

دانشگاه صنعتی شاهرود

دانشکده‌ی مهندسی کامپیوتر و فن‌آوری اطلاعات
رشته‌ی مهندسی کامپیوتر، گرایش هوش مصنوعی

رساله‌ی دکتری

یک مدل آماری فرآیند-آگاه جهت تعدیل بارکاری بر پایه‌ی یادگیری تقویتی

نگارنده: مهدی یعقوبی

استاد راهنما

دکتر مرتضی زاهدی

استاد مشاور

دکتر علیرضا احمدی‌فرد

شهریور ۱۳۹۶

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

دانشگاه صنعتی شاهرود

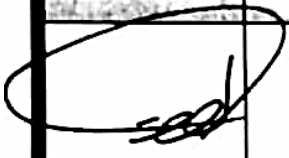
دانشکده : مهندسی کامپیوتر و فن آوری اطلاعات

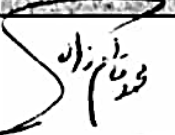
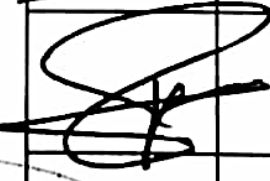
گروه : هوش مصنوعی

رساله دکتری آقای مهدی یعقوبی

تحت عنوان: یک مدل آماری فرآیند-آگاه جهت تعدیل بارکاری بر پایه‌ی یادگیری تقویتی

در تاریخ ۱۳۹۶/۰۶/۰۸ توسط کمیته تخصصی زیر جهت اخذ مدرک دکتری ارزیابی گردید و با درجه عالی مورد پذیرش قرار گرفت.

امضاء	اساتید مشاور	امضاء	اساتید راهنما
	نام و نام خانوادگی: دکتر علیرضا احمدی فرد		نام و نام خانوادگی: مرتضی زاهدی

امضاء	نماینده تحصیلات تکمیلی	امضاء	اساتید داور
	نام و نام خانوادگی: دکتر محسن رضوانی		نام و نام خانوادگی: دکتر محمد قاسم زاده
			نام و نام خانوادگی: دکتر سید علی سلیمانی
			نام و نام خانوادگی: دکتر علی اکبر پویان

تقدیم به پدرم، مادرم و همسر مهربانم

تقدیر و تشکر

با تشکر و قدردانی از جناب آقای دکتر مرتضی زاهدی که راهنمایی‌ها و تشویق‌های ایشان بود که توانستم این رساله را به پایان برسانم و با کمال تشکر از جناب آقای دکتر علیرضا احمدی‌فرد که با سوالات هوشمندانه، اشاره به نکات ظریف و راهنمایی‌های دلسوزانه در تکمیل این رساله مرا یاری نمودند.

با تشکر و قدردانی از داوران جلسه‌ی دفاع از پیشنهاد رساله، جناب آقای دکتر علی اکبر پویان، جناب آقای دکتر سید علی سلیمانی و جناب آقای دکتر محمد قاسم‌زاده که نظرات و پیشنهادهای ایشان ابهامات موجود در رساله را برایم روشن کرد.

با تشکر فراوان از جناب آقای دکتر حمید حسن‌پور که در دوره‌ی تحصیل اینجانب به عنوان ریاست دانشکده مرا در پیش‌برد امور آموزشی و پژوهشی یاری نمودند.

با تشکر از دوستان عزیز آزمایشگاه سرکار خانم مهندس مرضیه رحیمی، جناب آقایان مهندس مهدی حسینی و مهندس ایمان فیروزیان که در جلسات و بحث‌های انجام شده در آزمایشگاه راهگشای مسائل مطرح شده در این رساله بودند.

تعهدنامه

اینجانب مهدی یعقوبی دانشجوی دوره دکتری رشته مهندسی کامپیوتر گرایش هوش مصنوعی دانشکده کامپیوتر و فناوری اطلاعات دانشگاه صنعتی شاهرود نویسنده رساله دکتری با عنوان یک مدل آماری فرآیند-آگاه جهت تعدیل بارکاری بر پایه یادگیری تقویتی تحت راهنمایی جناب آقای دکتر مرتضی زاهدی متعهد می شوم.

- تحقیقات در این رساله توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در رساله تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی رساله تأثیرگذار بوده اند در مقالات مستخرج از رساله رعایت می گردد.
- در کلیه مراحل انجام این رساله، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این رساله، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ ۹۶/۶/۸
امضای دانشجو
مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم افزارها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در رساله بدون ذکر مرجع مجاز نمی باشد.

چکیده

مدیریت منابع انسانی یکی از مهم‌ترین اهداف سیستم‌های مدیریت فرآیندهای سازمانی (BPMS) است که در جهت بهبود کیفیت محصولات و سرویس‌دهی بهتر خدمات تلاش می‌کند. برقراری تعادل بارکاری منابع انسانی و ایجاد یکنواختی در بارکاری هر منبع، در اجرای فرآیندهای سازمانی باعث بهبود عملکرد سیستم‌های مدیریت فرآیندهای سازمانی می‌شود. برای مدل‌سازی اجرای فرآیندهای سازمانی از تکنیک‌ها و روش‌های مختلفی استفاده شده است، از آن قبیل می‌توان فرآیندهای تصمیم‌گیری مارکوف (MDPs)، شبکه‌ی صف (QNM) و شبکه‌ی جریان کاری (WF-net) را نام برد. در این رساله یک مدل آماری فرآیند-آگاه جهت مدل‌سازی اجرای فرآیندهای سازمانی تحت عنوان فرآیندهای تصمیم‌گیری توسعه‌یافته‌ی مارکوف (XMDPs) ارائه می‌شود. در این مدل نگاشت بهتری از اجرای فرآیندهای سازمان نسبت به مدل MDPs فراهم شده است.

ایده‌ی اصلی در توسعه مدل MDPs، پشتیبانی بیش‌تر از رویدادهای موجود در فرآیندهای سازمانی است. مسئله‌ی برقراری تعادل بارکاری و ایجاد یکنواختی در بارکاری منابع به‌عنوان هدف بهینه‌سازی در نگاشت فرآیندهای سازمانی بر مدل پیشنهادی در نظر گرفته می‌شود و این مسئله با استفاده از یادگیری تقویتی حل می‌گردد تا به کمک آن تخصیص بهتری از منابع در جهت حصول اهداف تعریف شده ایجاد شود. روش پیشنهادی با یک روش تخصیص منابع بر اساس یادگیری تقویتی (RLRAM) که بر پایه‌ی مدل MDPs ارائه شده است روی پایگاه داده‌ی BPI Challenge 2102 مقایسه می‌شود. آزمایش‌های انجام شده نشان می‌دهد با یک نگاشت مناسب از اجرای فرآیندهای سازمانی بر مدل پیشنهادی می‌توان مسئله‌ی برقراری تعادل بارکاری و ایجاد یکنواختی بارکاری منابع را ۱۲/۷ درصد بهتر مدیریت کرد.

در این رساله علاوه بر ارائه‌ی مدل XMDPs، با استفاده از یک روش آماری، نوعی وابستگی در اجرای هم‌زمان وظایف از سابقه‌ی رویدادها استخراج می‌شود و چند الگوریتم برای استفاده از این وابستگی در جهت بهبود تخصیص منابع ارائه می‌گردد. الگوریتم‌های ارائه شده نیز با سایر الگوریتم‌های موجود در پیشینه‌ی پژوهش مقایسه می‌شوند. نتایج مشاهده شده اثر مطلوب اجرای هم‌زمان وظایف را در تخصیص منابع نشان می‌دهد. در ادامه‌ی تحقیقات انجام شده در این رساله مسئله‌ی تنظیم هم‌روندی فرآیندهای سازمانی با هدف تعادل و یکنواختی بارکاری معرفی می‌شود. حل این مسئله، با سه روش مختلف مورد مطالعه و بررسی قرار می‌گیرد. تنظیم هم‌روندی فرآیندها روی یک پایگاه داده‌ی واقعی آزمایش می‌شود. آزمایش انجام شده، بهبود ۲۱/۶۴ درصدی را در برقراری تعادل و یکنواختی بارکاری منابع نشان می‌دهد.

کلمات کلیدی

سیستم مدیریت فرآیندهای سازمانی، تعادل بارکاری، یکنواختی بارکاری، مدل‌سازی فرآیندهای سازمانی، مدل آماری فرآیند-آگاه، وابستگی وظایف، تنظیم هم‌روندی فرآیندهای سازمانی.

مقالات پذیرفته شده در نشریات علمی پژوهشی

- مهدی یعقوبی، مرتضی زاهدی، ایجاد تعادل بارکاری و کاهش چرخه‌ی زمانی با استفاده از اجرای هم‌زمان وظایف وابسته در BPMS، *مجله‌ی مهندسی برق دانشگاه تبریز*، پذیرش قطعی در دی ماه ۱۳۹۵، در نوبت چاپ.
- مهدی یعقوبی، مرتضی زاهدی، بهینه‌سازی در هم‌روندی فرآیندهای کسب‌وکار با هدف تعادل بارکاری، *مجله‌ی مدل‌سازی در مهندسی دانشگاه سمنان*، پذیرش قطعی در خرداد ماه ۱۳۹۶، در نوبت چاپ.
- مهدی یعقوبی، مرتضی زاهدی، علیرضا احمدی‌فرد، ارائه‌ی الگوریتم پویا برای تنظیم هم‌روندی فرآیندهای کسب‌وکار، *فصل‌نامه پردازش علائم و داده‌ها*، پذیرش قطعی در مرداد ماه ۱۳۹۶، در نوبت چاپ.

مقالات ارائه شده در کنفرانس‌ها

- Resource allocation using task similarity distance in business process management systems, *International Conference of Signal Processing and Intelligent Systems (ICSPIS)*, Amirkabir university of technology, March 2017
- بهینه‌سازی در هم‌روندی فرآیندها جهت تعدیل بارکاری منابع با الگوریتم PSO، دومین کنفرانس پردازش سیگنال و سیستم‌های هوشمند، دانشکده برق دانشگاه صنعتی امیر کبیر، آذر ماه ۱۳۹۵
- کاهش چرخه‌ی زمانی با استفاده از تخصیص هم‌زمان وظایف وابسته، دومین کنفرانس پردازش سیگنال و سیستم‌های هوشمند، دانشکده برق دانشگاه صنعتی امیر کبیر، آذر ماه ۱۳۹۵

فهرست مطالب

فصل ۱- مقدمه	۱.....
۱-۱ سیر تحول سیستم‌های اطلاعاتی تا BPMS	۳.....
۲-۱ ادبیات موضوع و مفاهیم	۴.....
۱-۲-۱ درخواست، پرونده و مورد	۴.....
۲-۲-۱ منابع در فرآیندهای سازمانی	۵.....
۳-۲-۱ وظیفه، واحد کاری و فعالیت	۷.....
۴-۲-۱ سابقه‌ی رویدادها	۸.....
۵-۲-۱ فرآیند سازمانی	۱۱.....
۶-۲-۱ بارکاری منابع	۱۲.....
۷-۲-۱ تعادل بارکاری	۱۳.....
۸-۲-۱ یکنواختی بارکاری یک منبع	۱۴.....
۳-۱ طرح مسئله	۱۵.....
۴-۱ هدف مسئله	۱۸.....
۵-۱ فرضیه‌ها	۱۹.....
۶-۱ چالش‌ها	۱۹.....
فصل ۲- پیشینه‌ی پژوهش	۲۱.....
۱-۲ از دیدگاه مدل	۲۲.....
۱-۱-۲ مدل فرآیندهای تصمیم‌گیری مارکوف (MDPs)	۲۳.....
۲-۱-۲ مدل شبکه‌ی صف	۲۹.....
۳-۱-۲ مدل WF-Net	۳۰.....
۲-۲ از دیدگاه هدف	۳۶.....
۱-۲-۲ کاهش چرخه‌ی زمانی	۳۶.....
۲-۲-۲ تعادل بارکاری منابع	۴۱.....
فصل ۳- توسعه‌ی مدل MDPS و ارائه‌ی مدل پیشنهادی	۴۵.....
۱-۳ مدل پیشنهادی XMDPS	۴۶.....
۲-۳ حل مسئله‌ی فرآیندهای تصمیم‌گیری مارکوف با مدل XMDPS	۴۸.....
۳-۳ استخراج مدل XMDPS از مدل فرآیند	۵۰.....
۱-۳-۳ استخراج دنباله‌ی حالات مدل فرآیند	۵۱.....
۲-۳-۳ استخراج مدل XMDPS از مجموعه‌ی دنباله‌ی حالات	۵۳.....
۴-۳ آزمایش مدل پیشنهادی XMDPS	۵۴.....
۱-۴-۳ مشاهده‌ی اثر افزایش بازه‌ی یکنواختی در بهبود یادگیری	۶۰.....
۵-۳ جمع‌بندی و نتیجه‌گیری	۶۴.....

