیکی در تکامل جمعیت انجام می‌شود که می‌تواند با تغییرات محیطی سازگار گردد. الگوریتم‌های تکاملی گوناگونی تاکنون برای بهینه‌سازی ارائه شده‌اند که از آن میان می‌توان به الگوریتم ژنتیک (GA) ارائه شده توسط هالند^۶ در سال ۱۹۶۲

در بسیاری از مسائل بهینه‌سازی بویژه مسائل بزرگ انتخاب بهترین جواب از طریق جستجوی همه جانبه اگر چه غیرممکن نیست ولی کاری بسیار مشکل است. هدف مسائل بهینه‌سازی کاهش زمان جستجو و افزایش دقت و صحت جواب (بهینه تابع هدف) می‌باشد. امروزه با بزرگ شدن مسائل و پیچیده‌تر شدن آنها استقبال از روش‌های ابتکاری بطور چشم‌گیری افزایش یافته است [۱]. تاکنون الگوریتم‌های



شکل ۱- نحوه حرکت کشورها در فضای جستجو بر اساس الگوریتم ICA [۱۷]

در مسائل بهینه‌سازی با در نظر گرفتن تابع $f(x)$ سعی بر آن است تا آرگومان x به گونه‌ای یافته شود که هزینه متناظر با آن، بهینه (معمولاً کمینه) باشد. در یک مسئله بهینه‌سازی Nvar بعدی، یک کشور، یک آرایی $1 \times Nvar$ است. این آرایی به صورت رابطه (۱) تعریف می‌شود.

$$country = [p_1, p_2, p_3, \dots, p_{Nvar}] \quad (1)$$

در الگوریتم ICA برای شروع، تعداد "Ncountry" کشور اولیه ایجاد و "Nimp" از بهترین اعضای این جمعیت (کشورهای دارای کمترین مقدار تابع هزینه رابطه (۲)) به عنوان استعمارگر انتخاب می‌شوند.

$$cost_i = f(country_i) = f(p_1, p_2, p_3, \dots, p_{Nvar}) \quad (2)$$

باقیمانده "Ncol" از کشورها، مستعمراتی را تشکیل می‌دهند که هر کدام به یک امپراطوری تعلق دارند [۴]. استعمارگران با توجه به قدرتش، این مستعمرات را با رابطه (۳)، به سمت خود می‌کشند.

$$T.C_n = Cost(imperialist_n) + \xi \text{mean}\{Cost(colonies\ of\ empire_n)\} \quad (3)$$

قدرت کل هر امپراطوری، با محاسبه قدرت هر دو بخش تشکیل دهنده آن یعنی قدرت کشور استعمارگر (قسمت اول معادله (۳))، به اضافه ξ برابر درصدی از میانگین قدرت مستعمرات آن، تعیین می‌شود. همانطور که در شکل (۱) دیده می‌شود، اگر فاصله بین کشور مستعمره تا استعمارگر برابر d باشد، حرکت کشور مستعمره به اندازه x و به سمت محل استعمارگر نظیر آن خواهد بود. البته این حرکت با زاویه θ منحرف می‌شود که مقدار حرکت x و زاویه θ به طور تصادفی تعیین می‌گردد. معمولاً مقدار زاویه θ به طور یکنواخت در بازه $[-\gamma, \gamma]$ و مقدار حرکت x به طور یکنواخت در بازه $[0, \beta d]$ انجام می‌شود مقادیر γ و β به عنوان پارامترهای الگوریتم ICA هستند که به ترتیب برابر ۴۵ درجه و "۲" پیشنهاد شده است [۱۷].

اگر در طول انجام الگوریتم و حرکت کشورها، یک کشور مستعمره قدرت بیشتری از استعمارگر نظیر خود پیدا کند، جای کشور مستعمره و استعمارگر عوض خواهد شد. به عبارت دیگر در مراحل بعدی اجرای الگوریتم، تمام کشورهای مستعمره استعمارگر قبلی، به استعمارگر جدید تعلق خواهند گرفت و حرکت این مستعمرات به سمت استعمارگر جدید خواهد بود. در هر مرحله از تکرار الگوریتم ICA، رقابتی استعماری میان استعمارگران برقرار است. در این رقابت، استعمارگری که نسبت به دیگر استعمارگران قدرت کمتری دارد، یکی از مستعمرات خود را از دست می‌دهد. اگر استعمارگری به دلیل از دست دادن مستعمرات خود، هیچ مستعمره‌ای نداشته باشد، آن استعمارگر خود به صورت

[۵، ۶، ۷] الگوریتم بهینه‌سازی حرکت دسته جمعی ذرات (PSO) توسط کندی^۷ و ابرهات^۸ در سال ۱۹۹۵ [۸، ۹، ۱۰]، الگوریتم شبیه سازی سرمایه‌های فلزات^۹ (SA) [۱۱-۱۴]، الگوریتم تکامل فرهنگی [۱۵] و سایر الگوریتم‌های موجود در این زمینه اشاره کرد. در سال ۲۰۰۷ الگوریتم جدیدی با نام رقابت استعماری توسط آتشیپ گرگری^{۱۱} و لوکاس^{۱۱} ارائه شد [۱۶] که نه از یک پدیده طبیعی، بلکه از یک پدیده اجتماعی-سیاسی الهام گرفته شده است [۴]. در این الگوریتم، کشورها (مستعمرات) با یک زاویه معین به سمت کشور استعمارگر حرکت می‌کنند تا در نهایت جواب یا نقطه بهینه تابع هدف بدست آید. می‌توان گفت که از جمله مشکلات مهم این الگوریتم احتمال گیر افتادن در دام بهینه محلی و تا حدودی پایین بودن سرعت همگرایی آن است.

در این تحقیق برای اولین بار با استفاده از منطق فازی، کنترل پارامترهای ضریب جذب (β) و هزینه مستعمرات (ξ) را توسط یک تابع غیرخطی هوشمند به منظور افزایش سرعت همگرایی، صحت و دقت الگوریتم استاندارد ICA ارائه کرده‌ایم، این امر موجب بهبود عملکرد الگوریتم ICA در حل مسائل بهینه‌سازی می‌شود. سپس توسط رگرسیون چند متغیره، سیستم فازی را به یک رابطه چندجمله‌ای غیرخطی تبدیل می‌شود. الگوریتم پیشنهادی را بر روی توابع محک آزمایش شده و نتایج را در جداول و شکل‌ها (نمودارها) بخش‌های بعد که همگی نشان دهنده برتری الگوریتم پیشنهادی می‌باشند، بررسی و اثبات می‌شود. در بخش دوم به بررسی الگوریتم استاندارد ICA پرداخته، در بخش سوم منطق فازی، در بخش چهارم ارزیابی الگوریتم پیشنهادی فازی رقابت استعماری، در بخش پنجم رگرسیون چند متغیره، سپس الگوریتم پیشنهادی و در آخر نتایج بدست آمده را بررسی و به اثبات بهبود کارایی الگوریتم پیشنهادی می‌پردازیم.

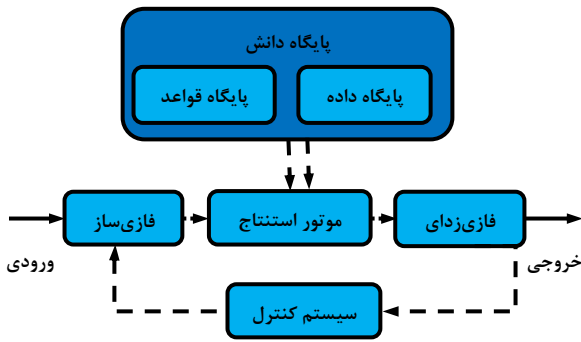
۲- الگوریتم استاندارد رقابت استعماری ICA

برای فهم بهتر الگوریتم ICA تلاش می‌کنیم رابطه‌ای میان الگوریتم GA و ICA برقرار کنیم. در الگوریتم GA تعدادی فرد وجود دارند که یک جمعیت را تشکیل می‌دهند. افراد جمعیت بر اثر عملگرهای ترکیب و جهش در فضای جستجو به دنبال پیدا کردن بهترین جواب جایجا می‌شوند. انتخاب والدین و انتخاب فرزندان جدید برای نسل بعد در الگوریتم GA براساس میزان برآزش هر فرد انجام می‌شود [۴]. در الگوریتم ICA به جای افراد، کشورهایی وجود دارند که هر کشور مشابه یک فرد در الگوریتم GA دارای مشخصاتی است که مکان آن کشور را در فضای جستجو مشخص می‌کند.

در این مجموعه از کشورها که به عنوان نقاطی از فضای جستجو هستند، تعدادی کشور که میزان برآزش بیشتری دارند به عنوان استعمارگر انتخاب می‌شوند. در الگوریتم ICA به جای اصطلاح میزان برآزش از اصطلاح قدرت (Power) استفاده می‌شود. به این ترتیب کشورهای قدرتمند به عنوان استعمارگر و کشورهای ضعیف به عنوان مستعمره قرار می‌گیرند. هر چه قدرت استعمارگر بیشتر باشد، تعداد کشورهای مستعمره بیشتری را به خود اختصاص خواهد داد. در ابتدای الگوریتم، کشورها به طور تصادفی تولید می‌شوند و چند کشور قدرتمند به عنوان استعمارگر انتخاب می‌شوند. سپس سایر کشورها به طور تصادفی به یکی از استعمارگران منتسب می‌شوند، به طوری که تعداد مستعمرات هر استعمارگر متناسب با قدرتش خواهد بود.

حرکت در فضای جستجو به این شکل می‌باشد که هر کشور در راستای کشوری که مستعمره آن است به صورت تصادفی حرکت می‌کند. به عنوان مثال اگر یک استعمارگر را با ستاره و کشور مستعمره آن را با دایره نشان دهیم، حرکت کشور مستعمره به سمت کشور استعمارگر مطابق شکل (۱) خواهد بود [۱۷].