فصل ۴- کاهش چرخه‌ی زمانی در تخصیص متعادل بارکاری	۶۵
۱-۴ مفاهیم و تعاریف	۶۷
۱-۱-۴ مدل تحلیلی فرآیند	۶۷
۲-۱-۴ رویداد ذخیره شده	۶۹
۳-۱-۴ استخراج اطلاعات فعالیت‌ها از سابقه‌ی رویداد	۶۹
۲-۴ تخمین زمان اجرای وظایف	۷۰
۱-۲-۴ تخمین زمان اجرای وظایف مستقل	۷۰
۲-۲-۴ تخمین زمان اجرای وظایف وابسته	۷۰
۳-۲-۴ فاصله‌ی وابستگی وظایف	۷۲
۳-۴ تعادل بارکاری و هم‌زمان‌سازی وظایف وابسته	۷۳
۱-۳-۴ بهبود الگوریتم هم‌زمان‌سازی وظایف وابسته	۷۵
۴-۴ آزمایش‌ها و نتایج	۷۷
۱-۴-۴ آزمایش اول: فرآیند مالی	۷۹
۲-۴-۴ آزمایش دوم: فرآیند تعیین اعتبار	۸۱
۳-۴-۴ آزمایش سوم: فرآیند تولید	۸۳
۴-۴-۴ آزمایش چهارم: مقایسه‌ی الگوریتم بهبود یافته	۸۵
۵-۴ جمع‌بندی و نتیجه‌گیری	۸۷
۶-۴ پیشنهادات	۸۸
فصل ۵- بهینه‌سازی در تنظیم هم‌روندی چند فرآیند	۸۹
۱-۵ مفاهیم و تعاریف	۹۱
۱-۱-۵ مجموعه‌ی فرآیندها با منابع مشترک	۹۱
۲-۱-۵ الگوی بارکاری منابع در طول زمان	۹۱
۲-۵ بهینه‌سازی در تنظیم هم‌روندی فرآیندها	۹۲
۳-۵ حل مسئله‌ی تنظیم هم‌روندی	۹۴
۱-۳-۵ جستجوی فضای حالت	۹۴
۲-۳-۵ بهینه‌سازی پویا	۹۵
۳-۳-۵ الگوریتم تکاملی PSO	۹۸
۴-۵ آزمایش و نتایج	۱۰۰
۵-۵ مطالعه‌ی موردی	۱۰۴
۶-۵ نتیجه‌گیری	۱۰۷
۷-۵ پیشنهادات	۱۰۸
منابع و مراجع	۱۰۹
چکیده‌ی انگلیسی	۱۱۳

فهرست شکل‌ها

- شکل (۱-۱) تحول سیستم‌های اطلاعاتی ۴
- شکل (۲-۱) دسته‌بندی منابع انسانی بر اساس جایگاه و پُست سازمانی. منبع مشخص شده با دایره‌ی قرمز رنگ دارای جایگاه‌ی سازمانی در گروه آمار در دانشکده‌ی علوم است و دارای پُست یا نقش سازمانی عضو هیئت علمی است..... ۵
- شکل (۳-۱) ارتباط مفاهیم وظیفه، واحد کاری و فعالیت ۷
- شکل (۴-۱) وابستگی مفاهیم در تکمیل مفهوم فعالیت ۸
- شکل (۵-۱) چرخه‌ی حیات واحد کاری در کنار وضعیت‌های درخواست ۹
- شکل (۶-۱) فرا مدل قالب XES [۲]..... ۱۰
- شکل (۷-۱) بخشی از سابقه‌ی رویداد فرآیند موسسه‌ی مالی هلند که در پایگاه داده‌ی BPI Challenge 2012 موجود است در قالب فایل XES..... ۱۱
- شکل (۸-۱) قاعده‌ی Yerkes-Dodson [۷] ۱۴
- شکل (۹-۱) وضعیت بارکاری قبل و بعد از اجرای روش مورد انتظار در بازه‌ی زمانی مشخص ۱۶
- شکل (۱۰-۱) یکنواختی بارکاری یک منبع قبل و بعد از اجرای روش مورد انتظار در بازه‌ی زمانی مشخص ۱۶
- شکل (۱۱-۱) بارکاری تعدادی از منابع در پایگاه داده‌ی BPI Challenge 2012..... ۱۷
- شکل (۱۲-۱) نمایش نقطه‌ای بارکاری منابع در پایگاه داده‌ی BPI Challenge 2012. نقاط تیره‌تر بیانگر بارکاری بیش‌تر است..... ۱۸
- شکل (۱۳-۱) نمایش نقطه‌ای انحراف بارکاری (غیرصفر) از میانگین بارکاری منابع در پایگاه داده‌ی BPI Challenge 2012..... ۱۸
- شکل (۱-۲) درصد موضوع تحقیقات انجام شده در BPMS [۸] ۲۲
- شکل (۲-۲) فرآیند نیمه‌خودکار یادگیری تقویتی با نظارت برای تخصیص منبع [۹] ۲۴
- شکل (۳-۲) فرآیند Radiology CT-scan [۱۰] ۲۵
- شکل (۴-۲) مثالی از مدل فرآیندهای تصمیم‌گیری مارکوف ۲۵
- شکل (۵-۲) نمایی کلی از روش RLRAM..... ۲۸
- شکل (۶-۲) صف تک سرویس‌دهنده ۲۹
- شکل (۷-۲) یک مثال از محاسبه‌ی متوسط نرخ ورود درخواست در هر یک از وظایف در شبکه‌ی صف [۴] ۳۰
- شکل (۸-۲) استفاده از یک زیرفرآیند در فرآیند ۳۱
- شکل (۹-۲) نگاشت مفاهیم فرآیند در WF-net [۱۲] ۳۲
- شکل (۱۰-۲) یک انتقال حلقوی جهت اثبات درستی WF-Net ۳۴

شکل (۱۱-۲) نتایج الگوریتم یادگیری تقویتی با در نظر گرفتن ارتباط اجتماعی برای ۲۰۰ بار اجرا	۳۸
[۲۷]	۳۹
شکل (۱۲-۲) مقایسه‌ی چرخه‌ی زمانی و میزان تعادل بارکاری سه الگوریتم پایه [۳]	۴۱
شکل (۱۳-۲) تخصیص بهتر علاوه بر ایجاد تعادل در بارکاری باعث کاهش چرخه‌ی زمانی شده است	[۳]
شکل (۱-۳) مدل پیشنهادی XMDPs و عملکرد عنصر trigger	۴۸
شکل (۲-۳) الگوریتم یادگیری تقویتی برای مدل پیشنهادی XMDPs	۴۹
شکل (۳-۳) روش استفاده از مدل پیشنهادی در تخصیص منابع	۵۵
شکل (۴-۳) تعداد درخواست‌های ورودی به تفکیک روز در پایگاه داده‌ی BPI Challenge 2012 ...	۵۶
شکل (۵-۳) تعداد درخواست‌ها به تفکیک طول چرخه‌ی زمانی در پایگاه داده‌ی BPI Challenge 2012	۵۶
شکل (۶-۳) داده‌های آموزش و آزمایش مدل پیشنهادی XMDPs	۵۶
شکل (۷-۳) فرآیند موسسه‌ی مالی هلند [۲۷]	۵۷
شکل (۸-۳) نتایج مقایسه‌ی الگوریتم یادگیری تقویتی بر اساس مدل‌های (RLRAM)MDPs و XMDPs با الگوریتم پایه‌ی تخصیص بارکاری به منبع با کم‌ترین بارکاری (MinWorkload)	
در بهینه‌سازی برقراری تعادل بارکاری و یکنواختی بارکاری منابع روی پایگاه داده‌ی BPI Challenge 2012 در مقایسه با الگوریتم تخصیص استفاده شده در پایگاه داده (DataSet)	
شکل (۹-۳) میزان کاهش $f(s)$ در الگوریتم QL+XMDPs نسبت به الگوریتم RLRAM	۵۹
شکل (۱۰-۳) مجموع روزانه‌ی تعداد درخواست‌ها از ابتدای دوره در پایگاه داده‌ی BPI Challenge 2012	۶۰
شکل (۱۱-۳) تعداد درخواست‌ها به تفکیک روز در پایگاه داده‌ی BPI Challenge 2012	۶۱
شکل (۱۲-۳) واریانس تعداد درخواست‌ها برای بازه‌های یکنواختی متفاوت	۶۱
شکل (۱۳-۳) تغییر در تابع محاسبه‌ی بارکاری و اثر آن در یکنواختی و تعادل بارکاری منابع	۶۳
شکل (۱-۴) چرخه‌ی حیات یک واحد کاری	۶۸
شکل (۲-۴) مراحل اجرای الگوریتم بهبود یافته هم‌زمان‌سازی وظایف وابسته	۷۶
شکل (۴-۴) فرآیند موسسه‌ی مالی هلند [۲۷]	۷۹
شکل (۵-۴) نتایج آزمایش اول روی فرآیند واقعی موسسه‌ی مالی هلند	۸۰
شکل (۶-۴) فرآیند تعیین اعتبار [۴]	۸۱
شکل (۷-۴) نتایج آزمایش دوم روی فرآیند تعیین اعتبار	۸۳
شکل (۸-۴) فرآیند تولید [۶]	۸۳
شکل (۹-۴) نتایج آزمایش سوم روی فرآیند تولید	۸۴

- شکل (۴-۱۰) مقایسه‌ی الگوریتم تکمیلی بهبود یافته‌ی CPDT2 با سایر الگوریتم‌ها..... ۸۶
- شکل (۴-۱۱) درصد کاهش چرخه‌ی زمانی الگوریتم CPDT2 نسبت به CPDT و الگوریتم CPDT2 نسبت به DSWB و الگوریتم CPDT نسبت به DSWB..... ۸۶
- شکل (۴-۱۲) درصد حداقل و حداکثر بارکاری منابع برای دو الگوریتم پیشنهادی (CPDT و CPDT2) در مقایسه با الگوریتم DSWB..... ۸۷
- شکل (۵-۱) پنج فرآیند کسب‌وکار و نمودار بارکاری آن‌ها..... ۱۰۰
- شکل (۵-۲) نتیجه‌ی تنظیم هم‌روندی فرآیندها با استفاده از الگوریتم‌های پیشنهاد شده..... ۱۰۱
- شکل (۵-۳) مقادیر LBFAT فرآیندهای هم‌روند در تکرارهای الگوریتم بهینه‌ساز PSO..... ۱۰۳
- شکل (۵-۴) تعداد دانشجویان انتخاب واحد کرده، ثبت نامی و فارغ‌التحصیل در پایگاه داده‌ی موسسه‌ی آموزش عالی میرداماد..... ۱۰۴
- شکل (۵-۵) میانگین بارکاری منابع انسانی در ابتدای نیم‌سال تحصیلی..... ۱۰۵
- شکل (۵-۶) مجموع بارکاری منابع انسانی در سه فرآیند..... ۱۰۶

فهرست جدول‌ها

جدول (۱-۲) انواع مسیریابی پرونده در فرآیندها [۱۲].....	۳۳
جدول (۲-۲) مقایسه‌ی ابزارهای مدل‌سازی فرآیند.....	۳۵
جدول (۱-۳) یک مثال از استخراج دنباله‌ی حالات مدل فرآیند.....	۵۲
جدول (۲-۳) تقسیم‌بندی پایگاه داده‌ی BPI Challeng 2012 جهت آزمایش مدل پیشنهادی.....	۵۷
جدول (۱-۴) الگوریتم‌های مقایسه شده.....	۷۸
جدول (۲-۴) مقادیر فاصله‌ی وابستگی محاسبه شده برای منبع با شناسه‌ی ۱۱۱۲۱.....	۸۰
جدول (۳-۴) میانگین زمان سرویس‌دهی هر منبع مسئول برای هر وظیفه‌ی مشخص.....	۸۱
جدول (۴-۴) احتمال تخصیص وظیفه به منابع برای ایجاد تعادل بارکاری با نرخ ورود ۱/۱۲.....	۸۲
جدول (۵-۴) میانگین زمان سرویس‌دهی هر منبع مسئول برای هر وظیفه‌ی مشخص در فرآیند تولید [۴].....	۸۴
جدول (۶-۴) احتمال تخصیص وظیفه به منابع برای ایجاد تعادل بارکاری با نرخ ورود ۱/۱۳.....	۸۴
جدول (۷-۴) میزان کاهش الگوریتم پیشنهادی CPDT.....	۸۵
جدول (۱-۵) کاهش LBFAT و تعیین پارامتر Δt در بارکاری فرآیندهای هم‌رند با استفاده از الگوریتم بهینه‌سازی PSO.....	۱۰۲
جدول (۲-۵) پیچیدگی زمانی و زمان اجرای الگوریتم‌های ارائه شده ($P = 5, R_s = 5, N = 211$).....	۱۰۳

فهرست اختصارات

B	BPEL	Business Process Execution Language
	BPM	Business Process Management
	BPMN	Business Process Model and Notation
	BPMS	Business Process Management System
	BSR	Bayesian Selection Rule
C	CPDT	Concurrent Performing of Dependent Tasks
	CRM	Customer Relationship Management
	DSWB	Dynamic Stochastic task allocation approach for Workload Balancing
E	EDD	Earliest Due Date
	EPCs	Event-driven Process Chains
	ERP	Enterprise Resource Planning
	EWf-net	Extended Workflow-net
F	FIFO	First In First Out
H	HWR	High Workload Resources
I	ICA	Imperialist Competitive Algorithm
L	LBFAT	Load Balancing Factor Along Time
	LWR	Low Workload Resources
M	MDPs	Markov Decision Processes
	MST	Minimum Slack Time
N	NBSR	Naïve Bayes Selection Rule
P	PAIS	Process Aware Information System
	PSO	Particle Swarm Optimization
	PTG	Probabilistic Timed Graph
Q	QNM	Queuing Network Model
	QSR	Q-learning for task allocation based on Social Relation
R	RDBMS	Relational Database Management System
	RL	Reinforcement Learning
	RLRAM	Reinforcement Learning Resource Allocation Mechanism
S	SPT	Shortest Processing Time
	SR	Social Relation
	SWN	Stochastic Workflow Net
U	UIMS	User Interface Management System
	ULD	Uniformity of workload
V	VLD	Variance of workload
W	WB	Workload Balancing
	WFM	Workflow Management
	WFMS	Workflow Management Systems
	WF-net	Workflow Net

X	XES	Extensible Event Stream
	XMDPs	Extended Markov Decision Processes
Y	YAWL	Yet Another Workflow Language

فصل ۱ – مقدمه

سیر تحول سیستم‌های اطلاعاتی

تعریف مفاهیم درخواست، منابع، وظیفه، واحد کاری و فعالیت

تعریف مدل تحلیلی فرآیند

تعریف بار کاری، تعادل بار کاری و یکنواختی بار کاری

مدیریت فرآیندهای سازمانی (BPM^۱) یک چارچوب خوش تعریف از ترکیب نرم افزار، تکنولوژی اطلاعات و دانش مدیریت است [۱ و ۲]. BPM در شرکت ها و سازمان های بزرگ به صورت بالقوه امکان افزایش بهره وری و صرفه جویی هزینه ها را در فرآیندهای سازمانی به وجود می آورد، لذا BPM امروزه توجه بسیاری از محققان را به خود جلب کرده است. BPM یک سیستم اطلاعاتی است که امکان طراحی، اجرا، مدیریت و آنالیز فرآیندهای سازمانی را فراهم می کنند. BPM را می توان یک توسعه ی هوشمندانه از مدیریت جریان های کاری (WFM^۲) دانست. WFM فرآیندها سازمانی را خودکارسازی می کند. درحالی که BPM علاوه بر خودکارسازی و اجرای فرآیند، به تحلیل، آنالیز و بهبود فرآیندهای سازمانی می پردازد. به عنوان مثال با مدل سازی و شبیه سازی یک فرآیند سازمانی امکان تحلیل و آنالیز وضعیت فرآیند به صورت آماری در طول زمان به وجود می آید و می توان هزینه ی تولید یا کیفیت محصول یا کیفیت سرویس دهی را بهبود بخشید. BPM نرم افزاری است که به کمک آن می توان بر وضعیت اجرای فرآیندهای سازمانی نظارت جامع، کامل و هوشمندانه ای داشت. در حقیقت BPM با بهره گیری از روش های آماری، الگوریتم های داده کاوی و فرآیند کاوی، اطلاعات و دانش نهفته در سابقه ی رویدادها را کشف می کند و از آن در جهت بهبود و تغییر سیاست های سازمان استفاده می کند. سیستم های اطلاعاتی فرآیند-آگاه (PAIS^۳) سیستم های سنتی WFM را با روش های نوین BPM ترکیب می کنند و انعطاف پذیری بیش تری را برای کاربران به موجود می آورد و طراحی فرآیندهای خاص را امکان پذیر می کنند. به عنوان نمونه می توان سیستم های بزرگ ERP^۴ (SAP و Oracle)، سیستم های CRM^۵، سیستم های مدیریت پرونده ها و سیستم های بر پایه ی قانون را نام برد.

^۱ Business Process Management

^۴ Enterprise Resource Planning

^۲ Workflow Management

^۵ Customer Relationship Management

^۳ Process Aware Information System

۱-۱ سیر تحول سیستم‌های اطلاعاتی تا BPMS

در دهه‌ی ۱۹۶۰ میلادی برنامه‌های کاربردی فقط با سیستم عامل در ارتباط بودند و برنامه‌نویسان و توسعه‌دهندگان برنامه‌های کاربردی برای طراحی و پیاده‌سازی یک سیستم اطلاعاتی کوچک مجبور بودند به طور مستقیم برای کار با فایل‌ها، برنامه‌نویسی کنند. در این دهه برنامه‌نویسان برای ذخیره و بازیابی اطلاعات باید از مجموعه‌ی کتابخانه‌های تعریف شده در سیستم عامل استفاده می‌کردند. با ظهور سیستم‌های مدیریت پایگاه داده‌های رابطه‌ای (^۱RDBMS) در اوایل دهه‌ی ۱۹۷۰ در پیاده‌سازی سیستم‌های اطلاعاتی انقلابی اساسی به وجود آمد به‌طوری‌که توسعه‌دهندگان دیگر برای ذخیره‌سازی، ویرایش، به‌روزرسانی و جستجوی اطلاعات نیازی به استفاده مستقیم از روش‌های دسترسی به فایل نداشتند و مدیریت ذخیره و بازیابی اطلاعات به‌طور کامل بر عهده سیستم مدیریت پایگاه داده‌ها بود. در این دهه برنامه‌نویسان برای کار با داده‌ها فقط کافی بود به زبان SQL مسلط شوند. ظهور سیستم‌های مدیریت واسط کاربری (^۲UIMS) در دهه‌ی ۱۹۸۰ میلادی در توسعه‌دهندگان این توانایی را ایجاد کرد که بخش مدیریت واسط کاربری را خارج از برنامه کاربردی قرار دهند. در دهه‌ی ۱۹۹۰ یک بخش دیگر با عنوان سیستم مدیریت جریان‌های کاری از برنامه‌های کاربردی جدا شد. هدف اصلی سیستم مدیریت جریان‌های کاری، تعریف، اجرا، کنترل و ثبت فرآیندهای کسب‌وکار (سازمانی) بود.

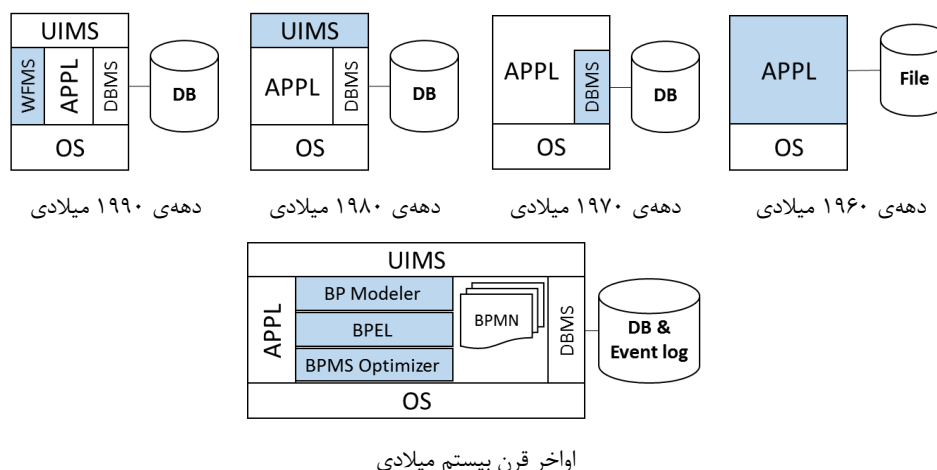
در اواخر قرن بیستم میلادی WFMS جای خود را به BPMS داد که حداقل شامل زیرسیستم‌های مدل‌سازی فرآیندهای سازمانی^۳، موتور فرآیند یا زبان اجرای فرآیندهای سازمانی^۴ و مهم‌ترین بخش، بهینه‌ساز سیستم است. در بهینه‌ساز عملکرد گذشته‌ی سیستم مورد بررسی قرار می‌گیرد و بر اساس تحلیل و آنالیز آماری که روی آن انجام می‌شود؛ تغییراتی در مدل‌سازی یا اجرای فرآیند سازمانی در جهت بهبود عملکرد سیستم انجام می‌شود. شکل (۱-۱) نمایشی از تحول سیستم‌های اطلاعاتی را طی دهه‌های ۱۹۶۰ الی ۲۰۱۰ نشان می‌دهد.

^۱ Relational Database Management System

^۲ User Interface Management System

^۳ Business Process Modeler

^۴ Business Process Execution language



شکل (۱-۱) تحول سیستم‌های اطلاعاتی

۲-۱ ادبیات موضوع و مفاهیم

در این بخش، با مجموعه‌ای از مفاهیم اساسی در BPMS آشنا می‌شویم. این مفاهیم عبارت‌اند از: درخواست، پرونده، مورد، منبع، وظیفه، واحد کاری، فعالیت، رویداد و سابقه‌ی رویداد؛ که در ادامه آن‌ها را شرح خواهیم داد.

۱-۲-۱ درخواست، پرونده و مورد^۱

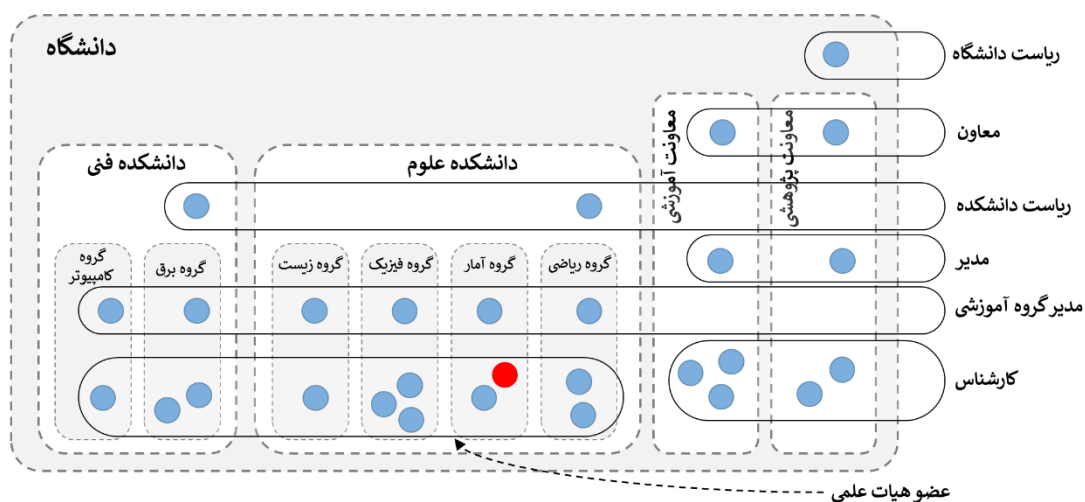
هدف BPMS سرویسی‌دهی به درخواست‌های ورودی یک سازمان است یا مدیریت و نظارت بر فرآیند تولید یک محصول است. در حین مدیریت و نظارت بر تولید یک محصول، اطلاعات عملیات و اقداماتی که روی محصول انجام می‌شود در قالب پرونده‌ی درخواست جمع‌آوری و ثبت می‌گردد. به بیان دیگر، هدف اصلی BPMS اداره کردن پرونده‌ها است. به‌عنوان مثال یک پرونده می‌تواند اطلاعات یک بیمار در بیمارستان باشد یا اطلاعات تحصیلی یک دانشجو در دانشگاه یا موارد مشابه در سایر سیستم‌ها اطلاعاتی باشد. هر پرونده در سیستم با یک شناسه منحصر به فرد مشخص می‌شود و این شناسه امکان ارجاع به پرونده را در دوره‌ی حیاتش فراهم می‌کند. پرونده در نقطه شروع فرآیند ایجاد می‌شود و با هر فعالیتی که روی آن انجام می‌شود در مسیر مشخص شده‌ای در فرآیند حرکت می‌کند تا رسیدن پرونده به حالت پایانی، به ترتیب فعالیت‌های تعریف شده در فرآیند روی آن انجام می‌شود. به مجموعه‌ی اطلاعات درخواست ورودی و اطلاعات ثبت شده در پرونده‌ی آن «مورد» گفته می‌شود.

^۱ Case

۲-۲-۱ منابع در فرآیندهای سازمانی

به‌طور کلی، در فرآیند سازمانی برای انجام فعالیت‌های تعریف شده از منابع انسانی، منابع سخت-افزاری و منابع نرم‌افزاری استفاده می‌شود. برخی منابع مصرفی هستند مانند کاغذ چاپگر که تمام می‌شوند و برخی این‌گونه نیستند. برای انجام برخی از فعالیت‌ها نیاز به تخصص و تجربه‌ی یک نیروی انسانی است و برخی، برای اجرا صرفاً به یک سخت‌افزار خاص نیاز دارند. بنابراین برای تکمیل فعالیت‌ها در فرآیند سازمانی به مجموعه‌ی متنوعی از منابع نیاز است. در این رساله توجه تحقیق روی منابع انسانی است.

منابع انسانی در فرآیندهای سازمانی به دو صورت کلی تقسیم‌بندی می‌شوند؛ جایگاه سازمانی و پُست یا نقش سازمانی. وظایف تعریف شده در فرآیندها با توجه به نقش و جایگاه سازمانی منابع به آن‌ها تخصیص داده می‌شوند. به‌عنوان مثال در یک سیستم دانشگاهی که متشکل از مجموعه‌ای از دانشکده‌ها و گروه‌های آموزشی است و شامل پُست‌های سازمانی ریاست دانشگاه، ریاست دانشکده، مدیر گروه آموزشی، عضو هیئت علمی و دانشجویان است. کلیه‌ی منابع انسانی در این محیط مطابق شکل (۲-۱) دسته‌بندی می‌شوند.



شکل (۲-۱) دسته‌بندی منابع انسانی بر اساس جایگاه و پُست سازمانی. منبع مشخص شده با دایره‌ی قرمز رنگ دارای جایگاهی سازمانی در گروه آمار در دانشکده‌ی علوم است و دارای پُست یا نقش سازمانی عضو هیئت علمی است.

در فرآیند سازمانی برای انتساب وظایف تعریف شده به منابع، الگوهای مختلفی وجود دارد. Russell و همکاران [۱] ۴۳ الگوی تخصیص واحد کاری به منبع معرفی کرده است و پشتیبانی این الگوهای را در پنج BPMS متفاوت^۱ مورد بررسی قرار داده‌اند. فهرست مهم‌ترین الگوهای تخصیص منبع عبارت‌اند از:

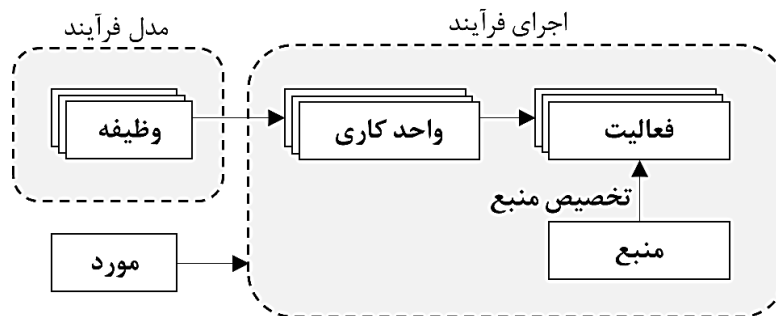
۱. تخصیص بر اساس نقش منابع
۲. تخصیص بر اساس چارت سازمانی
۳. تخصیص تصادفی
۴. تخصیص Round Robin
۵. تخصیص به کوچک‌ترین صف در لیست کاری
۶. تخصیص مستقیم به منبع

این الگوها را می‌توان به سه دسته‌ی کلی Pull، Push و Auto-start تقسیم‌بندی کرد. در الگوهای Pull یک واحد کاری در لیست مشترک به یک یا مجموعه‌ای از منابع پیشنهاد می‌شود سپس منبع به اختیار خود واحد کاری را انتخاب می‌کند. به عبارت دیگر تخصیص ابتدا با یک پیشنهاد همراه است. در این حالت منبع می‌تواند واحد کاری پیشنهاد شده را رد کند و واحد کاری توسط منبعی دیگر انتخاب شود. در الگوهای Push، واحد کاری به‌طور مستقیم در لیست کاری اختصاصی منبع قرار می‌گیرد و منبع مجبور به سرویس‌دهی به واحد کاری است. در این الگو فضایی برای لیست مشترک واحدهای کاری پیشنهادی وجود ندارد و فقط از لیست اختصاصی منابع استفاده می‌شود. وقتی منبع یک سخت-افزار یا یک سرویس نرم‌افزاری باشد می‌توان از الگوهای Auto-start استفاده کرد در این الگوها واحد کاری بدون قرار گرفتن در صف لیست اختصاصی یا لیست مشترک بعد از ایجاد، بلادرنگ سرویس‌دهی آن شروع می‌شود [۱].