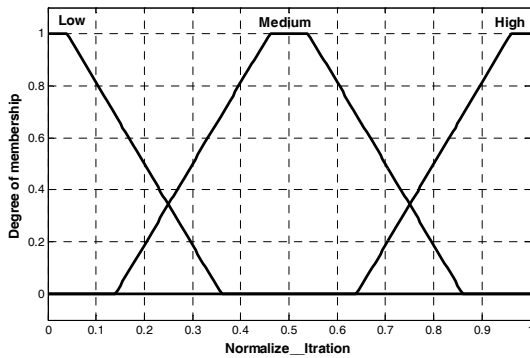
نقاط فضا به درجه عضویت (عددی بین ۰ یا ۱) می‌باشد [۱۹، ۲۰]. هر سیستم فازی تعدادی قانون دارد که بصورت عبارات IF_THEN در بانک قواعد آن ذخیره می‌گردد. مرحله پردازش که به آن موتور استنتاج گفته می‌شود بر اساس بانک قواعد فازی عمل می‌کند. ۲ نوع سیستم استنتاج فازی ممدانی و سوگنو وجود دارد. این ۲ نوع سیستم تا حدودی در روش تعیین فازی ساز ورودی‌ها و عملگرهای فازی شبیه هم هستند. اما در روش تعیین خروجی با هم تفاوت دارند [۲۱]. ما در طراحی سیستم مورد نظر، از سیستم استنتاج ممدانی استفاده کرده‌ایم. تمامی مراحل سیستم فازی در فلوچارت (۲) مشخص شده است.



شکل ۳- تمام مراحل یک سیستم فازی

۴-۱- پیاده‌سازی سیستم فازی

سیستم فازی طراحی شده ۲ ورودی (نرمالیزه تکرار فرآیند و تفاوت نطقه بهینه سراسری) و دو خروجی ضریب جذب و هزینه مستعمرات دارد. توابع عضویت در این سیستم از نوع ذوزنقه‌ای برای هر ورودی و خروجی مانند شکل (۴) می‌باشند.



شکل ۴- تابع عضویت برای سیستم فازی، اینجا تابع ورودی اول

محدوده ورودی اول طبق رابطه (۴) در بازه [۰ ۱] مشخص می‌شود.

$$Normalize_Iteration(i) = \frac{iter(i)}{\max(iter)} \quad (4)$$

محدوده ورودی دوم نیز طبق رابطه (۵) بر اساس شکل (۵) در بازه [0 1] مشخص می‌شود.

$$D_Best = \frac{1}{Npar} (MeanCost - MinimumCost) \quad (5)$$

مستعمره یک استعمارگر دیگر در خواهد آمد. مراحل الگوریتم ICA به همین ترتیب ادامه می‌یابد، تا تعداد استعمارگران به یک برسد. در این حالت تمام کشورها، مستعمره یک استعمارگر هستند و الگوریتم به پایان می‌رسد بدین ترتیب نقطه بهینه تابع هدف مشخص می‌شود، تمام مراحل و پارامترها این الگوریتم در فلوچارت (۱) و جدول (۱) بیان شده است.



شکل ۲- مراحل اجرای الگوریتم پیشنهادی

۳- روش پیشنهادی

مشکل عمده و اساسی الگوریتم رقابت استعماری بیان شد، ابتدا توسط سیستم فازی، کنترل کننده‌ای در قالب یک تابع غیرخطی به‌منظور برآورد و کنترل ضرایب جذب و هزینه مستعمرات طراحی می‌شود، سپس توسط رگرسیون چند متغیره، رابطه یا تابع غیرخطی برگرفته از تابع غیرخطی که نمایانگر رفتار سیستم فازی است، تعریف می‌شود. سپس رابطه غیرخطی در الگوریتم اصلی به جای سیستم فازی جایگذاری می‌شود، با این کار تمام سیستم فازی طراحی شده در یک رابطه چند جمله‌ای خلاصه می‌شود. در انتها به منظور اثبات همگرایی و کارایی الگوریتم پیشنهادی، از توابع محک استفاده می‌شود.

۴- منطق فازی

منطق فازی روش نسبتاً جدیدی است که در مقابل روش‌های مرسوم برای طراحی و مدل سازی سیستم‌هایی که نیازمند ریاضیات و آمار پیچیده هستند به کار می‌روند. منطق فازی اولین بار توسط پروفیسور زاده [۱۸] معرفی شد. 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

دو خروجی سیستم فازی (ضریب جذب و هزینه مستعمرات) می‌باشند. دلایل انتخاب این دو پارامتر بعنوان خروجی‌ها، وابستگی زیاد الگوریتم استاندارد ICA به این پارامترها می‌باشد.

نتیجه می‌رسیم که برای یکسان بودن شانس مستعمره برای همه، باید ضریب جذب در ابتدا زیاد و هزینه مستعمره کم باشد [۱۶]. همچنین برای فرار از نقاط بهینه محلی باید در انتهای الگوریتم ضریب جذب کاهش یابد و نیز برای افزایش سرعت در رسیدن به قله سراسری و نقطه بهینه باید در انتهای الگوریتم ضریب هزینه مستعمره زیاد و در حد استاندارد قرار گیرد [۱۶]. به این منظور برای ایجاد یک رابطه غیرخطی مناسب بین این ورودی‌ها و خروجی‌ها در این مقاله از یک سیستم فازی استفاده می‌شود. قسمت دیگر سیستم فازی بانک قواعد آن می‌باشد که براساس مطالب بیان شده بصورت جدول (۲) طراحی می‌گردد.

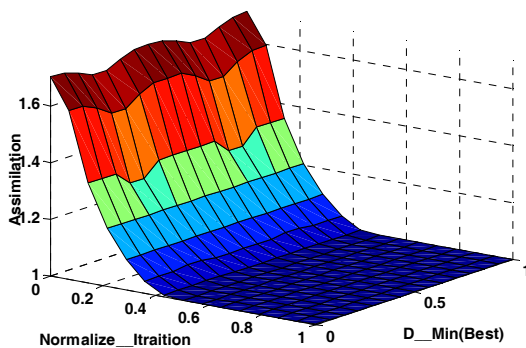
مرحله پردازش که به آن موتور استنتاج گفته می‌شود بر اساس بانک قواعد فازی عمل می‌کند. مشخصات سیستم فازی طراحی شده در جدول (۳) آمده است.

جدول ۳- مشخصات سیستم فازی طراحی شده

استلزام	ممدانی	تعداد متغیرها	2in / 2out
اشتراک	min	تعداد توابع عضویت	3MF
اجتماع	max	نوع توابع عضویت	Trap MF
استنتاج	min	فازی زدا	مرکزقل

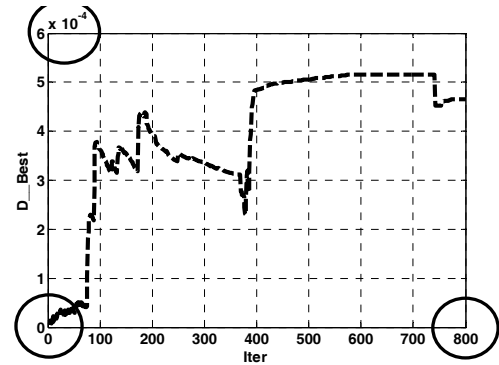
با ورود متغیرهای ورودی، سیستم فازی در داخل برنامه اصلی جایگذاری می‌شود، با توجه به بانک قواعد و روش‌های استنتاج و استلزام، سیستم فازی خروجی‌های مناسب را با توجه به جداول (۲ و ۳) (یعنی β ، γ ، β ، γ) ضرایب جذب و هزینه مستعمرات) به برنامه اصلی پاس می‌دهد. این منجر به بهبود کاوش و افزایش سرعت مشهود در همگرایی الگوریتم نهایی می‌شود، البته (رابطه سطحی بین ورودی‌ها و خروجی‌ها) در شکل (۷) بیان شده است.

پس از اعمال قوانین براساس توابع عضویت‌ها، Surface کنترلی نهایی بدست می‌آید. هر Surface نشان دهنده میزان تغییرات یک خروجی بر اساس دو ورودی است. در این سیستم به ازای هر خروجی یک Surface تعریف می‌شود، دو Surface در این سیستم متفاوت هستند. Surface با توجه به شکل‌های بدست آمده، صحت قوانین و توابع عضویت متغیرهای ورودی و خروجی و نیز نرمی و عملکرد آنها در شکل‌های (۷ و ۸) رسم شده است که به نمایش درآمده است.



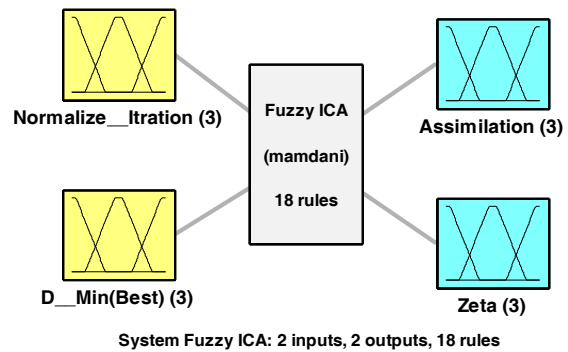
شکل ۷- سطح فیزیکی بین ورودی‌ها و خروجی اول

رابطه سطحی بین ورودی‌ها و خروجی دوم در شکل (۸) رسم شده است.



شکل ۵- محدوده مشخص شده برای ورودی دوم

تمام پارامترها و سیستم فازی طراحی شده در شکل (۶) مشخص شده است.



شکل ۶- سیستم فازی طراحی شده

الگوریتم رقابت استعماری که در [۴] ارائه شده است دارای تنظیمات و پارامترهایی خاص می‌باشد، که ما در این تحقیق از این پارامترها استفاده می‌نماییم. البته تمام مشخصات مذکور در جدول (۱) مشخص شده است.