^۱ Staffware, WebSphere, FLOWer, COSA, iPlanet

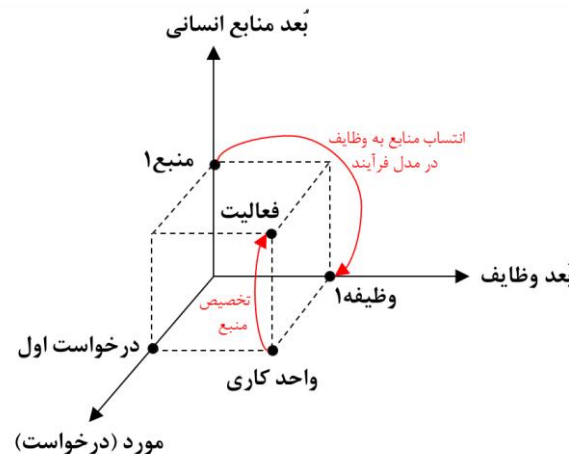
۳-۲-۱ وظیفه، واحد کاری و فعالیت

یک وظیفه، یک مفهوم انتزاعی به عنوان بخش غیرقابل تجزیه از مجموعه کارهایی است که باید انجام شود تا تولید یک محصول کامل گردد یا یک سرویس مشخص ارائه شود. این مفهوم مستقل از یک درخواست، پرونده یا یک «مورد» خاص است ولی وقتی یک درخواست وارد سیستم می شود به ازای وظایف تعریف شده در فرآیند سازمانی و به ازای هر درخواست (یا مورد) یک واحد کاری در اجرای فرآیند ایجاد می شود. وقتی واحد کاری ایجاد شده در اجرای فرآیند سازمانی به یک منبع تخصیص داده می شود مفهوم «فعالیت» به خود می گیرد. شکل (۳-۱) ارتباط مفاهیم ذکر شده را نشان می دهد.



شکل (۳-۱) ارتباط مفاهیم وظیفه، واحد کاری و فعالیت

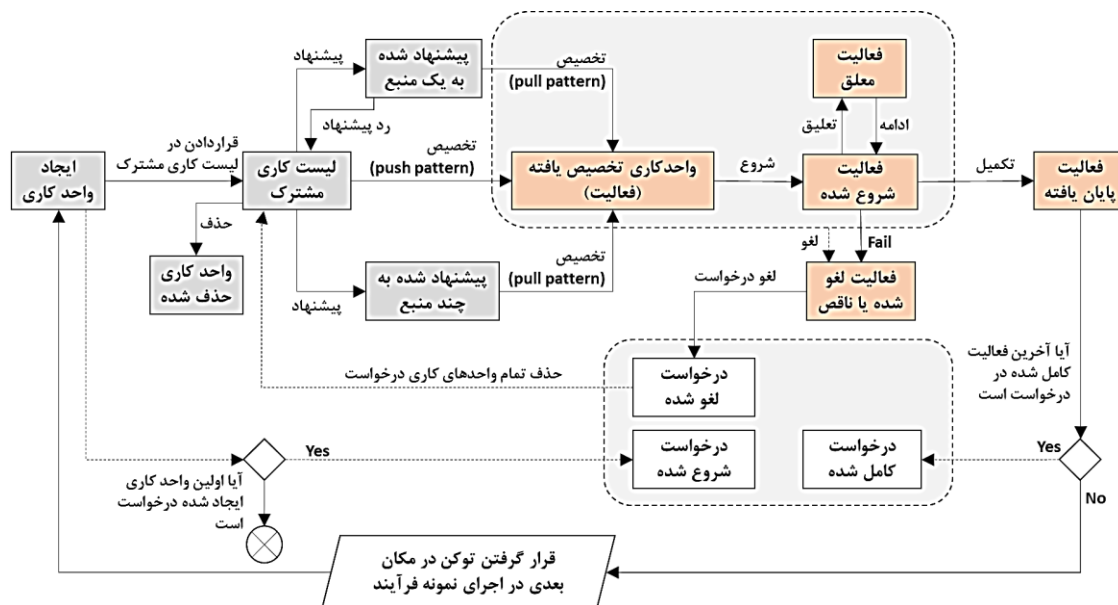
وقتی یک درخواست (مورد) وارد سیستم می شود یک نمونه از فرآیند ایجاد و اجرا می شود و به ازای هر وظیفه ی تعریف شده در مدل فرآیند یک واحد کاری در اجرای فرآیند ایجاد می شود. بنابراین واحد کاری باوجود درخواست (مورد) معنا می یابد. وقتی به واحد کاری در اجرای فرآیند به یک منبع تخصیص داده شود مفهوم «فعالیت» به خود می گیرد. بنابراین مفهوم «فعالیت» در کنار مفاهیم درخواست و منبع در مدل فرآیند کامل می شود. به ازای هر درخواست و وظیفه در مدل فرآیند «واحد کاری» تعریف می شود و در اجرای فرآیند از تخصیص یک واحد کاری به منبع، مفهوم «فعالیت» تعریف می شود. شکل (۴-۱) ارتباط مفاهیم فعالیت، وظیفه، واحد کاری، منبع و مورد را در فرآیند نشان می دهد.



شکل (۴-۱) وابستگی مفاهیم در تکمیل مفهوم فعالیت

۴-۲-۱ سابقه‌ی رویدادها

در اجرای فرآیند سازمانی وقتی یک واحد کاری ایجاد می‌شود؛ مجموعه‌ای از رویدادها را در چرخه‌ی حیات خود تجربه می‌کند که باعث تغییر حالت واحد کاری می‌شود. در موتور اجرای فرآیند به ازای هر وظیفه‌ی تعریف شده در مدل فرآیند یک لیست کاری مشترک وجود دارد. بعد از ایجاد واحد کاری، اولین رویداد «قرار گرفتن در لیست کاری مشترک» روی واحد کاری رخ می‌دهد و واحد کاری در لیست مشترک (متناظر با وظیفه‌ی تعریف شده در مدل فرآیند) قرار می‌گیرد. سپس با توجه به الگوهای تخصیص منبع یکی از رویدادهای «تخصیص» به یک منبع مشخص یا «پیشنهاد» به یک یا مجموعه‌ای از منابع رخ می‌دهد و حالت واحد کاری را تغییر می‌دهد. بعد از تخصیص واحد کاری به منبع مربوطه، واحد کاری از لیست مشترک واحدهای کاری خارج می‌شود و در لیست کاری اختصاصی منبع قرار می‌گیرد. واحد کاری تخصیص یافته به منبع مفهوم «فعالیت» را به خود می‌گیرد و رویدادهای شروع، تکمیل، تعلیق، لغو، شکست و ادامه را تجربه خواهد کرد. وقتی رویداد «تکمیل» روی واحد کاری رخ دهد؛ اگر این واحد کاری آخرین واحد کاری تعریف شده در نمونه فرآیند (متناظر با هر درخواست یک نمونه فرآیند ایجاد می‌شود) نباشد، یک واحد کاری جدید برای وظیفه‌ی بعدی تعریف می‌شود و چرخه‌ی حیات برای واحد کاری جدید شروع می‌شود تا زمانی که آخرین واحد کاری متناظر با درخواست تکمیل شود.

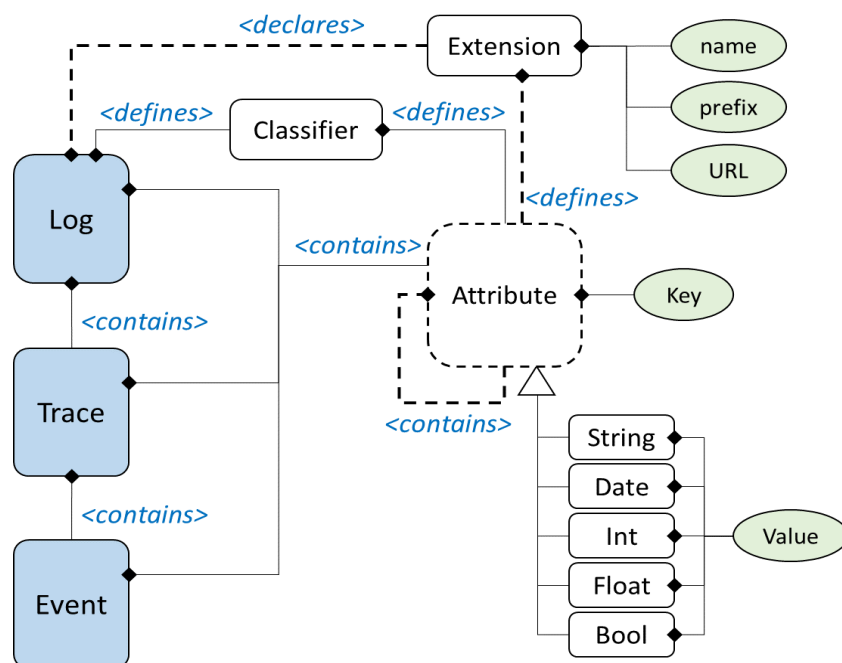


شکل (۵-۱) چرخه‌ی حیات واحد کاری در کنار وضعیت‌های درخواست

در شکل (۵-۱) چرخه‌ی حیات واحد کاری نشان داده شده است. بخشی از این شکل نمودار حالات واحد کاری است که در کنار سه حالت درخواست نشان می‌دهد. همان‌طور که در شکل مشخص شده است وقتی اولین واحد کاری از یک درخواست در اجرای نمونه فرآیند ایجاد می‌شود، درخواست در حالت «شروع شده یا فعال» قرار می‌گیرد و بعد از تکمیل آخرین واحد کاری (فعالیت)، وضعیت درخواست در حالت «کامل شده» قرار می‌گیرد. زمانی که یک درخواست فعال است مجموعه‌ای مشخص از واحدهای کاری، یکی پس از دیگری ایجاد و تکمیل می‌شوند ولی ممکن است در این بین، کل درخواست لغو شود یا یکی از واحدهای کاری آن با شکست مواجه شود. در این حالت، وضعیت درخواست به «لغو شده» تغییر می‌یابد و این رویداد باعث می‌شود سایر واحدهای کاری مرتبط به آن درخواست از لیست‌های اختصاصی یا لیست مشترک واحدهای کاری حذف شوند.

یکی از اهداف PBMS نظارت بر روند اجرای فرآیندهای سازمانی است. برای این منظور کلیه‌ی رویدادهایی که در اجرای فرآیند سازمانی رخ می‌دهد، می‌بایست به‌گونه‌ای مناسب ثبت شود تا امکان بررسی مجدد آن جهت بهبود عملکرد سیستم به‌سادگی امکان‌پذیر باشد. Verbeek و همکاران [۲] قالبی

تحت عنوان جریان رویداد توسعه‌پذیر^۱ (XES) به پایه‌ی XML معرفی کرده‌اند. به کمک این قالب امکان ثبت سابقه‌ی رویدادهای هر فرآیند سازمانی میسر می‌شود. شکل (۶-۱) یک فرا مدل از نمودار کلاس‌های قالب XES را نشان می‌دهد. در این مدل سه کلاس اصلی وجود دارد که شامل Log، Trace و Event است. هر Log شامل مجموعه‌ای از Trace و هر Trace شامل مجموعه‌ای Event است. هر یک از اشیاء تعریف شده در این قالب می‌توانند شامل مجموعه‌ای از خصوصیت (Attribute) باشند. در این قالب امکان توسعه و اتصال شیء Log به قالب‌های مشابه و Log‌های دیگر از طریق اشیاء Extension وجود دارد.



شکل (۶-۱) فرا مدل قالب XES [۲].

وقتی یک درخواست به یک فرآیند سازمانی می‌رسد در سابقه‌ی رویداد یک شیء Trace ایجاد می‌شود و به ازای هر رویداد که روی واحدهای کاری متناظر با آن درخواست رخ می‌دهد یک شیء Event در سابقه‌ی رویداد ثبت می‌شود. شکل (۷-۱) بخشی از فایل پایگاه داده BPI Challenge 2012 که در قالب XES ذخیره شده است را نشان می‌دهد.

^۱ Extensible Event Stream

```

<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xmlns="http://www.xes-standard.org/">
  <extension name="Metadata_Time" prefix="meta_time"
uri="http://www.xes-standard.org/meta_time.xesext"/>
  <classifier name="Resource classifier"
keys="org:resource"/>
  <trace>
    <date key="REG_DATE" value="2011-10-01, 38:44"/>
    <string key="concept:name" value="173688"/>
    <string key="AMOUNT_REQ" value="20000"/>
    <event>
      <string key="org:resource" value="112"/>
      <string key="lifecycle:transition" value="COMPLETE"/>
      <string key="concept:name" value="SUBMITTED"/>
      <date key="time:timestamp" value="2011-10-01, 38:44"/>
    </event>
    :
  </trace>
  <trace>
    <date key="REG_DATE" value="2011-10-01, 08:58"/>
    <string key="concept:name" value="173691"/>
    <string key="AMOUNT_REQ" value="5000"/>
    <event>
      <string key="org:resource" value="112"/>
      <string key="lifecycle:transition" value="COMPLETE"/>
      <string key="concept:name" value="A_PARTLYSUBMITTED"/>
      <date key="time:timestamp" value="2011-10-01, 09:02"/>
    </event>
    :
  </trace>
  :
</log>
</xml>

```

شکل (۷-۱) بخشی از سابقه‌ی رویداد فرآیند موسسه‌ی مالی هلند که در پایگاه داده‌ی BPI Challenge 2012 موجود است در قالب فایل XES

۱-۲-۵ فرآیند سازمانی

یک فرآیند سازمانی به صورت چندتایی $p = \langle \lambda(t), W, R, L, U, \mu, \xi \rangle$ تعریف می‌شود.

ا. $\lambda(t)$ تابع نرخ ورود درخواست مشتریان است. (در مدل تحلیلی فرآیند توزیع آماری درخواست‌های

ورودی را پواسن فرض می‌کنیم)

ب. مجموعه‌ی W وظایف تعریف شده در فرآیند است.

ت. مجموعه‌ی R منابع انسانی موجود در سازمان است.

ث. مجموعه‌ی $L \subseteq W \times W$ شامل اتصالات بین وظایف را مشخص می‌کند. به‌عنوان مثال، $\langle w_1, w_2 \rangle \in L$

نشان می‌دهد در فرآیند p وظیفه w_2 بعد از وظیفه w_1 انجام می‌شود.

ج. مجموعه‌ی $U \subseteq W \times R$ منابع مسئول هر وظیفه را مشخص می‌کند. به‌عنوان مثال $\langle w, r \rangle \in U$

نشان می‌دهد در فرآیند p منبع r مسئول انجام دادن وظیفه w است. طبق تعریف U می‌توان نتیجه

گرفت در فرآیند p برای هر وظیفه $w \in W$ ممکن است چند منبع مسئول وجود داشته باشد،

بنابراین به‌اختصار $R_w = \{r | \langle w, r \rangle \in U\}$ تعریف می‌کنیم که لیست کلیه‌ی منابع

مسئول فعالیت w در فرآیند p است.

ح. $\mu: U \rightarrow \mathbb{R}^+$ یک نگاشت از مجموعه‌ی U به اعداد حقیقی مثبت ایجاد می‌کند. مقدار $\mu(\langle w, r \rangle)$

بیانگر متوسط زمان انجام وظیفه w توسط منبع r است که به‌اختصار با $\mu_{w,r}$ نشان می‌دهیم.

خ. $\xi: U \rightarrow [0,1]$ یک نگاشت از مجموعه‌ی U به اعداد حقیقی بین صفر و یک است. از آنجایی که برای

هر وظیفه w مجموعه‌ای از منابع مسئول R_w وجود دارد و در اجرا فرآیند فقط یکی از منابع وظیفه‌ی

w را انجام می‌دهد، $\xi(\langle w, r \rangle)$ بیانگر احتمال انجام وظیفه w توسط منبع r است که به‌اختصار

به‌صورت $\xi_{w,r}$ نشان می‌دهیم.

۱-۲-۶ بارکاری منابع

وقتی یک فرآیند سازمانی در شبکه‌ی صف مدل می‌شود، هر وظیفه در فرآیند یک ایستگاه

سرویس‌دهی است و هر منبع منتسب به این وظیفه، یک عامل سرویس‌دهنده است. از آنجایی که در هر

وظیفه w احتمال قرار گرفتن واحد کاری در لیست اختصاصی منبع $\xi_{w,r}$ است، رابطه‌ی (۱-۱) برای

تمام سرویس‌دهنده‌های (منابع) تعریف‌شده در یک ایستگاه سرویس‌دهی (وظیفه) برقرار است.

$$\sum_{r \in R_w} \xi_{w,r} = 1, \forall w \in W \quad (1-1)$$

میزان بارکاری هر ایستگاه سرویس‌دهی را می‌توان با داشتن نرخ ورود درخواست به هر ایستگاه

و نرخ سرویس‌دهی به دست آورد. اگر مشتریان در هر صف سرویس‌دهی به چند دسته تقسیم شوند و

نرخ ورود دسته i ام را با λ_i نشان دهیم و میانگین نرخ سرویس دهی را با μ_i نشان دهیم. متوسط نرخ بهره‌وری ρ از سرویس‌دهنده‌ها از رابطه‌ی (۲-۱) به دست می‌آید.

$$\rho = \sum \frac{\lambda_i}{\mu_i} \quad (2-1)$$

هر دسته در رابطه‌ی (۲-۱) بیانگر یک وظیفه در فرآیند و منابع مسئول آن است. اگر $\varphi_w(t)$ تابع نرخ ورود وظیفه w در زمان t باشد و $\xi_{w,r}$ احتمال تخصیص وظیفه w به منبع r باشد و $\mu_{w,r}$ میانگین نرخ سرویس دهی منبع r روی وظیفه w باشد. رابطه‌ی (۳-۱) میزان بارکاری منبع r را در زمان t محاسبه می‌کند. در رابطه‌ی (۳-۱)، $W_r = \{w | \langle w, r \rangle \in U\}$ مجموعه تمام وظایفی است که در فرآیند برای منبع r تعریف شده است.

$$ld_r(t) = \sum_{w \in W_r} \frac{\xi_{w,r} \cdot \varphi_w(t)}{\mu_{w,r}} \quad (3-1)$$

۷-۲-۱ تعادل بارکاری

در سابقه‌ی تحقیق تعاریف متفاوتی از تعادل بارکاری منابع ارائه شده است [۳-۶]. در اینجا دو تعریف را ارائه می‌دهیم که از نظر مفهوم، یکسان هستند. در تعریف اول طبق رابطه‌ی (۴-۱) حداکثر بارکاری منابع باید حداقل شود.

$$\min \max_{r \in R} \left\{ \int ld_r(t) d(t) \right\} \quad (4-1)$$

در تعریف اول مجموع بارکاری را در طول زمان در نظر گرفته است در حالی که در تعریف دقیق‌تر، بارکاری منابع در هر لحظه از زمان باید متعادل باشد. به عبارت دیگر، برای یک بازه‌ی زمانی مشخص $[t_0, t_N]$ مجموع تغییرات بارکاری باید کمینه شود. به بیان دیگر، اگر در بازه‌ی زمانی $[t_0, t_N]$ مقدار رابطه‌ی (۵-۱) کمینه شود تعادل بارکاری برای منابع به وجود آمده است.

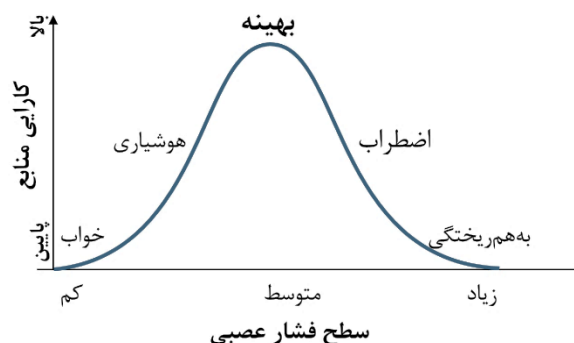
$$EL(t) = \sum_{r \in R} ld_r(t) / |R| \quad (5-1)$$

$$\min \left\{ \frac{\sum_{r \in R} \sum_{i=0}^N (ld_r(t_i) - EL(t_i))^2}{N|R|} \right\} \quad (6-1)$$

از آنجایی که تغییرات در بارکاری منابع در لحظه‌ی تخصیص وظایف به منابع انجام می‌شود، مقادیر t در رابطه‌ی (۶-۱) گسسته فرض شده است. تعریف ارائه شده از تعادل بارکاری در رابطه‌ی (۶-۱) به تغییرات بارکاری در هر لحظه از زمان توجه دارد به‌طوری‌که ابتدا میانگین بارکاری در لحظه‌ی t محاسبه می‌شود ($EL(t)$)، سپس تغییرات بارکاری نسبت به میانگین در هر لحظه از زمان، باید کمینه شود تا تعادل بارکاری برای منابع در طول زمان ایجاد شود.

۸-۲-۱ یکنواختی بارکاری یک منبع

در تحقیقی که توسط Alluisi و Morgan انجام شد [۷]، میزان تأثیر فشار عصبی منابع بر بازدهی آن‌ها اندازه‌گیری شد. همان‌طور که در شکل (۸-۱) آمده است، این تحقیق نشان داد. منابع انسانی در حد متعادلی از فشار عصبی دارای بازدهی بهینه هستند و بازدهی منابع نسبت به میزان فشار عصبی آن‌ها دارای تابع گوسی شکل است. از آنجایی که بارکاری منابع با فشار عصبی آن‌ها رابطه‌ی مستقیم دارد بنابراین از نتیجه‌ی این تحقیق می‌توان برای مدل کردن میزان کارایی منابع بر اساس بارکاری آن‌ها استفاده کرد، به عبارت دیگر برای هر منبع سطح مشخصی از بارکاری وجود دارد که اگر بارکاری در این سطح باشد، منبع مورد نظر دارای حداکثر کارایی و بازدهی در انجام وظایف خود است.



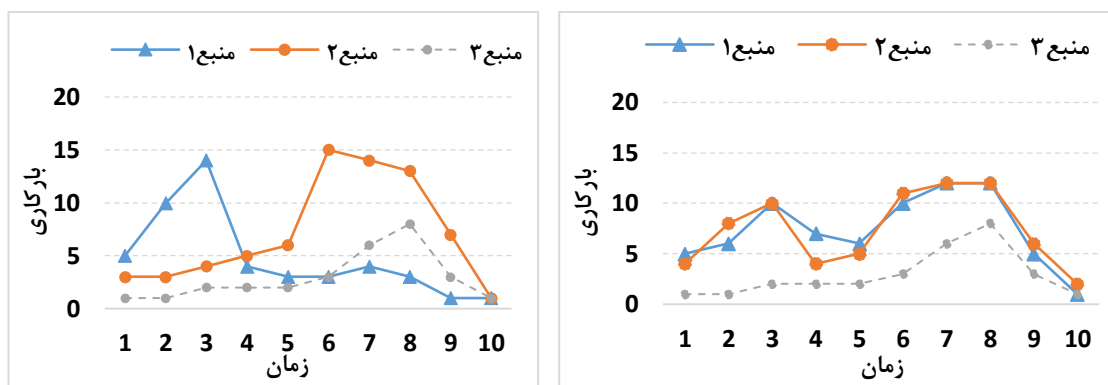
شکل (۸-۱) قاعده‌ی Yerkes-Dodson نشان می‌دهد هرچه سطح فشار عصبی منابع انسانی پایین باشد کارایی آن پایین است و حتی اگر سطح فشار عصبی بیش از حد باشد باز هم کارایی پایین است. فقط در سطح معینی از فشار عصبی کارایی منابع در حالت بهینه خود قرار می‌گیرد [۷]

از آنجایی که میزان بارکاری هر منبع رابطه‌ی مشخصی با میزان فشار عصبی آن دارد، بهتر است بارکاری هر منبع r در سطحی مشخص μ_r ، با حداقل تغییرات باشد. بنابراین ایجاد یکنواختی در بارکاری هر منبع در طول زمان در سطح مشخص μ_r می‌تواند باعث توزیع بهتری از حجم کاری منبع در طول دوره‌ی کاری‌اش ایجاد کند و در نتیجه باعث افزایش کارایی آن منبع و افزایش کیفیت سرویس‌دهی شود. رابطه‌ی (۷-۱) تعریف کم‌ترین تغییرات نسبت به سطح بهینه برای بارکاری منابع را نشان می‌دهد. هر چه این مقدار کم‌تر باشد به معنای افزایش یکنواختی بارکاری منابع است.

$$\min \left\{ \frac{\sum_{r \in R} \sum_{i=0}^N (ld_r(t_i) - \mu_r)^2}{N|R|} \right\} \quad (7-1)$$

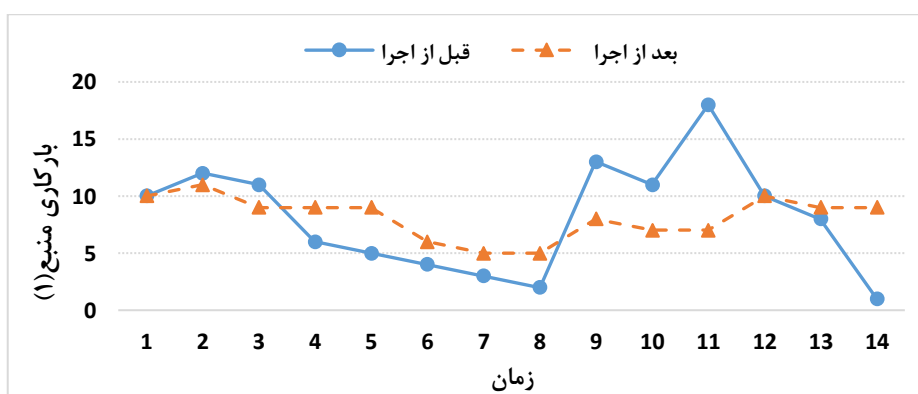
۳-۱ طرح مسئله

نظارت بر فعالیت‌های منابع انسانی و بررسی بارکاری آن‌ها در بسیاری از سازمان‌ها، بسیار مشکل و گاهی غیرممکن است. پیشبرد اهداف سازمان و رعایت عدالت کاری منابع انسانی در صورتی حاصل می‌شود که بر فعالیت‌های آن‌ها کنترل جامع و کاملی صورت گیرد و نظارت بر آن‌ها با توجه به قابلیت‌ها و توانایی‌های آن‌ها باشد. به عنوان مثال در بازه‌های زمانی مشخص با توجه به نوع فرآیندها، ممکن است بارکاری دو منبع با توانایی یکسان متفاوت باشد یا حتی در مورد یک منبع ممکن است بارکاری در ساعات مختلف کاریش یکسان نباشد. درحالی‌که اگر بارکاری توزیع یکنواختی در روز می‌داشت، می‌توانست در اجرای باکیفیت‌تر فعالیت‌ها تأثیرگذار باشد. بنابراین وجود الگوریتمی، برای نظارت و کنترل فرآیندها و بارکاری منابع ضروری است. این الگوریتم می‌بایست بتواند، با توجه به بارکاری منابع؛ مسیر توکن‌ها را کنترل کند و اختصاص مناسب و زمان-واقعی وظایف به منابع را انجام دهد به‌طوری‌که تعادل در بارکاری منابع با نقش‌های یکسان و یکنواختی بارکاری را در بازه زمانی فعالیت کاربر تضمین کند. شکل (۹-۱) وضعیت بارکاری سه منبع، قبل و بعد از اجرای روش مورد انتظار را نشان می‌دهد.