جدول ۱- مشخصات سیستم فازی

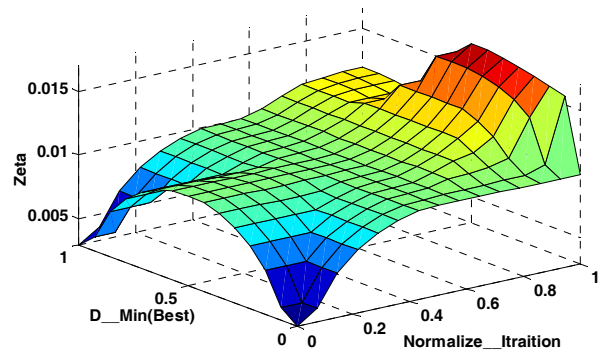
Parametr	Value
Num of Country	500
Num of Initial Imperialists	10
Num of Decades	800
Revolution Rate	0.4
Assimilation Coefficient	Set By Proposed ALG
Assimilation Angle Coefficient	0.05
Zeta	Set By Proposed ALG
Damp Ratio	0.99
Uniting Threshold	0.02

جدول ۲- بانک قواعد سیستم فازی طراحی شده

D_Min(Best)	High	Assimilation: High Zeta: High	Assimilation: Med Zeta: Med	Assimilation: Low Zeta: Med
	Medium	Assimilation: High Zeta: Med	Assimilation: High Zeta: Med	Assimilation: Med Zeta: High
Low	Assimilation: High Zeta: Low	Assimilation: Med Zeta: Med	Assimilation: Low Zeta: Med	
		Low	Medium	High
		Normalize_Itration		

جدول ۴- صورت ریاضی توابع محک [۴]

Function	Relation	Range
Rastregin Or F1	$f(x) = \sum_{i=1}^D (x_i^2 - 10 \times \cos(2\pi x_i) + 10)$	(-10,10)
Rosenbrock Or F2	$f(x) = \sum_{i=2}^D (100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(-100,100)
Sphere Or F3	$f(x) = \sum_{i=1}^D x_i^2$	(-100,100)
Griewank Or F4	$f(x) = \frac{1}{4000} \times \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	(-600,600)



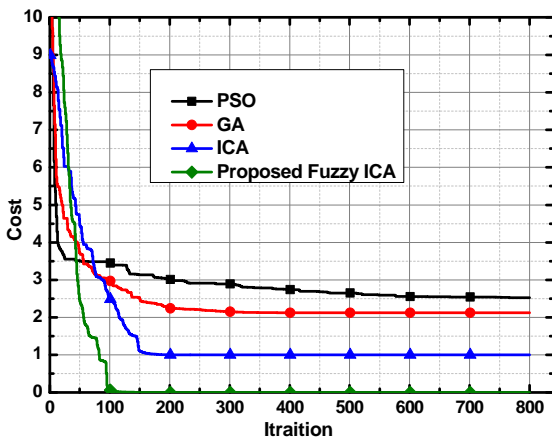
شکل ۸- سطح فیزیکی بیان کننده رابطه بین ورودی‌ها و خروجی دوم

اولین ارزیابی بر روی تابعی ۱۵ بعدی انجام می‌شود که برای اولین بار الگوریتم ICA بر روی آن پیاده‌سازی شد [۱۶، ۲۲]. الگوریتم استاندارد و نیز الگوریتم فازی ICA هر دو بر روی آن پیاده‌سازی شد، سپس به منظور مقایسه بهتر و جامع‌تر از دو الگوریتم ژنتیک (GA) استاندارد [۳۵] و همینطور الگوریتم بهینه‌سازی گروهبی ذرات (PSO) استاندارد [۳۶] (با پیش فرض آشنایی با این دو الگوریتم) به منظور اعمال روی توابع جدول [۴] استفاده می‌نماییم که پارامترهای تنظیمی این دو الگوریتم نیز در جدول (۵) معرفی شده‌اند.

جدول ۵- پارامترهای قابل تنظیم در الگوریتم‌های PSO، GA

Algorithm	Setting	Setting
GA	pCrossover(Recom)=0.7	pMutation=0.2
PSO	W = 0.729 C ₁ =1.4962	C ₂ =1.4962

پس از اعمال الگوریتم‌ها و همینطور پارامترهای تنظیمی آن‌ها، اولین نتیجه مقایسه در شکل (۱۰) رسم شده است.

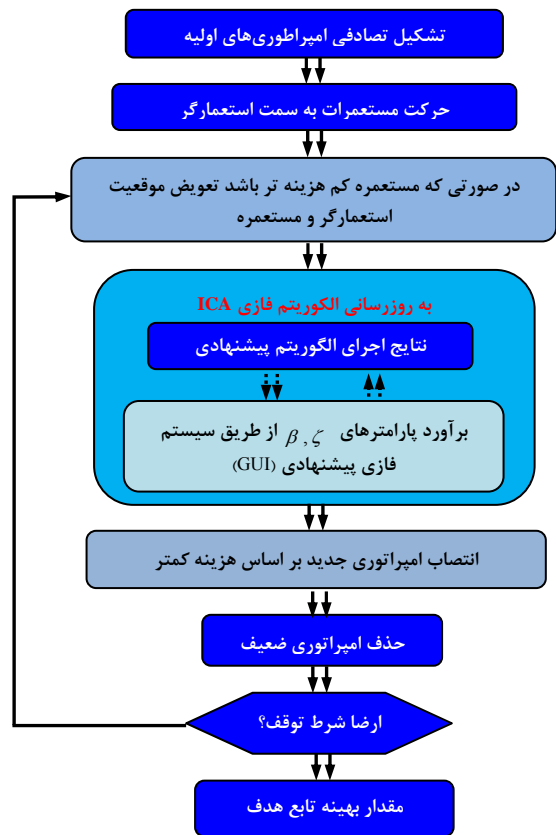


شکل ۱۰- ارزیابی الگوریتم فازی رقابت استعماری با الگوریتم‌های ICA استاندارد، PSO و GA در حل توابع از [۱۶، ۲۲]

به منظور بررسی بهتر و دقیق‌تر سیستم فازی طراحی شده برای بهبود عملکرد الگوریتم رقابت استعماری، نتایج اولین ارزیابی که در شکل (۱۰) مشخص شده‌اند، را بصورت لگاریتمی در شکل (۱۱) نمایش می‌دهیم. توابع جدول (۴)، دارای نقاط بهینه محلی فراوان و از طرف دیگر دارای نقطه بهینه سراسری هستند. به همین منظور از ۴ تابع مشهور جدول (۴) از [۴] در ابعاد ۵۰، ۱۰۰ و ۲۰۰ استفاده می‌کنیم، هدف مینیمم‌سازی این توابع در نقطه صفر است، نتایج بدست آمده در بهینه‌سازی تابع Rastregin در ۵۰، ۱۰۰ و ۲۰۰ بعد در شکل‌های (۱۲) و (۱۳) و (۱۴) مشاهده می‌شود.

۵- ارزیابی الگوریتم فازی ICA (Fuzzy ICA)

در الگوریتم ICA اساس کار بر روی تغییر وضعیت و جستجوی فضای مساله توسط استعمارگر می‌باشد. برای بررسی میزان صحت و دقت، الگوریتم‌ها در نرم‌افزار MATLAB® با استفاده از راهنمایی‌های [۱۸، ۱۹] بر روی توابع ریاضی مسائل محک [۴] مطابق با جدول (۴) شبیه‌سازی شدند و مورد ارزیابی قرار گرفته‌اند. فلوجارت الگوریتم فازی رقابت استعماری طبق فلوجارت (۳) می‌باشد. همچنین برای رسم نمودارهای بدست آمده از نرم افزار Origin® استفاده شد.

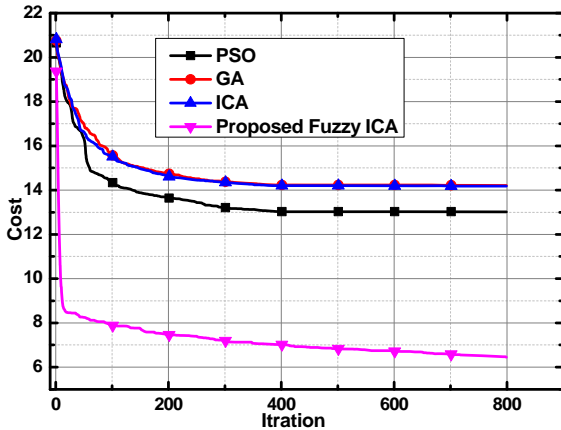


شکل ۹- روند اجرای الگوریتم رقابت استعماری پیشنهادی

توابع ارزیابی پیشنهادی در مرجع اصلی الگوریتم رقابت استعماری در جدول (۴) بیان شده‌اند.

تهیه شده است)، دقت مناسبی ندارد، و خیلی کندتر نیز به همگرایی رسیده است. شکل (۱۳) روند همگرایی الگوریتم پیشنهادی در بهینه‌سازی تابع Rastregin ۱۰۰ بعدی را نشان می‌دهد.

شکل ۱۴ نیز مشخص است که الگوریتم استاندارد و سایر الگوریتم‌ها، سرعت و مقداری را که بهینه کرده‌اند، ضعیف تر و کندتر از الگوریتم پیشنهادی بوده است.

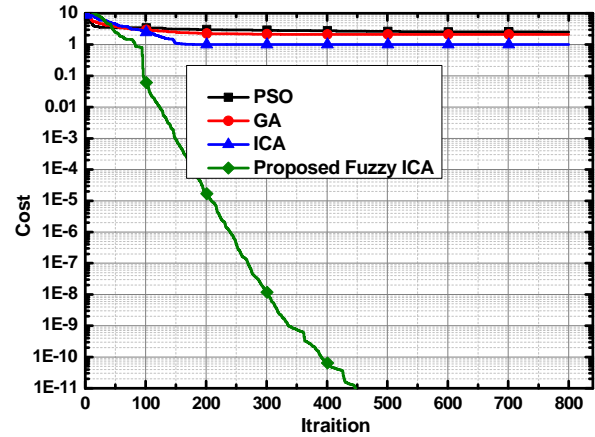


شکل ۱۴- تقابل الگوریتم پیشنهادی و الگوریتم‌های مذکور در ۲۰۰ بعد

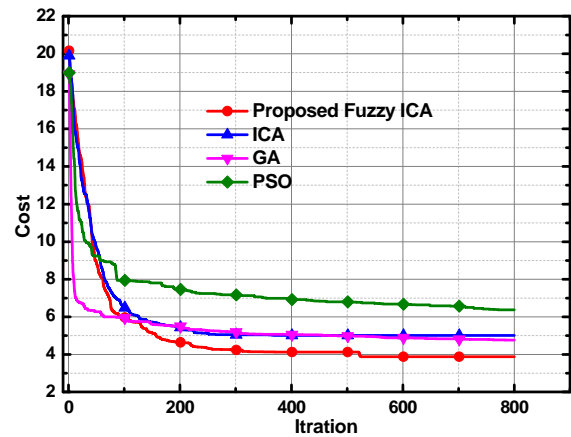
در نهایت در اشکال (۸ و ۹ و ۱۰ و ۱۱)، اجرای الگوریتم‌های مذکور در شرایط یکسان (تعداد تلاش الگوریتم برای همگرایی یا به عبارتی دیگر ایجاد شرط ارضاء یکسان) پس از میانگین گیری در ۱۰۰ بار اجرا، بهترین نتیجه به ازای ۸۰۰ بار اجرای هر الگوریتم دیده می‌شود. تا این مرحله، سیستم فازی پیشنهادی مورد ارزیابی قرار گرفت. براساس نتایج مشخص شده در اشکال (۸ و ۹ و ۱۰ و ۱۱) به خوبی برتری الگوریتم فازی رقابت استعماری مشخص می‌شود. پس می‌توان نتیجه گرفت که سیستم فازی طراحی شده که در الگوریتم رقابت استعماری جایگذاری شده است، به خوبی الگوریتم رقابت استعماری را هدایت کرده و با توجه به مقادیر ورودی برای این الگوریتم، خروجی‌های ضریب جذب و هزینه مستعمرات را کنترل و هدایت کرده است.