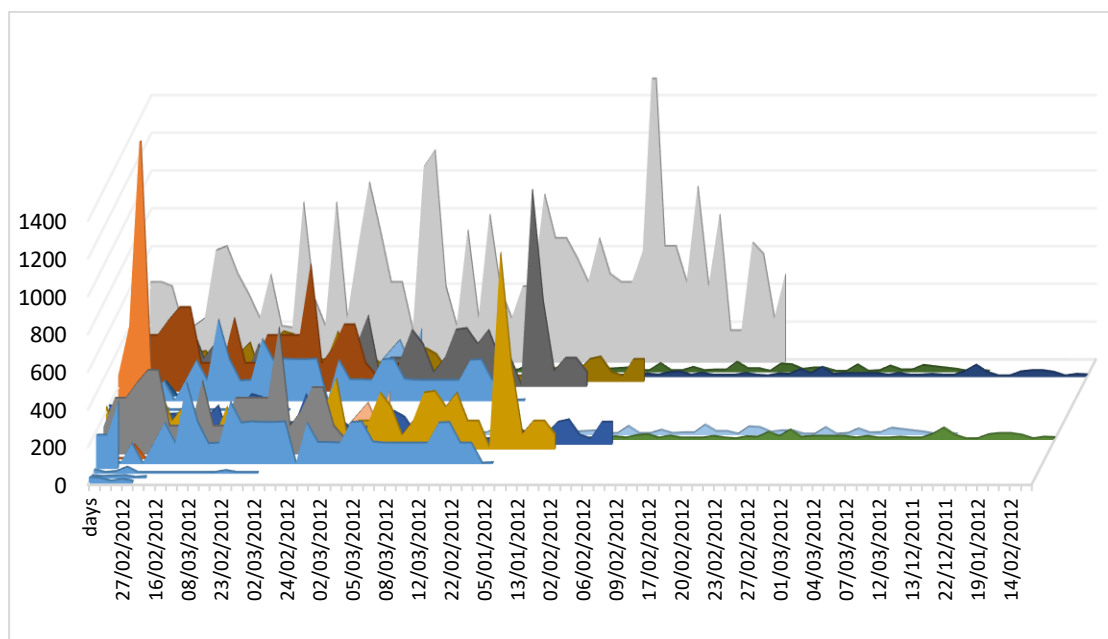
الف) بعد از اجرا
ب) قبل از اجرا
شکل (۹-۱) وضعیت بارکاری قبل و بعد از اجرای روش مورد انتظار در بازه‌ی زمانی مشخص

از طرفی دیگر ایجاد یکنواختی در بارکاری هر منبع در سطحی مشخص می‌تواند باعث بهبود عملکرد منبع و در نهایت باعث بهبود کیفیت محصول یا سرویس‌دهی شود. شکل (۱۰-۱) وضعیت بارکاری منبع (۱) را در دو حالت بعد و قبل از اجرای روش مورد انتظار را نشان می‌دهد. نمودار بارکاری در وضعیت بعد از اجرا دارای یکنواختی بیشتری است و این یکنواختی باعث می‌شود سطح فشار عصبی منبع در سطح متعادلی جهت بهبود کارایی منبع قرار بگیرد و در طولانی مدت باعث بهبود در سرویس‌دهی منبع خواهد شد.



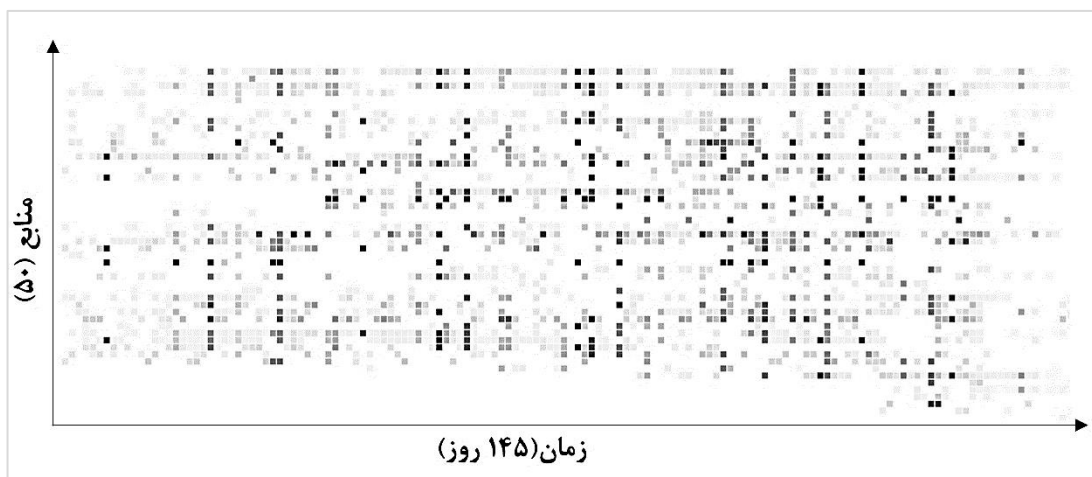
شکل (۱۰-۱) یکنواختی بارکاری یک منبع قبل و بعد از اجرای روش مورد انتظار در بازه‌ی زمانی مشخص

برای روشن شدن مسئله می‌توان به یک مثال از پایگاه داده‌ی واقعی BPI Challeng 2012 توجه کرد. شکل (۱۱-۱) بارکاری ۳۰ منبع از پایگاه داده ذکر شده را نشان می‌دهد. با اینکه در این پایگاه داده منابع دارای نقش و توانایی یکسانی بودند اما همان‌طور که در شکل مشخص است بارکاری آن‌ها در طول ۱۴۵ روز کاملاً باهم متفاوت است به بیان دیگر تعادل بارکاری در منابع رعایت نشده است.

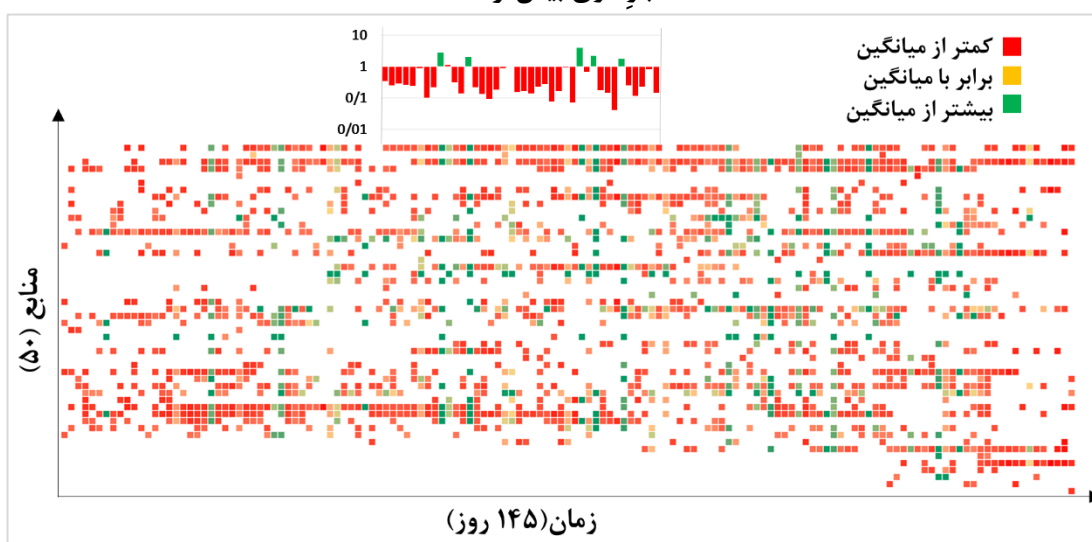


شکل (۱۱-۱) بارکاری تعدادی از منابع در پایگاه داده‌ی BPI Challenge 2012

شکل‌های (۱۲-۱) و (۱۳-۱) نمایش نقطه‌ای بارکاری منابع را در پایگاه داده‌ی BPI Challenge 2012 نشان می‌دهند. همان‌طور که در شکل (۱۲-۱) مشخص است حجم بیشتر بارکاری فقط در چند نقطه‌ی محدود (نسبت به کل تصویر) جمع شده است. اگر بخواهیم مفاهیم تعادل و یکنواختی بارکاری را بر اساس نمایش نقطه‌ای بارکاری بیان کنیم آنگاه تعادل بارکاری منابع به معنای تغییرات کم‌تر رنگ نقاط در راستای عمودی (محور منابع) در نمایش نقطه‌ای بارکاری در شکل (۱۲-۱) است و یکنواختی بارکاری منابع به معنای تغییرات کم‌تر رنگ نقاط در راستای افقی (محور زمان) در نمایش نقطه‌ای بارکاری است. به بیان دیگر هر چه نقاط زرد رنگ در شکل (۱۳-۱) بیش‌تر باشد، یکنواختی و تعادل بارکاری منابع بیش‌تر است. به‌عنوان مثال در بخش میانی بالاترین سطر از نقاط در شکل (۱۳-۱) تغییرات بارکاری منبع در حدود ۰/۱ میانگین کل بارکاری منابع است که می‌تواند به یکنواختی بارکاری منبع سطر مربوطه در بازه‌ی زمانی محدود اشاره کند.



شکل (۱۲-۱) نمایش نقطه‌ای بارکاری منابع در پایگاه داده‌ی BPI Challenge 2012. نقاط تیره‌تر بیانگر بارکاری بیش‌تر است



شکل (۱۳-۱) نمایش نقطه‌ای انحراف بارکاری (غیرصفر) از میانگین بارکاری منابع در پایگاه داده‌ی BPI Challenge 2012

۴-۱ هدف مسئله

هدف اصلی این رساله ارائه‌ی یک مدل آماری بر پایه‌ی اطلاعات ثبت شده در سابقه‌ی رویداد (فرآیند-آگاه) برای حل مسئله‌ی تخصیص منابع با هدف (۱) ایجاد بارکاری متعادل برای منابع و (۲) ایجاد یکنواختی بارکاری هر منبع، در محیط‌های چند فرآیندی در BPMS است. در این مدل از تکنیک‌های آماری جهت ایجاد مدل فرآیند سازمانی استفاده می‌کنیم و به دلیل تعاملی بودن محیط طرح مسئله، از الگوریتم یادگیری تقویتی جهت حل مسئله‌ی تعریف شده، بهره می‌بریم.

۵-۱ فرضیه‌ها

- توسعه‌ی مدل فرآیندهای تصمیم‌گیری مارکوف (MDPs) با هدف افزایش تطبیق مدل‌سازی اجرای فرآیندهای سازمانی.
- استخراج دانش نهفته در سابقه‌ی رویدادها و تحلیل و آنالیز آماری آن در جهت ارائه‌ی یک الگوریتم تخصیص منبع در قالب مدل پیشنهادی به‌طوری‌که اهداف رساله تأمین شود.
- طرح مسئله‌ی رساله در محیط‌های چند فرآیندی و استفاده از خصوصیات و قابلیت‌های موجود در محیط‌های چند فرآیندی در جهت رسیدن به اهداف رساله.

۶-۱ چالش‌ها

- تغییر آهنگ ورود درخواست‌ها در طول زمان باعث می‌شود اثر نامطلوب روی یکنواختی بارکاری منابع داشته باشد. این اثر نامطلوب با توجه به محل تخصیص وظایف به منابع متفاوت است. روی منابعی که در ابتدای فرآیند هستند تغییر آهنگ ورود درخواست تأثیر نامطلوب بیش‌تری ایجاد می‌کند. در مدل‌سازی فرآیندها در شبکه‌ی صف به دلیل پیچیده شدن معادلات زمان انتظار و طول صف فرض می‌شود نرخ ورود درخواست‌ها دارای توزیع مشخص با پارامترهای ثابت است.
- توزیع نامتقارن منابع بر اساس نقش آن‌ها در مدل فرآیند نیز باعث نامتعادل شدن بارکاری منابع و غیریکنواختی بارکاری هر منبع می‌شود. منابع با توجه به توانایی و تخصصی که دارند در قسمت‌های مختلف فرآیند سازمانی فعالیت می‌کنند و وظایف متفاوتی به آن‌ها منتسب می‌شود لذا بدیهی است این تفاوت‌ها باعث تغییر در زمان و مقدار بارکاری منابع می‌شود.

- تخصیص فعالیت به منابع لزوماً به صورت کسری امکان پذیر نیست. نرخ ورود ممکن است به اندازه کافی نباشد به نحوی که سهم هر منبع حداقل انجام یک فعالیت شود. فرض کنید به وظیفه‌ی ابتدای فرآیند، سه منبع منتسب شده باشد. اگر ورودی این فرآیند، یک درخواست در ساعت باشد. در اولین ساعت کار فرآیند فقط یکی از سه منبع مذکور مشغول خواهد بود و اگر بخواهیم تعادل بارکاری منابع را در هر ساعت بررسی کنیم، آهنگ ورود بیش از حد کم یک فرآیند اثر نامطلوب در تعادل بارکاری منابع خواهد داشت و این به دلیل عدم تخصیص کسری وظایف در مدل فرآیند است. برای رفع این مشکل عموماً از الگوهای تخصیص pull استفاده می‌شود.
- اساساً مقدار بارکاری به ازای هر فعالیت یک مقدار تخمینی است. الگوریتم‌های تخصیص منابع باید از مقدار دقیق بارکاری هر وظیفه تخصیص داده شده مطلع باشند، درحالی که بارکاری هر وظیفه با توجه به اطلاعات درخواست از یک مورد تا مورد دیگر دائماً در حال تغییر است و تعیین مقدار دقیق بارکاری هر وظیفه میسر نیست لذا الگوریتم‌های موجود با بررسی سابقه‌ی رویدادهای سیستم از مقدار تخمینی بارکاری استفاده می‌کنند بنابراین مقدار خطای موجود در مقدار تخمینی بارکاری هر وظیفه، در برقراری تعادل بارکاری منابع اثر نامطلوب خواهد داشت.
- ساختار نامتقارن فرآیند سازمانی یکی دیگر از عللی است که باعث نامتعادل شدن بارکاری منابع در اجرای فرآیندهای سازمانی می‌شود. تنوع ساختارهای کنترلی در طراحی مدل فرآیند سازمانی مانند امکان تعریف حلقه، انشعاب OR، AND و XOR باعث می‌شود بارکاری که به ازای هر درخواست ورودی به فرآیند تحمیل می‌شود مسیرهای متفاوتی را در جریان اجرای فرآیند طی کند در نتیجه با توجه به محل انتساب منابع در مدل فرآیند، بارکاری تحمیلی هر درخواست بر بارکاری منابع متفاوتی اثر خواهد داشت.