در پیاده‌سازی سیستم فازی مشکل اصلی که وجود دارد در حله اول این است که، خود سیستم فازی به علت انجام محاسبات مطمئناً در افزایش زمان محاسبات دخیل خواهد بود و نکته مهم تر این که برای تعمیم الگوریتم فازی رقابت استعماری در زبان‌های دیگر برنامه نویسی (واسط کاربری یا GUI که موجب طراحی سیستم فازی شده است، فقط در زبان MATLAB وجود دارد) مانند، ++C و Java، ... باید بتوانیم سیستم فازی را پیاده‌سازی نماییم، که البته استفاده از واسط کاربری (GUI) امکان پذیر نمی‌باشد و از طرف دیگر اگر سعی در نوشتن سیستم فازی به همان زبان کنیم، حتماً کاری سخت، طاقت فرسا و سرعت گیر خواهد بود، مضافاً بر این که در کنار کاهش حجم محاسبات به دلیل کاهش حجم حافظه، کاهش انرژی و افزایش سرعت در کنار کاهش چشمگیر هزینه‌ها را نیز می‌توان در نظر گرفت.

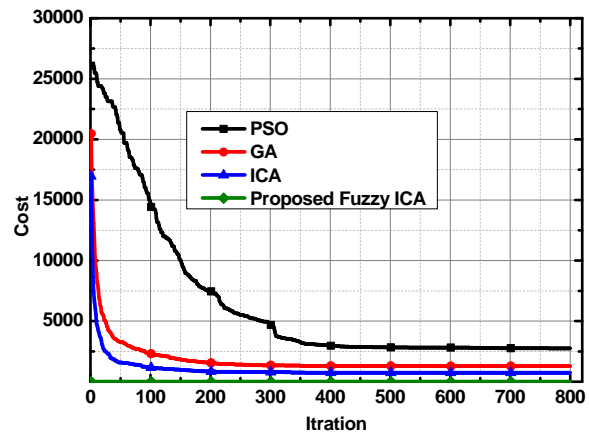
رگرسیون چندمتغیره یکی از الگوریتم‌های پیشنهادی است که می‌تواند اصطلاحاً سیستم فازی را در یک رابطه چند جمله‌ای خلاصه نماید. برای اعمال رگرسیون چند متغیره، باید به رفتار سیستم مورد نظر دسترسی داشته باشیم، یکی از پارامترهایی که موجب شناسایی و تعریف عملکرد سیستم فازی می‌شود، سطح فیزیکی یا Surface است، که بیان کننده رابطه بین ورودی‌ها و خروجی‌های یک سیستم (در اشکال ۵ و ۶) می‌باشد. در قسمت بعد به نحوه استخراج پارامترهای سیستم فازی یا خلاصه سازی سیستم فازی در یک رابطه غیرخطی چند جمله‌ای می‌پردازیم.



شکل ۱۱- نمایش لگاریتمی الگوریتم رقابت استعماری فازی در حل تابع [۱۶]، [۲۲] که در اولین ارزیابی الگوریتم رقابت استعماری بیان شده است



شکل ۱۲- تقابل الگوریتم پیشنهادی فازی رقابت استعماری و الگوریتم‌های مذکور در ۵۰ بعد



شکل ۱۳- تقابل الگوریتم پیشنهادی و الگوریتم‌های مذکور در ۱۰۰ بعد

در شکل‌های ۱۲ و ۱۳ و ۱۴ مقایسه عملکرد الگوریتم پیشنهادی با الگوریتم‌های مذکور را در بهینه‌سازی تابع Rastregin به ترتیب به ابعاد ۵۰، ۱۰۰ و ۲۰۰ نشان می‌دهد. در این شکل‌ها به خوبی مشخص است که الگوریتم پیشنهادی از همان ابتدا همگرا شده و کمترین مقادیر را در بهینه‌سازی بدست آورده است، اما الگوریتم‌های دیگر حتی ICA (با توجه به برنامه‌ای که در [۲۲]

۶- الگوریتم رقابت استعماری پیشنهادی

در این قسمت از الگوریتم پیشنهادی که در نهایت منجر به افزایش سرعت و دقت الگوریتم استعماری می‌شود، سعی بر آن داریم تا قسمت فازی الگوریتم رقابت استعماری فازی به یک تابع غیرخطی خلاصه شود، یعنی توسط یک رابطه چند جمله‌ای عملاً قسمت فازی حذف می‌شود و به جای آن از یک رابطه چند جمله‌ای استفاده می‌کنیم تا مقادیر جذب و هزینه متعمرات برآورد شوند تا از آنها در رابطه نهایی استفاده شوند. برای این کار لازم است تا از رفتار سیستم فازی اطلاع یابیم، سیستم فازی طراحی شده توسط سطح فیزیکی (Surface) در واسط کاربری (GUI) بیان کننده رابطه بین ورودی‌ها و خروجی‌های سیستم فازی است. با اعمال رگرسیون چند متغیره بر روی سطح فیزیکی، روابط چند جمله‌ای بدست خواهند آمد. به همین منظور در قسمت‌های بعدی ابتدا رگرسیون چند متغیره بحث و بررسی می‌شود سپس الگوریتم نهایی و در آخر ارزیابی الگوریتم پیشنهادی بیان می‌شود.

۶-۱- رگرسیون چند متغیره

یکی از هدف‌های اصلی پژوهش‌های آماری ایجاد روابطی است تا پیش بینی یک یا چند متغیر را بر حسب سایرین میسر گرداند. مسئله اصلی رگرسیون عبارت است از تعیین میانگین شرطی مقدار Y به ازای مقادیر مفروضی از X . رابطه (۶) نشان دهنده معادله رگرسیون خطی می باشد.

$$\mu_Y|_X = \alpha + \beta x \quad (6)$$

با توجه به اینکه برخلاف بودن رگرسیون تأکید کردیم، هدف از حل مسئله ضرایب α ، β از روی داده‌های نمونه می‌باشد، به طوری که خط $y = \alpha + \beta x$ بهترین برازش ممکن برای داده‌های مفروض شود. یکی از روش‌هایی که معمولاً برای تعیین ضرایب α ، β مورد استفاده قرار می‌گیرد روش کمترین مربعات خطا (Least Square Error) است که در اوایل قرن نوزدهم توسط ریاضیدان فرانسوی آدرین لژاندر (Adrien Legendre) پیشنهاد شده است [۲۳]. مسائل متعددی وجود دارند که در آنها می‌توان متغیری را بر حسب متغیر دیگر به صورت دقیق پیش گویی کرد، اما با در نظر گرفتن اطلاعات بیشتر می‌توان پیشگویی‌های بهتری انجام داد. بدین ترتیب رگرسیون چند متغیره تکامل یافته رگرسیون تک متغیره می باشد. گرچه روابط متعددی موجودند که می‌توان از آنها برای بیان روابط رگرسیونی بین بیش از دو متغیره، استفاده کرد [۲۴]، البته استفاده از رگرسیون چند متغیره در علوم متفاوت همچون مراجع [۲۵-۳۴] رایج بوده است، که کاربردهای متعددی از قبیل تخمین، پیش بینی و برازش منحنی دارد. رایجترین آنها معادلات خطی، به صورت رابطه (۷) می‌باشند.

$$\mu_{Y|X_1, X_2, X_3, \dots, X_k} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k \quad (7)$$

به دلایل متعدد معادلات رگرسیون خطی مورد توجه خاصی هستند، این معادلات به سادگی در سایر اعمال ریاضی وارد می‌شوند و اغلب، تقریب خوبی برای معادلات پیچیده هستند. در رابطه (۷)، Y متغیری تصادفی است که می‌خواهیم مقادیر آن را برحسب مقدارهای معلوم x_1, x_2, \dots, x_k پیشگویی کنیم، و $\beta_0, \beta_1, \dots, \beta_k$ ضرایب رگرسیون چندگانه، اعداد ثابتی هستند که باید از روی داده‌های مشاهده شده Y ، تعیین شوند. ضرایب رگرسیون چندگانه به کمک روش

کمترین مربعات خطا برآورد می‌شود. اگر n نقطه داده به صورت رابطه (۸) داشته باشیم.

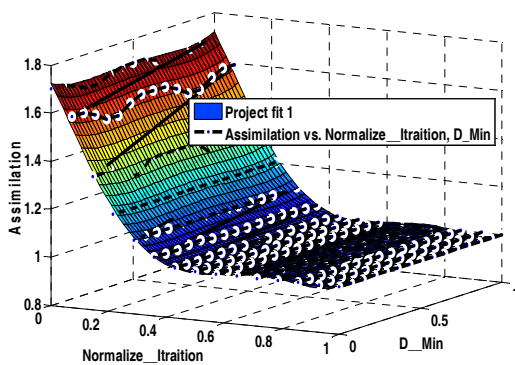
$$\{(x_{i1}, x_{i2}, \dots, x_{ik}, y_i) ; i = 1, 2, \dots, n\} \quad (8)$$

در این صورت هدف از بکارگیری روش کمترین مربعات خطا، یافتن مقادیری از $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$ می‌باشد که به ازای آنها کمیت q در رابطه (۹) مینیمم شود. در این معادله و معادلات بعدی، x_{i1} ، x_{i2} ، \dots ، x_{ik} ، y_i ، i امین مقدار متغیر x_1 می‌باشد.

$$q = \sum_{i=1}^n [\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik}]^2 \quad (9)$$

برای مینیمم کردن q ، نسبت به β ها، مشتق جزئی می‌گیریم و با برابر صفر قرار دادن این مشتق‌های جزئی و ساده سازی آن‌ها، مجموعه روابط (۱۰) را بدست می‌آوریم:

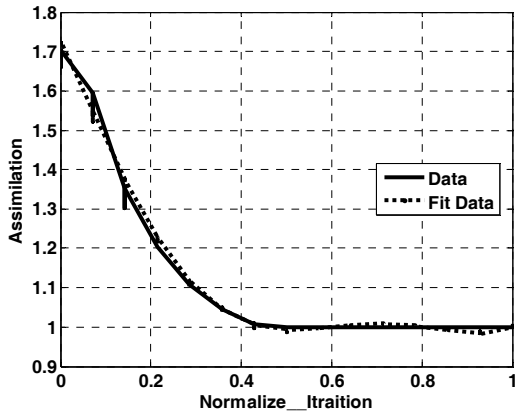
$$\begin{aligned} \sum_{i=1}^n y_i &= \beta_0 n + \beta_1 \sum_{i=1}^n x_{i1} + \beta_2 \sum_{i=1}^n x_{i2} + \dots + \beta_k \sum_{i=1}^n x_{ik} \\ \sum_{i=1}^n x_{i1} y_i &= \beta_0 \sum_{i=1}^n x_{i1} + \beta_1 \sum_{i=1}^n x_{i1}^2 + \beta_2 \sum_{i=1}^n x_{i1} x_{i2} + \dots + \beta_k \sum_{i=1}^n x_{i1} x_{ik} \\ \sum_{i=1}^n x_{ik} y_i &= \beta_0 \sum_{i=1}^n x_{ik} + \beta_1 \sum_{i=1}^n x_{ik} x_{i1} + \beta_2 \sum_{i=1}^n x_{ik} x_{i2} + \dots + \beta_k \sum_{i=1}^n x_{ik}^2 \end{aligned} \quad (10)$$



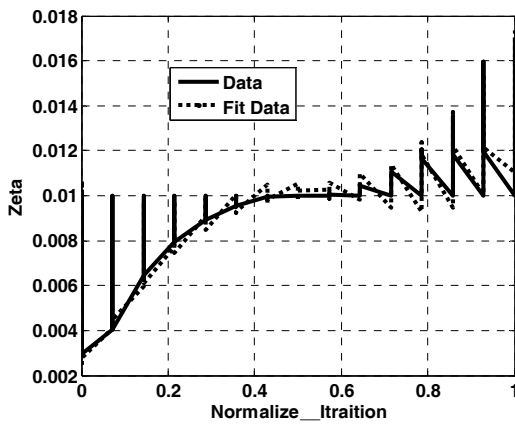
شکل ۱۵- خروجی سطح فیزیکی توسط رگرسیون چند متغیره برای خروجی اول بر اساس شکل (۷)، رابطه ۱۰ و سیستم فازی مورد نظر

بدین ترتیب از حل $k+1$ معادله فوق می‌توان متغیرهای مجهول $\beta_0, \beta_1, \dots, \beta_k$ و با قرار دادن آن‌ها در معادله (۷) تقریبی خطی برای داده‌های مورد نظر بدست آورد. اکنون با توجه به مباحث مطرح شده، می‌توان رگرسیون چندگانه را برای داده‌های خروجی کنترل کننده فازی مورد بحث که نمودارهای ورودی - خروجی آن در شکل (۵) و (۶) رسم شده بود، انجام داد. برای افزایش دقت در تقریب داده در تقریب داده‌های حاصل از خروجی سیستم فازی مورد بحث، آن را مطابق شکل (۱۲) و (۱۳) به نواحی متعدد تقسیم می‌کنیم.

مقایسه نماییم، به همین منظور به بررسی قطعات جانبی صفحات سیستم فازی و صفحه طراحی بدست آمده توسط رگرسیون چند متغیره می‌پردازیم. در شکل (۱۴) رابطه بین ورودی اول و یکی از خروجی‌های (هزینه مستعمرات) سیستم اصلی فازی مشخص شده است. در شکل (۱۵) رابطه بین همان متغیر ورودی از سیستم فازی و خروجی دیگر (ضریب جذب) مشخص شده است. با کمی دقت مشخص می‌شود که مقادیر مشخص شده و بیان شده توسط روش رگرسیون چند متغیره با مقادیر سیستم فازی در هر ۲ خروجی تقریباً برابر با مقادیر اصلی می‌باشند. نکات بیان شده در هر ۲ شکل (۱۷) و (۱۸) مشخص شده‌اند. در آخر پس از بحث و ارائه سیستم فازی پیشنهادی، ارزیابی پیشنهادی را ارائه می‌نماییم.



شکل ۱۷- نمایش مقادیر مشخص شده ورودی اول و خروجی اول توسط رگرسیون چند متغیره بر اساس مقادیر واقعی در سیستم فازی

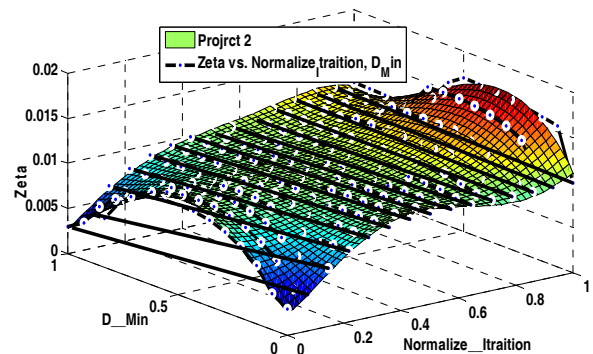


شکل ۱۸- نمایش مقادیر مشخص شده ورودی اول و خروجی دوم توسط رگرسیون چند متغیره بر اساس مقادیر واقعی در سیستم فازی

۷- ارزیابی الگوریتم

در این قسمت نیز برای شبیه‌سازی از نرم‌افزار MATLAB® استفاده شد. ابتدا سیستم فازی براساس مطالعه روی مقاله اصلی الگوریتم رقابت استعماری در [۴] طراحی شد، سپس با استفاده از دانش فرد خیره، تسلط بر الگوریتم استعماری و منطق فازی، سیستمی فازی به منظور بهبود عملکرد الگوریتم رقابت استعماری طراحی شد، سپس سیستم فازی طراحی شده پیاده‌سازی شد، در بخش (۵) سیستم فازی مورد ارزیابی قرار گرفت تا صحت سیستم فازی طراحی شده مورد بررسی قرار گیرد. پس از اثبات عملکرد سیستم فازی، از جعبه ابزار sftool در نرم افزار MATLAB® استفاده شد؛ تا اینکه توسط رابطه‌ای غیرخطی، عملکرد

تا برآوردی دقیق از شکل‌های (۷ و ۸) که معرفی کننده سیستم فازی هستند، بدست آید. در روابط (۱۱) توابع بدست آمده با استفاده از رگرسیون دوگانه و نواحی آنها آمده است. در شکل (۱۶) خروجی رگرسیون چند متغیره برای خروجی دوم مشخص شده است.



شکل ۱۶- خروجی سطح فیزیکی توسط رگرسیون چند متغیره برای خروجی دوم، بر اساس شکل (۸)، رابطه ۱۰ و سیستم فازی مورد نظر

برای اعمال رگرسیون خطی از جعبه ابزار sftool® در نرم‌افزار MATLAB® استفاده شده است. با اعمال رگرسیون چند متغیره، رابطه (۱۱) استخراج می‌شود، رابطه (۱۱) رابطه‌ای تقریبی یا معادله‌ای غیرخطی از سیستم فازی طراحی شده است.

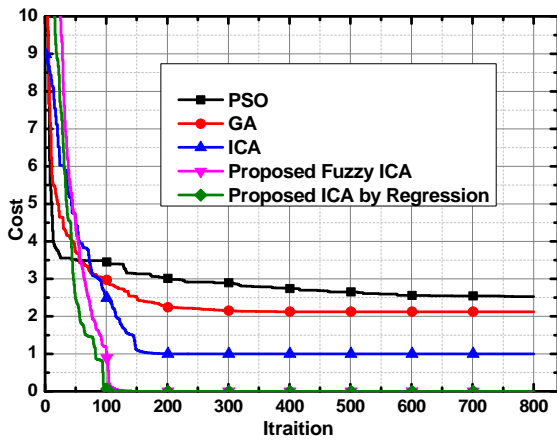
$$F(x, y)_{\text{Assimilation \& Zeta}} = P_{00} + P_{10}x + P_{01}y + P_{20}x^2 + P_{11}xy + P_{02}y^2 + P_{30}x^3 + P_{21}x^2y + P_{12}xy^2 + P_{03}y^3 + P_{40}x^4 + P_{31}x^3y + P_{22}x^2y^2 + P_{13}xy^3 + P_{04}y^4 + P_{50}x^5 + P_{41}x^4y + P_{32}x^3y^2 + P_{23}x^2y^3 + P_{14}xy^4 + P_{05}y^5 \quad (11)$$

مقادیر معرفی شده یا مقادیر عددی ثابت توسط جعبه ابزار در جدول (۶) مشخص شده‌اند.

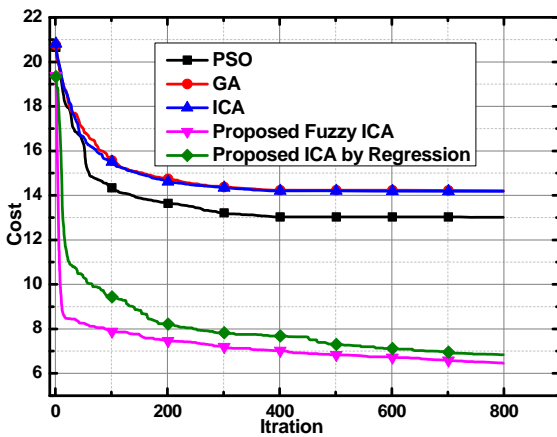
جدول ۶- پارامترهای معرفی شده توسط جعبه ابزار Sftool

Varies	Assimilation Fun	Zeta Fun
P ₀₀	1.725	0.002453
P ₁₀	-2.333	0.01841
P ₀₁	-0.4146	0.02594
P ₂₀	-2.992	0.07839
P ₁₁	0.9056	-0.1748
P ₀₂	1.675	0.01049
P ₃₀	19.12	-0.2922
P ₂₁	-0.6865	0.1788
P ₁₂	-2.674	0.2362
P ₀₃	-2.521	-0.08615
P ₄₀	-24.09	0.3014
P ₃₁	0.328	0.09883
P ₂₂	0.6865	-0.363
P ₁₃	3.536	-0.03292
P ₀₄	1.26	0.0581
P ₅₀	9.57	-0.0974
P ₄₁	2.071e-013	-0.08906
P ₃₂	-0.328	0.05029
P ₂₃	3.435e-014	0.119
P ₁₄	-1.768	-0.02324
P ₀₅	1.157e-013	-0.008066

به منظور بررسی عملکرد جعبه ابزار ناگزیر باید صفحات اصلی (متعلق به سیستم فازی) و صفحات بدست آمده از رابطه رگرسیون چند متغیره، با یکدیگر را



شکل ۲۰- تقابل الگوریتم پیشنهادی با سایر الگوریتم‌های بهینه‌سازی



شکل ۲۱- تقابل الگوریتم بهینه‌سازی پیشنهادی با سایر الگوریتم‌های بهینه‌سازی برای بهینه‌سازی تابع Rstregin ۱۰۰ بعدی

در جدول (۷) به خوبی مشهود است که، الگوریتم پیشنهادی در بهینه‌سازی توابع مذکور برای ۲۰ بعد عملکرد برتری داشته است. در جدول (۸) نیز نتایج تقابل بهینه‌سازی الگوریتم پیشنهادی با سایر الگوریتم‌ها برای بهینه‌سازی توابع [۴] به ازای ۵۰ بعد مشخص شده است.