فصل ۲ – پیشینه‌ی پژوهش

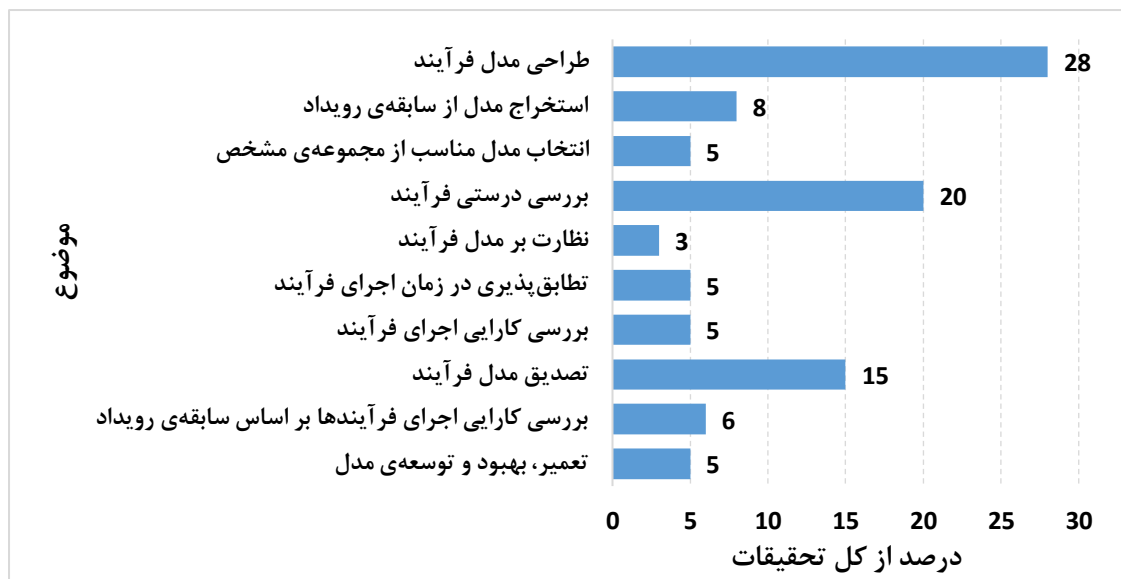
از دیدگاه مدل:

- مدل فرآیندهای تصمیم‌گیری مارکوف
- مدل شبکه‌ی صف‌بندی
- مدل WF-net

از دیدگاه هدف:

- تعادل بارکاری
- کاهش چرخه‌ی زمانی

همان‌طور که در ابتدای فصل قبل اشاره شده BPMS حداقل شامل سه بخش مدل‌سازی فرآیند، موتور اجرای فرآیند و بهینه‌ساز است. در ادبیات موضوع در هر یک از این بخش‌ها تحقیقات زیادی انجام شده است. شکل (۱-۲) نشان می‌دهد، برخی از این تحقیقات در پی طراحی و ارائه‌ی مدلی جامع و کاربردی در عین حال قابل محاسبه برای تعریف فرآیند هستند و برخی از تحقیقات در جهت طراحی و پیاده‌سازی موتوری قدرتمند و سازگار با مدل هستند. برخی دیگر به دنبال ارائه‌ی روش‌هایی برای بهبود عملکرد کلی سیستم در تخصیص وظایف و کاهش چرخه‌ی زمانی درخواست‌ها و ایجاد تعادل بارکاری منابع هستند. در ادامه‌ی این بخش تحقیقات انجام شده را با توجه به هدف رساله از دیدگاه روش‌های مدل‌سازی فرآیند سازمانی و از دیدگاه هدف بهینه‌سازی یا موتور اجرای فرآیند دسته‌بندی می‌کنیم.



شکل (۱-۲) درصد موضوع تحقیقات انجام شده در BPMS [۸]

۱-۲ از دیدگاه مدل

برای تحلیل و آنالیز فرآیندها در یک BPMS سه مدل تحلیلی در سابقه‌ی تحقیق دیده می‌شود [۳]. (۱) مدل تحلیلی مبتنی بر شبکه‌ی صف (QNM)^۱ (۲) مدل تحلیلی مبتنی بر پردازش‌های تصمیم‌گیری مارکوف^۲ (MDPs) که به‌طور گسترده برای حل دنباله‌ای از مسائل تصمیم‌گیری استفاده

^۱ Queuing Network Model

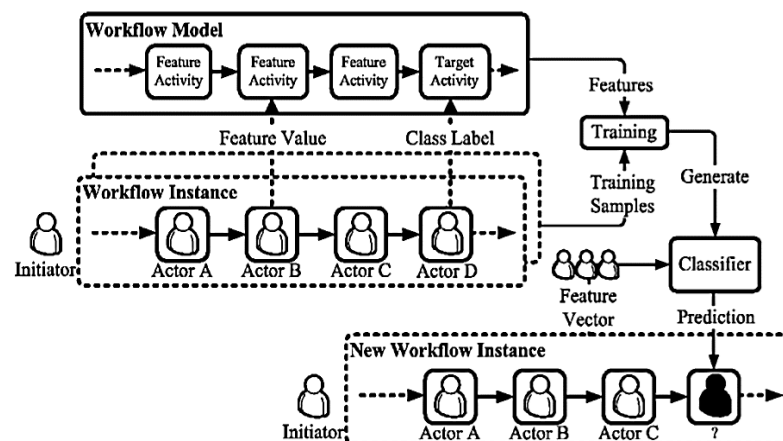
^۲ Markov Decision Processes

شده است و (۳) مدل تحلیلی WF-Net به عنوان یک مدل توسعه یافته‌ی از Petri-Net. در این مجموعه، MDPs مدلی است که برای حل آن از یادگیری تقویتی استفاده شده است، مهم‌ترین خصوصیت مدل MDPs تصمیم‌گیری بهینه برای هر حالت، مستقل از حالت قبل است. برای MDPs تعدادی الگوریتم وجود دارد که تصمیم‌گیری بهینه را برای حالت پایانی تضمین می‌کند. به عنوان مثال روش برنامه‌نویسی پویا می‌تواند برای پیدا کردن حالت بهینه مناسب باشد. دسته‌ی نسبتاً جدیدی از الگوریتم‌های شناخته شده تحت عنوان تکنیک یادگیری تقویتی^۱ (RL) ممکن است در حل مسئله‌ی تخصیص وظیفه به منابع نسبت به برنامه‌نویسی پویا بهتر عمل کند. تکنیک یادگیری تقویتی در زمینه‌های مختلفی برای حل مسائل استفاده شده است و نتایج رضایت بخشی از آن حاصل شده است. از نظر عملیاتی مسئله‌ی تصمیم‌گیری در تخصیص وظیفه به منابع یک مسئله‌ی تعاملی است. برای انتساب یک کار خاص به یک منبع مناسب در زمان مشخص می‌بایست تعاملی با محیط انجام کار وجود داشته باشد، با مشاهده‌ی درست محیط اجرای فرآیند می‌توان تصمیم درست گرفته شود.

۲-۱-۱ مدل فرآیندهای تصمیم‌گیری مارکوف (MDPs)

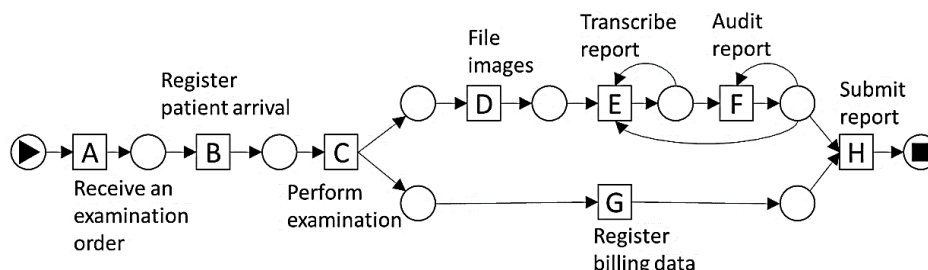
در تحقیق Yingbo Liu و همکاران یک روش نیمه خودکار با الگوریتم یادگیری با نظارت برای تخصیص منابع انجام شد. در این الگوریتم در فرآیند یادگیری، خصوصیات فعالیت‌ها و کاربران مورد مطالعه و بررسی قرار گرفت و نوعی ارتباط بین فعالیت و کاربر انجام دهنده‌ی آن ایجاد شد که منجر به الگوریتم هوشمندی جهت تخصیص منبع گردید. محققان برای پیاده‌سازی از کتابخانه ProM استفاده کردند [۹]. همان‌طور که در شکل (۲-۲) مشخص است؛ نمونه‌های مختلفی که از یک فرآیند روی پرونده‌ها اجرا می‌شوند و در هر مرحله در تخصیص منبع از اطلاعات منابع تخصیص یافته شده روی آن پرونده در مراحل قبلی استفاده می‌کند تا جایی که یک یادگیری با نظارت پارامترهای تخصیص منبع را تنظیم می‌کند و موتور اجرای فرآیند را آموزش می‌دهد.

^۱ Reinforcement Learning



شکل (۲-۲) فرآیند نیمه خودکار یادگیری تقویتی با نظارت برای تخصیص منابع [۹]

Huang و همکاران [۱۰] یک مدل تحلیلی مبتنی بر مدل تصمیم‌گیری مارکوف جهت تخصیص منابع ارائه دادند. در این مدل یک تابع هزینه بر اساس وضعیت‌های سیستم در طول زمان تعریف می‌شود و هدف یافتن یک سیاست کلی جهت تخصیص منابع به صورتی که هزینه‌ی کلی تغییر وضعیت‌های سیستم کمینه شود. Huang و همکاران مدلی با عنوان $RLRAM^1$ جهت تخصیص منابع، بر پایه یادگیری تقویتی ارائه کرده‌اند که با تخصیص منابع در فرآیندهای سازمانی به عنوان یک مسئله‌ی تعاملی برخورد می‌کنند. به‌طوری‌که الگوریتم تخصیص منابع می‌تواند با محیط اجرای فرآیندی تعامل داشته باشد تا وظایف را به منابع مناسب در طول مسیر حرکت توکن‌ها اختصاص دهد. در این تحقیق از MDPs برای مدل‌سازی محیط مسئله استفاده شد. روش پیشنهادی ایشان با یک الگوریتم حریصانه روی پایگاه داده‌ی Radiology CT-scan (شکل (۲-۳)) آزمایش شد. محققان در این تحقیق برای پیاده‌سازی روش خود از کتابخانه ProM استفاده کردند [۱۰].

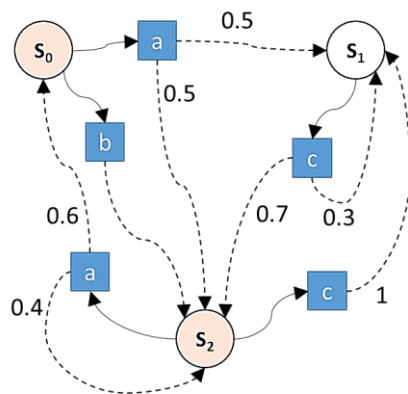


¹ Reinforcement Learning Resource Allocation Mechanism

شکل (۳-۲) فرآیند Radiology CT-scan [۱۰].

۲-۱-۱-۱ مسئله‌ی فرآیندهای تصمیم‌گیری مارکوف

در این مدل مجموعه‌ای از حالات یا وضعیت‌ها وجود دارد. در هر وضعیت مجموعه‌ای از عملیات متصور است با انجام هر عمل (Action) روی یک وضعیت، حالت یا وضعیت سیستم طبق یک احتمال مشخص تغییر می‌کند و با تغییر وضعیت سیستم به ازای یک عمل مشخص، مقدار مشخصی هزینه به سیستم تحمیل می‌شود. هدف انتخاب دنباله‌ای مشخص از عملیات است به‌طوری‌که مجموع هزینه‌ی تحمیلی به سیستم کمینه شود.



شکل (۴-۲) مثالی از مدل فرآیندهای تصمیم‌گیری مارکوف

اگر مدل فرآیند تصمیم‌گیری مارکوف را به صورت چهارتایی $M = \langle S, A, P, C \rangle$ در نظر بگیریم:

- S مجموعه‌ی حالت‌های سیستم یا محیط مسئله است.
- A مجموعه‌ی کل عملیاتی است که روی S قابل تعریف است و به ازای هر وضعیت $s \in S$ ، مجموعه‌ی $A_s \subseteq A$ عملیات مجاز در وضعیت s است.
- $P: S \times A \times S \rightarrow [0,1]$ تابع احتمال شرطی وضعیت بعدی سیستم مشروط به انتخاب یک عمل مشخص است، به بیان دیگر $P(s'|(s, a))$ احتمال رخ دادن حالت s' به عنوان حالت جدید سیستم است مشروط به اینکه در وضعیت s یک عمل $a \in A_s$ اتفاق افتاده باشد.

- $C: S \times A \rightarrow \mathbb{R}^+$ تابع هزینه به ازای عملیاتی است که در یک وضعیت اتفاق می‌افتد.

به عنوان مثال در شکل (۴-۲) یک مدل از فرآیندهای تصمیم‌گیری مارکوف نشان داده شده است.