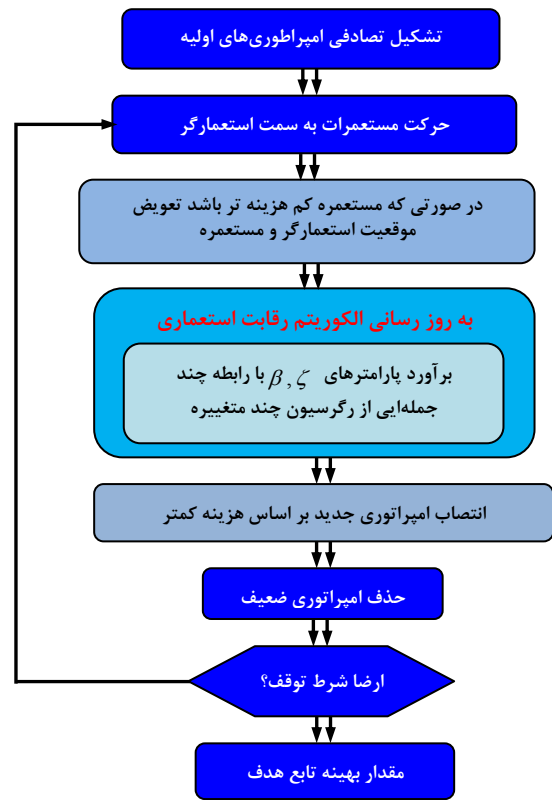
جدول ۸- تقابل الگوریتم‌های بهینه‌سازی به منظور در بهینه‌سازی توابع مذکور برای بهینه‌سازی ۵۰ بعدی توابع [۴]

Fcn by 50 Dimension	F1	F2	F3	F4
GA	5.01315	0.00124	-0.9359	356.296
PSO	6.36284	0.107	-0.9359	454.921
ICA	4.76591	0.00124	0.9359	250.932
Fuzzy ICA	3.87501	8.028E-11	-0.9359	152.456
MRICA	3.98162	5.788E-7	-0.9359	158.173

جدول ۹- تقابل بهینه‌سازی الگوریتم پیشنهادی با الگوریتم‌های [۴] به‌منظور بهینه‌سازی توابع مذکور به ازای ۱۰۰ بعد

Fcn by 100 Dimension	F1	F2	F3	F4
GA	9.15734	49.30344	-0.959	1271.8
PSO	9.54876	1077.2126	-0.959	2752.7
ICA	8.63942	1.19571	-0.939	1271.8
Fuzzy ICA	4.41572	0.00106	-0.959	-0.939
MRICA	5.43394	0.01274	-0.959	-0.939

سیستم فازی توسط چند جمله‌ای (۱۱) بیان گردید که در جدول (۸) مشخص و خلاصه شد. سپس رابطه چند جمله‌ای (۱۱) در الگوریتم اصلی هنگام بروز رسانی الگوریتم جاسازی شد تا مقادیر ضریب جذب و هزینه مستعمرات از این طریق برآورد شدند. الگوریتم نهایی مانند فلوچارت (۴) می‌باشد.



شکل ۱۹- روند اجرایی الگوریتم نهایی معرفی شده

به منظور بررسی و عملکرد الگوریتم پیشنهادی، از توابع محک معرفی شده در جدول (۴) از مرجع [۴] استفاده می‌نماییم. سپس نتایج مشخص شده را با الگوریتم‌های GA, PSO, Fuzzy ICA و الگوریتم پیشنهادی در ابعاد (۱۵ و ۲۰ و ۵۰ و ۱۰۰ و ۲۰۰) بررسی می‌نماییم. در انتها نتایج در اشکال (۱۶ و ۱۷) و جدول (۶ و ۷ و ۸ و ۹) بیان می‌شوند. در شکل (۲۰) نتایج اجرای الگوریتم پیشنهادی در حل تابع Rastregin ۱۵ بعدی، همان تابعی که اولین بار توسط [۴] معرفی شده است، مشخص می‌شود.

در شکل (۲۱) نتایج تقابل الگوریتم پیشنهادی با سایر الگوریتم‌های بهینه‌سازی در بهینه‌سازی تابع Rstregin ۱۰۰ بعدی را نمایش می‌دهد.

به منظور اثبات نتایج الگوریتم پیشنهادی، از نمایش اشکال در بهینه‌سازی توابع [۴]، خود دارای می‌کنیم و نتایج حاصل از میانگین گیری ۱۰۰ مرتبه‌ای را توسط جداول (۶ و ۷ و ۸ و ۹) نمایش می‌دهیم. در جدول (۷) مقادیر بهینه شده سایر الگوریتم‌های بهینه‌سازی بهمراه الگوریتم پیشنهادی برای بهینه‌سازی ۲۰ بعد توابع [۴] نمایش داده شده است.

جدول ۷- تقابل بهینه‌سازی الگوریتم پیشنهادی با سایر الگوریتم در توابع ۲۰ بعدی جدول [۴]

Fcn by 20 Dimension	F1	F2	F3	F4
GA	1.841728	0.0046	-0.9359	73.372
PSO	3.905521	0.84278	-0.9359	33.828
ICA	1.3944E-13	1.127E-22	-0.9359	27.858
Fuzzy ICA	5.4176E-15	2.5E-53	-0.9359	4.9748
MRICA	1.50921E-14	6.074E-29	-0.9359	26.207

توجه به ماهیت تصادفی روش حل 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 رقابت استعماری (ICA) برای حل توابع غیرخطی 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 سه ایده بهینه‌سازی رقابت استعماری، منطق فازی و همینطور رگرسیون چند متغیره می‌تواند پاسخی هوشمند برای این چالش محسوب شود. نتایج شبیه سازی نشان می‌دهد که الگوریتم استاندارد استعداد گیر افتادن در بهینه محلی را دارد. استفاده از منطق فازی و به طبع آن رگرسیون چند متغیره 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. در نهایت تمام تلاش‌ها برای بهبود عملکرد الگوریتم رقابت استعماری بود که البته منجر شد تا الگوریتم استعماری بهبود یافته بر اساس رگرسیون چند متغیره یا Improved ICA by based on Fuzzy Logic Multivariable Regression معرفی شود.

مراجع

[۱] م. شیبانی، مدل‌های ترکیبی بر پایه اتوماتای یادگیر سلولی و محاسبات تکاملی و کاربردهای آنها، پایان نامه کارشناسی‌ارشد مهندسی کامپیوتر، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، ۱۳۸۵.

[2] H. Sarimveis, and A. Nikolakopoulos, "A Line Up Evolutionary Algorithm for Solving Nonlinear Constrained Optimization Problems," *Journal of Computers and Operations Research*, vol. 32, no. 6, pp. 1499-1514, 2005.

[3] H. A. Abbass, R. Sarker, and C. Newton, "PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems," *Proc. IEEE Intl Conf. Genetic Programming*, pp. 503-511, 2001.

[۴] م. عبدچیری، م. صنیعی آبا، و م. بهرامی، "الگوریتم رقابت استعماری با شعاع حرکت تطبیقی"، دانشگاه تربیت مدرس کنفرانس، ۱۳۸۶.

[5] H. Muhlenbein, M. Schomisch, and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," *Proc. IEEE Intl Conf. Genetic Programming*, pp. 270-278, 1991.

[6] M. Melanie, *An Introduction to Genetic Algorithms*, Massachusetts, MIT Press, 1999.

[7] P. J. Angeline, "Evolving Fractal Movies," *Proc. IEEE Intl Conf. Genetic Programming*, pp. 503-511, 1997.

[8] J. Kennedy, and R. C. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Intl Conf. Neural Networks*, pp. 1942-1948, 1995.

[9] J. Kennedy, and R. C. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Intl Conf. Neural Networks*, pp. 1942-1948, 1995.

[10] X. Yang, J. Yuan, J. Yuan, and H. Mao, "A Modified Particle Swarm Optimizer with Dynamic Adaptation,"

در ادامه نیز به بررسی و تقابل الگوریتم‌های بهینه سازی در جدول (۹) برای بهینه‌سازی توابع مذکور به ازای توابع ۱۰۰ و ۲۰۰ بعدی می‌پردازیم.

در جدول (۹) نتایج اجرای الگوریتم پیشنهادی نیز در یک شرایط یکسان (شرط ارضاء) مشخص شده است. نکته‌ی قابل تامل در این بررسی، تفاوت چشم گیر، همگرایی و دقت دیگر الگوریتم‌های بحث شده در این ارزیابی در مقایسه با الگوریتم پیشنهادی است، به عنوان مثال در سطر آخر جدول (۹) و شکل (۲۱) به خوبی مشخص است که الگوریتم پیشنهادی دارای یک سرعت بالاتر در همگرایی و همینطور دقت برتر در کاوش بمنظور بدست آوردن نقطه بهینه را دارد. در جدول (۱۰) نیز تقابل الگوریتم پیشنهادی با سایر الگوریتم‌های بهینه سازی در ۲۰۰ بعد نشان داده شده است.

جدول ۱۰- تقابل بهینه‌سازی الگوریتم پیشنهادی با الگوریتم‌های [۴] به‌منظور بهینه‌سازی توابع مذکور به ازای ۱۰۰ و ۲۰۰ بعد

Fcn by 200 Dimension	F1	F2	F3	F4
GA	14.187	7022.902	-0.939	4480.04
PSO	14.24	195.15	-0.959	8035.5
ICA	13.029	40.171	-0.939	2296.2
Fuzzy ICA	6.43	1.70	-0.959	1269.6
MRICA	6.879	5.56	-0.959	2181.3

در تمامی این ارزیابی‌ها سعی شد تا الگوریتم‌های بررسی شده به درستی پیاده شوند، تا در نتایج کمترین خطا و نهایت دقت بوجود آید. با توجه به نتایج از قبیل شکل‌های (۲۰) و (۲۱) و همینطور جداول (۷) و (۸) و (۹) و (۱۰)، به راحتی می‌توان برتری الگوریتم پیشنهادی را نتیجه گرفت.

۸- بحث و نتیجه‌گیری

روند ارائه و تکامل در الگوریتم‌های بهینه‌سازی رو به رشد است، و همواره سعی بر این است تا نتایج این پیشرفت‌ها منجر به افزایش سرعت همگرایی و بهبود عمق کاوش در بهینه‌سازی شود. یکی از این الگوریتم‌ها، الگوریتم رقابت استعماری است، که در نوع خود قابل تحسین است. متأسفانه این الگوریتم نیز مشکل گیر افتادن در بهینه محلی، کندی سرعت در بهینه‌سازی و مهم تر از همه تنظیم صحیح پارامترهایش را در شرایط متفاوت دارد. ایده استفاده از منطق فازی برای رفع این مشکلات تحت عنوان اولین مدل فازی الگوریتم رقابت استعماری (تا زمان نگارش این اثر موردی در مراجع معتبر، نمایه نشده است) در اینجا مطرح شد. با استفاده از نرم افزار MATLAB® الگوریتم فازی پیشنهادی پیاده شد و سپس با توجه به مسائل مشهور محک ارزیابی گسترده انجام شد، سپس با توجه به مشکلاتی که در مورد پیاده‌سازی سیستم‌های فازی شد، از رگرسیون چند متغیره برای پیاده‌سازی و اصطلاحاً خلاصه‌سازی سیستم فازی در یک چند جمله‌ای غیرخطی استفاده شد. در ارزیابی نهایی طبق فلوجارت ۴، نحوه صحیح پیاده‌سازی الگوریتم پیشنهادی ارائه شد، سپس بحث و بررسی پیرامون ارزیابی الگوریتم پیشنهادی در شکل‌ها و جداولی ارائه شدند. با توجه به نتایج بدست آمده می‌توان نتیجه گرفت که الگوریتم پیشنهادی تقریباً در تمامی توابع محک استفاده شده بهترین بوده و برتری محسوسی نسبت به دیگر الگوریتم‌های مذکور دارد. این الگوریتم در تشخیص بهینه‌های محلی موفق بوده و توانسته خود را از دام آنها رها کند و بهینه سراسری را پیدا کند. در بعضی از نمودارها همانند شکل‌های (۲۰) و (۲۱) رفتار نسبتاً غیرخطی و حتی قابل تاملی از توابع دیده می‌شود که این بخاطر مقادیر کاملاً تصادفی و غیرقابل پیش بینی تولیدی توسط الگوریتم در تلاش برای همگرایی است. می‌توان گفت که استفاده از توابع غیرخطی رفتار الگوریتم استاندارد را با

- [26] S. Chehreh Chelgani, J. C. Hower, and B. Hart. "Estimation of Free-swelling Index based on Coal Analysis using Multivariable Regression and Artificial Neural Network," *Journal of Fuel Processing Technology*, vol. 9, no. 2, pp. 349-355, 2011.
- [27] A. R. Jafari, and T. Babadagli, "Estimation of Equivalent Fracture Network Permeability using Fractal and Statistical Network Properties," *Journal of Petroleum Science and Engineering*, vol. 5, no. 2, pp. 110-123, 2012.
- [28] L. dos Santos Coelho, and V. Cocco Mariani, "Firefly Algorithm Approach based on Chaotic Tinkerbell Map Applied to Multivariable PID Controller Tuning," *Journal of Computers and Mathematics with Applications*, vol. 64, no. 2, pp. 2371-2381, 2012.
- [29] N. Azam Baleghy, and S. K. Mousavi Mashhadi, "Design and Implementation Fuzzy Controller in the Frost-free Refrigerator by using Multivariate Regression," *Proc. IEEE Intl Conf. Electrical Engineering*, pp. 156-162, 2012.
- [30] L. Yong-song, Y. Jian-min, C. Jian-ping, and X. Jing, "Analysis of 3D In-situ Stress Field and Query System's Development based on Visual BP Neural Network," *Journal of Procedia Earth and Planetary Science*, vol. 5, no. 2, pp. 64-69, 2012.
- [31] K. Salahshoor, and M. Hamz ehnejad, "A Novel Online Affine Model Identification of Multivariable Processes using Adaptive Neuro-fuzzy Networks," *Journal of Chemical Engineering Research and Design*, vol. 8, no. 8, pp. 155-169, 2010.
- [32] D. Ansari, J. Nilsson, R. Andersson, S. Regner, B. Tingstedt, and B. Andersson, "Artificial Neural Networks Predict Survival from Pancreatic Cancer after Radical Surgery," *American Journal of Surgery*, vol. 20, no. 5, pp. 1-7, 2013.
- [33] Q. Lu, and M. Mahfouf, "Multivariable Self-organizing Fuzzy Logic Control using Dynamic Performance Index and Linguistic Compensators," *Journal of Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 1537-1547, 2012.
- [34] T. A. Choudhury, N. Hosseinzadeh, and C. C. Berndt, "Artificial Neural Network Application for Predicting in-flight Particle Characteristics of an Atmospheric Plasma Spray Process," *Journal of Surface and Coatings Technology*, vol. 20, no. 5, pp. 4886-4895, 2011.
- [35] E. Kinnear, L. Kenneth, L. Spector, and P. J. Angeline, *Advances in Genetic Programming*, Ph. D. Dissertation, MIT, Cambridge, MA, 1999.
- [36] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Intl Conf. Neural Networks*, pp. 1942-194, 1995.
- Journal of Applied Mathematics and Computation*, vol. 18, no. 2, pp. 1205-1213, 2007.
- [11] I. Tokuda, K. Aihara, and T. Nagashima, "Adaptive Annealing for Chaotic Optimization," *Journal of Physics*, vol. 58, no. 4, pp. 34-46, 1998.
- [12] A. Ingber, "Simulated Annealing: Practice versus Theory," *Journal of Mathematics, Computer and Modeling*, vol. 18, no. 11, pp. 29-57, 1993.
- [13] M. F. Cardoso, R. L. Salcedo, S. F. Azevedo and D. Barbosa, "A Simulated Annealing Approach to the Solution of Minlp Problems," *Journal of Computer, Chemistry and Engineering*, vol. 21, no. 12, pp. 1349-1364, 1997.
- [14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Journal of Science*, vol. 22, no. 2, pp. 671-680, 1983.
- [15] B. Franklin, and M. Bergerman, "Cultural Algorithms: Concepts and Experiments," *Proc. IEEE Intl Conf. Evolutionary Computation*, pp. 1245-1251, 2000.
- [16] A. Gargari, and C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition," *Proc. IEEE Intl Conf. Evolutionary Computation*, pp. 4661-4667, 2007.
- [۱۷] م. ح. سیگاری، و ک. لوکس، "کاربرد الگوریتم رقابت استعماری برای انتخاب ویژگی در سیستم تشخیص چهره،" قطب کنترل و پردازش هوشمند، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، ۱۳۸۶.
- [18] L. A Zadeh, "Fuzzy Sets," *Proc. IEEE Intl Conf. Information and Control*, pp. 338-346, 1965.
- [19] M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Book, New York, 1985.
- [20] J. Liebowitz, *the Handbook of Applied Expert Systems*, Prentice-Hall, 1999.
- [21] Li-Yeh Chuang, Sheng-Wei Tsai, and Cheng-Hong Yang, "Improved Catfish Particle Swarm Optimization with Fuzzy Adaptation," *Proc. IEEE Intl Conf. Electrical Engineering*, pp. 20-24, 2009.
- [22] International Biometrics Group, <http://www.atashpaz.com>, May 2003.
- [۲۳] ف. جان، محاسبات فازی، انتشارات دانش نگار، تهران، ۱۳۸۶.
- [24] P. J. Rousseeuw, A. M. Leroy, P. J. Rousseeuw, and A. M. Leroy, *Fuzzy Logic Analysis*, Wiley and Sons, 1987.
- [25] P. L. P. Wasantha, P. G. Ranjith, and D. R. Viete, "Constitutive Models Describing the Influence of the Geometry of Partially-spanning Joints on Jointed Rock Mass Strength: Regression and Fuzzy Logic Analysis of Experimental Data," *Journal of Expert Systems with Applications*, vol. 39, no. 3, pp. 7663-7672, 2012.

اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۲/۱۲/۲۳

تاریخ اصلاح: ۹۳/۸/۲۵

تاریخ قبول شدن: ۹۳/۹/۳

نویسنده مرتبط: محمد فیوضی، دانشکده مهندسی برق، دانشگاه علم و صنعت ایران، تهران، ایران.



محمد فیوضی در حال حاضر در دانشگاه علم و صنعت ایران، آزمایشگاه رباتیک و سیستم‌های غیر خطی (دکتر موسوی مشهدی)، دانشکده مهندسی برق مشغول به فعالیت می‌باشد. مدرک کارشناسی و کارشناسی ارشد خود را به ترتیب در سال‌های ۱۳۸۶ و ۱۳۹۱ از دانشگاه‌های آزاد

اسلامی واحد ساوه در رشته مهندسی برق، و دانشگاه حکیم سبزواری (تربیت معلم سبزوار) در رشته مهندسی برق بیوالکترونیک اخذ نموده است و در حال حاضر دانشجوی سال دوم مقطع دکتری مهندسی برق دانشگاه علم و صنعت ایران می‌باشند. زمینه‌های حرفه‌ای مورد علاقه ایشان، پردازش سیگنال (تصویرهای دیجیتالی و بیولوژیکی)، بازشناسی الگو، شبکه‌های عصبی مصنوعی و تطبیقی فازی عصبی، بهینه‌سازی، منطق فازی، محاسبات نرم، هوش مصنوعی کنترل مدل پیش بین و سیستم‌های غیر خطی می‌باشد. از ایشان تا به حال تعدادی مقاله در مجلات داخلی، خارجی و کنفرانس‌های مرتبط به چاپ رسیده است.

آدرس پست الکترونیکی ایشان عبارت است از:

mohammad_fiuzy@elec.iust.ac.ir



سید کمال‌الدین 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 استاد یاری در

دانشکده مهندسی برق مشغول به فعالیت می‌باشند. علایق مطالعاتی ایشان، کنترل خطی و غیر خطی، کنترل فازی، مدل سازی غیر خطی و کاربردهایش، آشوب، کنترل تطبیقی، کنترل فرآیندهای اتفاقی، کنترل چندمتغیره و محاسبات نرم می‌باشد.

آدرس پست الکترونیکی ایشان عبارت است از:

sk_mousavi@iust.ac.ir



جواد حدادنیا در حال حاضر دانشیار دانشکده مهندسی برق و کامپیوتر دانشگاه حکیم سبزواری (تربیت معلم سبزوار 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 دیجیتالی، 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 سردبیر مجله هوش مصنوعی در دانشگاه مربوطه می‌باشند، ایشان هم‌اکنون عضو دائم انجمن‌های بین‌المللی IEEE, SPIE, CIPPR, and IEICE می‌باشند.

آدرس پست الکترونیکی ایشان عبارت است از:

haddadnia@hsu.ac.ir

¹ Imperialist Competitive Algorithm

² Assimilation

³ Genetic Algorithm

⁴ Particle Swarm Optimization

⁵ Evolutionary Algorithm

⁶ John Holland

⁷ Russell Eberhart

⁸ John Kennedy

⁹ Simulated Annealing

¹⁰ Esmail Atashpaz-e-Gargari

¹¹ Caro Lucas

¹² Lotfi-e-Askar Zadeh

راهنمای تهیه مقالات

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 نمائید.
- 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 به‌مراه 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.

Improved ICA by Multivariable Regression Based on Fuzzy Logic

Mohammad Fiuzy¹ Seyed Kamaledine Mousavi Mashhadi¹ Javad Haddadnia²

¹Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.

²Electrical and Computer Engineering Faculty, Hakim Sabzevari University, Sabzevar, Iran.

ABSTRACT

Imperialist Competitive Algorithm¹ (ICA) is one of Optimization Algorithms based on random population which is inspired from idea evolution social-political of human being. In this algorithm based on information regarding best performance before (colonialized and colonizer countries) parameters of movement path and principle of unification (absorption) is arranged. Then number of colonialist countries along with colonized countries as movement of a population of answers will search in space of problem. In this method each possible answers (countries) try to converge towards best obtained experience (colonialist). Following this process optimum point of target function by an intelligent search is obtained. Regulation standard algorithm parameters has effective role in converging to optimum answer in solving complicated problems with vast range. Attempted with use of fuzzy logic to provide a method for control parameters of absorption coefficient and cost of colonization in order to increase accuracy, speed and correctness of ICA. Then by multivariable regression method, Fuzzy system in a non-linear relation binomial be briefed which of course will cause decrease of memory volume and calculation and increase of implementation speed. On this basis at first with use of expert knowledge and dominance on ICA Fuzzy system is designed. Then binomial non-linear relation with use of multivariable regression will be extracted. In every step of algorithm implementation, position of each country (space of problem answer) with attention to inputs of Fuzzy control (normalized process repetition and difference of optimum point global in each repetition) will be determined so that in obtaining next situation of countries, by relation non-linear binomial, absorption coefficient and cost of colonized countries is obtained. Proposed method in solving famous problems (Benchmark problems) with standard algorithms PSO², GA³ and ICA will be compared and results of comparison in order to approval advantage of proposed method by figures and tables will be provided.

Keywords: Optimization, Imperialist Competition, Local and Global Optimum, Fuzzy Logic, Multivariable Regression, Intelligent Search.

Designing an Online Face Recognition System using Graphics Processor

Mohsen Biglari¹

Faezeh Mirzaei¹

Hossein Ebrahimpour Komleh²

¹Department of Computer Engineering, Shahrood University of Technology, Shahrood, Iran

²Department of Computer Engineering, University of Kashan, Kashan, Iran

ABSTRACT

Face Recognition has attracted many researchers in the last three decades. There have been great improvements in controlled environments and static images. But there are still many challenges in uncontrolled environments and online applications. In this paper, a new parallel face recognition approach has been proposed that is almost robust to illumination and expression variations. Local binary pattern operator and SQL algorithm have been utilized for robust feature extraction and illumination variations elimination. To further improve performance, both of these algorithms have been extended. Moreover, System's speed has been increased using GPUs. In situations which using GPU was not optimal, parallelism has been done via CPU. The experimental results on two well-known databases, Yale B and CMU-PIE show our system's better performance over other compared methods. Some of these methods are QI, SQI, QIR, MQI, DMQI, DSFQI, PCA and LDA. The proposed parallel implementation resulted in speed-up more than 18 times in compare to unparallelled implementation. This novel parallel system can process a 325 by 205 pixels image in less than 213 milliseconds.

Keywords: Face Recognition, Graphics Processor (GPU), Local Binary Pattern (LBP), Illumination Variation, Expression Variation, Parallel Algorithms, SQL.

Probabilistic Büchi Automata of Records: Semantics of Coordinating Systems with Uncertain Channels

Hajar Abedi

Mohammad Izadi

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

Reo is a language for describing the relationships of the existing components of component based systems using a priori set of channels which is used to coordinate the concurrent processes in distributed and parallel environments. In a set of papers, already we have introduced a modeling formalism, called Büchi automaton of records (BAR), and its augmented version (ABAR) as the operational semantics of Reo with considering both synchronization and context dependency requirements. In this paper, we extend the notion of Büchi automaton of records by probabilistic parameters as the semantics of Reo circuits of channels with non-deterministic and probabilistic behaviors. These types of channels can be considered as unreliable channels in which there are some kinds of losing or corrupting of messages. In this way, we propose two probabilistic variants of Büchi automata of records. First, we propose simple probabilistic Büchi automata of records that apply probabilities just over the next configurations. Then, we introduce probabilistic Büchi automata of records in which for each state the next steps are determined by applying probabilities over the possible input and output operators. Also, we define the product operators over the both proposed models that are used as the semantics of channels joining in Reo. In addition, we define the operators of port-hiding for both models. The paper also contains some case studies and the comparison of expressive power of the proposed models with the other Reo probabilistic model namely Probabilistic Constraint Automata.

Keywords: Probabilistic Büchi Automata of Records, Reo Coordination Language, Component Based Systems, Büchi Automata of Records, Probabilistic Coordination Models.

A Preventive and Guidance Framework for Effective Establishment of Post-Disciplinary Education in Computing Science and Engineering in Iran

Seyed Ebrahim Abtahi

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

Transition to post-discipline education in the field of computing seems inevitable. However, this transition needs minimum knowledge and required skills of students to succeed. In this article, a framework for achieving this has been proposed. This proposal provides the possibility of advancement of cross-disciplinary knowledge and can be used as a measure of expanding such courses. This framework emphasizes minimum educational requirements of cross-disciplinary courses and recommends compensatory courses for new students. Elucidation of post-disciplinary after pre-disciplinary and disciplinary era and elaboration of post-disciplinary options, including multi-disciplinary, cross-disciplinary, inter-disciplinary, and trans-disciplinary educations are other subjects of this article. In regard with the high number of graduate students and inadequate facilities that affect the quality of education, this proposal can prevent the expanding low quality cross-disciplinary courses and make an opportunity from this threat.

Keywords: Curriculum Frameworks, Interdisciplinary Courses, Value Added Link Between Disciplines, Knowledge Areas of Computing, Course Chain Complementary, Skills of IT Professionals.

A Toolkit for Data Cloud Simulation with the Capability of Replica Management: An Extension to CloudSim

Saeedeh Mehri

Reza Entezari-Maleki

Ali Movaghar

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

Data replication is one of the most important data management mechanisms in distributed systems. Since performance evaluation of data replication and resource allocation policies in cloud computing is impractical because of limitations existing in real systems, simulation tools are widely used for this reason. CloudSim is a general framework for modeling, simulating and testing the application services and cloud computing infrastructures. However, there are some problems rising in simulating the data cloud in CloudSim such as: 1) disability in executing data-based requests and allocating an appropriate resource to them, 2) disability in handling data management and replication policies, and 3) lack of the ability of inquiring on number and location of replicas, costs, availability rate of data and so forth which are required in proposing new algorithms in data replication. On the other hand, there is no other simulation tool especially for simulating data cloud. Considering the above-mentioned problems, CloudSim is developed in this paper only with extending the existing entities without any change in its architecture layers. Therefore, a new tool for simulating data cloud called *Data CloudSim* is proposed in which it has the capability of extending replica management and data replication policies in a more informed way with the goal of improving quality of service factors and minimizing the cost.

Keywords: Data Cloud, Data Center, Data Replication, Data and Replica Management, CloudSim Simulator.

Studying and Characterizing Aspect-Oriented Programming Languages and Aspect Mining Techniques

Elahe Habibi

Abbas Heydarnoori

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ABSTRACT

Aspect-oriented software development (AOSD) is a new paradigm in complexity management through modularizing source code in large software systems. In particular, AOSD allows us to modularize the code for a functionality which is tangled and scattered across the program (i.e., crosscutting concerns) into modules. There are various tools and languages that support AOSD with their own advantages, disadvantages, syntaxes, and semantics. In this article, our goal is to introduce the basics of AOSD, and to study and to characterize different tools, techniques, and programming languages for this purpose. Additionally, we will survey and categorize aspect mining tools and techniques which are used in aspect identification in various software development phases.

Keywords: Aspect-Oriented Programming, Aspects, Aspect Mining, Crosscutting Concern, Programming Languages.

The CSI Journal on Computer Science and Engineering

Vol. 12

No. 3 (a)

2014

ABSTRACTS

- **Studying and Characterizing Aspect-Oriented Programming Languages and Aspect Mining Techniques** 1
Elahe Habibi and Abbas Heydarnoori
- **A Toolkit for Data Cloud Simulation with the Capability of Replica Management: An Extension to CloudSim** 2
Saeedeh Mehri, Reza Entezari-Maleki and Ali Movaghar
- **A Preventive and Guidance Framework for Effective Establishment of Post-Disciplinary Education in Computing Science and Engineering in Iran** 3
Seyed Ebrahim Abtahi
- **Probabilistic Büchi Automata of Records: Semantics of Coordinating Systems with Uncertain Channels** 4
Hajar Abedi and Mohammad Izadi
- **Designing an Online Face Recognition System using Graphics Processor** 5
Mohsen Biglari, Faezeh Mirzaei and Hossein Ebrahimpour Komleh
- **Improved ICA by Multivariable Regression Based on Fuzzy Logic** 6
Mohammad Fiuzy, Seyed Kamaledine Mousavi Mashhadi and Javad Haddadnia

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, 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. Asadnia (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.