در این مثال $S = \{s_0, s_1, s_2\}$ و $A = \{a, b, c\}$ است.

اگر فرض کنیم $\tau = 0, 1, 2, 3, \dots$ زمان‌های تصمیم‌گیری روی مدل فوق جهت انتخاب عملیات باشد و $s_\tau \in S$ وضعیت سیستم در زمان τ و $a_\tau \in A$ عملیات انجام شده در زمان τ باشد آنگاه $s_{\tau+1} \in S$ حالت بعدی سیستم است و اگر $c_\tau = C(s_\tau, a_\tau)$ هزینه‌ی تحمیل شده در زمان انجام عمل a_τ باشد؛ مسئله‌ی فرآیندهای تصمیم‌گیری مارکوف (MDPs) یافتن سیاست انتخاب عملیات در وضعیت‌های سیستم است به‌طوری‌که این سیاست دنباله‌ای از عملیات مجاز $a_0, a_1, a_2, \dots \in A$ ارائه دهد به‌طوری‌که مقدار $\sum_{\tau=0}^{\infty} c_\tau$ کمینه شود. به بیان دیگر اگر دو سیاست تصمیم‌گیری π و π' وجود داشته باشد وقتی سیاست π از π' بهتر است که رابطه‌های (۱-۲) و (۲-۲) برقرار باشد.

$$\forall s \in S, V_\pi(s) = E \left(\sum_{\tau=0}^{\infty} \gamma^\tau c_\tau \right) | s_0 = s, a_\tau = \pi(s_\tau), c_\tau = C(s_\tau, a_\tau) \quad (۱-۲)$$

$$\forall s \in S, V_\pi(s) \leq V_{\pi'}(s) \quad (۲-۲)$$

در رابطه‌ی (۱-۲) پارامتر γ فاکتور تخفیف هزینه است که مقداری بین صفر تا یک دارد. این فاکتور اثر هزینه‌ی c_τ را برای مقادیر بزرگ‌تر τ کم می‌کند.

۲-۱-۱-۲ حل مسئله‌ی MDPs با برنامه‌نویسی پویا

اگر π^* بهترین سیاست انتخاب عملیات باشد و $V^* = V_{\pi^*}$ در نظر گرفته شود. روابط Bellman مسئله‌ی فرآیندهای تصمیم‌گیری مارکوف را با استفاده از تکنیک برنامه‌نویسی پویا حل می‌کند. رابطه‌ی بازگشتی (۳-۲) نشان می‌دهد اگر بخواهیم مسئله‌ی MDPs را از وضعیت s به‌صورت بهینه‌ی $V^*(s)$ حل کنیم کافی است به ازای تمام عملیات $a \in A_s$ تمام زیر مسئله‌های $V^*(s')$ را به‌صورت بهینه حل کنیم و سپس با استفاده از رابطه‌ی بازگشتی (۳-۲) مسئله‌ی بزرگ‌تر $V^*(s)$ قابل حل است. بعد از به دست آمدن $V^*(s)$ به ازای $\forall s \in S$ تابع $\pi^*(s)$ روش انتخاب عمل مناسب را طبق رابطه‌ی (۴-۲) نشان می‌دهد.

$$V^*(s) = \min_{a \in A_s} \left(C(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^*(s') \right) \quad (۳-۲)$$

$$\forall s \in S, \pi^*(s) = \operatorname{argmin}_{a \in A} \left(C(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^*(s') \right) \quad (۴-۲)$$

۳-۱-۱-۲ حل مسئله‌ی MDPs با یادگیری تقویتی

یکی دیگر از روش‌هایی که مسئله‌ی MDPs را می‌توان حل کرد، استفاده از تکنیک یادگیری تقویتی است. تکنیک یادگیری تقویتی یک چارچوب ساده‌ی یادگیری در تعامل با محیط و رسیدن به هدف را ارائه می‌دهد. در این تکنیک دو عامل یادگیرنده و تصمیم‌گیر وجود دارد و هر آنچه با این دو عامل در تعامل هستند به‌عنوان محیط در نظر گرفته می‌شود. تعامل عامل‌ها با محیط به‌طور مستمر انجام می‌شود و عامل یک عمل^۱ را روی محیط انجام می‌دهد و محیط در پاسخ به آن عمل، واکنش نشان می‌دهد و این واکنش موقعیت جدیدی را برای عامل‌ها به وجود می‌آورد. با گذر زمان محیط یک مقدار عددی را به‌عنوان پاداش یا هزینه برای عامل‌ها در نظر می‌گیرد و عامل‌ها تلاش می‌کنند با انجام عمل مناسب مقادیر پاداش یا هزینه را به ترتیب بیشینه یا کمینه کنند.

برای حل مسئله‌ی MDPs به کمک تکنیک یادگیری تقویتی برای هر عمل $a \in A$ (Action) در هر وضعیت $s \in S$ یک Q-Value در نظر می‌گیریم. با قرار دادن تابع هزینه در رابطه‌ی معرفی شده توسط Barto [۱۱] رابطه‌ی (۵-۲) حاصل می‌شود.

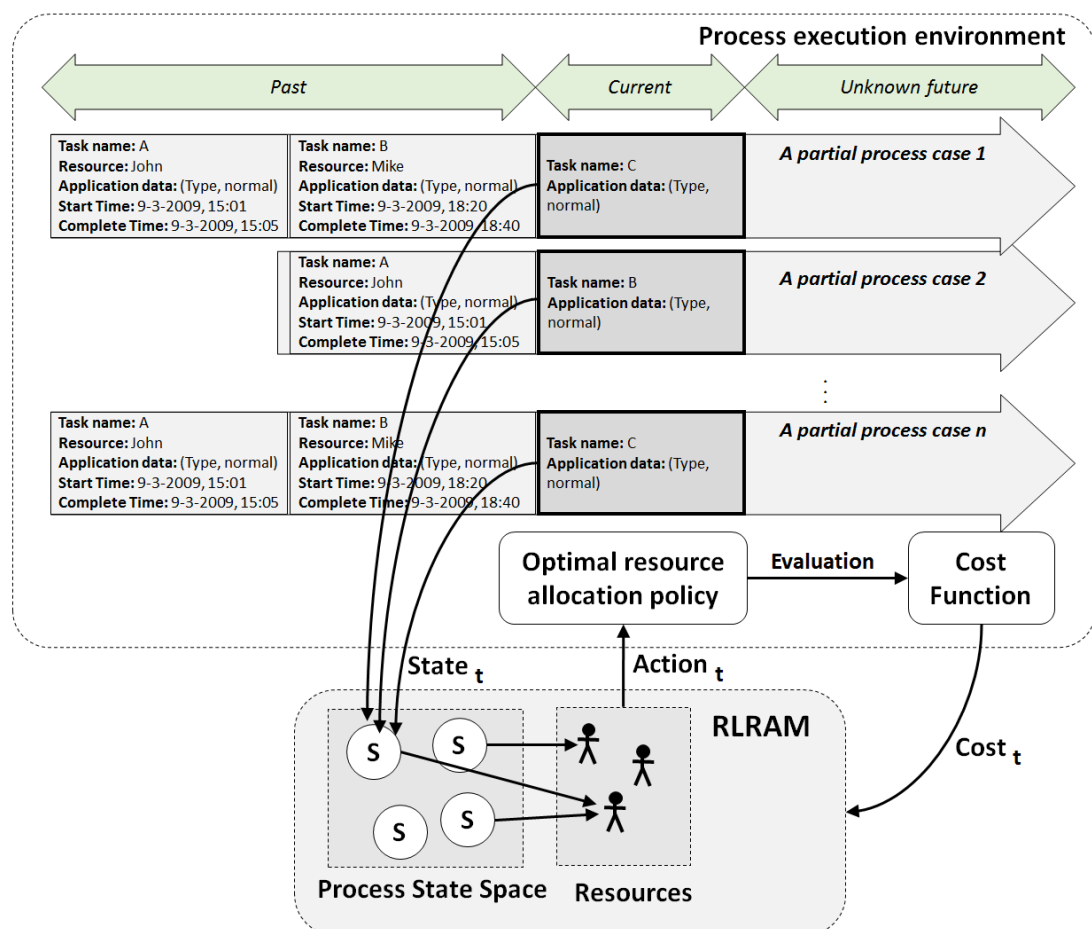
$$Q_{t+1}(s, a) = (1 - \alpha_t)Q_t(s, a) + \alpha_t \left(C(s, a) + \gamma \times \min_{a' \in A_{s'}} Q(s', a') \right) \quad (۵-۲)$$

در رابطه‌ی فوق α_t نرخ یادگیری که مقداری بین صفر و یک دارد و در برخی تحقیقات مقدار $\frac{1}{t}$ برای $t \geq 2$ پیشنهاد کرده‌اند. عبارت $C(s, a) + \gamma \times \min_{a' \in A_{s'}} Q(s', a')$ مقدار یادگیری است که خود شامل هزینه‌ی انجام عمل، ضریب تخفیف γ و تخمینی از مقدار کمینه برای انجام عمل‌های بعدی است.

^۱ Action

۴-۱-۱-۲ تخصیص منابع بر پایه‌ی یادگیری تقویتی

Huang و همکاران [۱۰] با نداشت مسئله‌ی تخصیص منابع در اجرای فرآیندهای سازمانی بر مدل MDPs روشی را بر پایه‌ی یادگیری تقویتی تحت عنوان RL RAM ارائه دادند. در این نگاشت حالات مدل MDPs شامل اطلاعات بارکاری منابع، داده‌های برنامه‌ی کاربردی و ترتیب واحدهای کاری در صف لیست کاری منابع است و هر عمل (Action) با تخصیص واحد کاری در محل مشخصی از لیست کاری منبع رخ می‌دهد. شکل (۵-۲) نمایی کلی از روش RL RAM را نشان می‌دهد. در این روش مقادیر Q-Value از روی گذشته‌ی سیستم تنظیم می‌شود و برای تخصیص منبع و تنظیم لیست کاری منبع در زمان حال استفاده می‌شود.



شکل (۵-۲) نمایی کلی از روش RL RAM

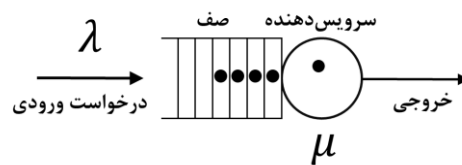
۲-۱-۲ مدل شبکه‌ی صف

مدل شبکه‌ی صف یک روش عمومی برای مدل‌سازی سیستم‌های کامپیوتری است. در این شبکه مجموعه‌ای صف سرویس‌دهی مانند شکل (۶-۲) وجود دارد. وقتی یک فرآیند سازمانی در شبکه‌ی صف مدل می‌شود، هر وظیفه در فرآیند یک ایستگاه سرویس‌دهی است و هر منبع منتسب به این وظیفه، یک سرویس‌دهنده است. از آنجایی که در هر وظیفه w احتمال قرار گرفتن واحد کاری در لیست اختصاصی منبع $\xi_{w,r}$ است، رابطه‌ی (۶-۲) برای تمام سرویس‌دهنده‌های (منابع) تعریف شده در یک ایستگاه سرویس‌دهی (وظیفه) برقرار است.

$$\sum_{r \in R_w} \xi_{w,r} = 1, \forall w \in W \quad (6-2)$$

میزان بارکاری هر ایستگاه سرویس‌دهی را می‌توان با داشتن نرخ ورود درخواست به هر ایستگاه و نرخ سرویس‌دهی به دست آورد. اگر مشتریان در هر صف سرویس‌دهی به چند دسته تقسیم شوند و نرخ ورود دسته i را با λ_i نشان دهیم و میانگین نرخ سرویس‌دهی را با μ_i نشان دهیم. متوسط نرخ بهره‌وری ρ از سرویس‌دهنده‌ها از رابطه‌ی (۷-۲) به دست می‌آید.

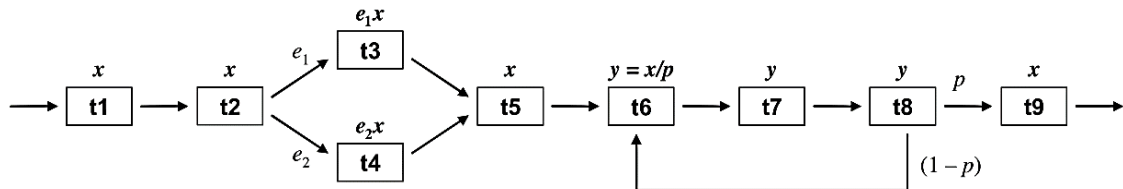
$$\rho = \sum \frac{\lambda_i}{\mu_i} \quad (7-2)$$



شکل (۶-۲) صف تک سرویس‌دهنده

در یک صف تک سرویس‌دهنده وقتی متوسط نرخ ورود λ باشد و متوسط نرخ سرویس‌دهی μ باشد. متوسط نرخ خروجی وابسته به مقدار متوسط نسبت λ/μ یا همان مقدار متوسط نرخ بهره‌وری ρ است. اگر نسبت λ/μ بزرگ‌تر از یک باشد آنگاه طول صف افزایش می‌یابد و متوسط نرخ خروجی برابر با نرخ سرویس‌دهی می‌شود. اگر نسبت λ/μ کوچک‌تر از یک باشد آنگاه متوسط نرخ خروجی برابر با متوسط نرخ ورودی می‌شود. در مدل‌سازی با شبکه‌ی صف همیشه فرض می‌شود متوسط مقدار λ/μ برای هر صف در شبکه کمتر از یک است تا شبکه پایدار باشد. با تعمیم روابط بین متوسط نرخ ورود،

متوسط نرخ سرویس‌دهی و متوسط نرخ خروجی به شبکه‌ی صف می‌توان متوسط نرخ ورود را در هر بخشی از شبکه‌ی صف محاسبه کرد. شکل (۷-۲) با یک مثال روش محاسبه‌ی نرخ ورود را در بخش‌های مختلف شبکه بر اساس ساختار شبکه نشان می‌دهد.



شکل (۷-۲) یک مثال از محاسبه‌ی متوسط نرخ ورود درخواست در هر یک از وظایف در شبکه‌ی صف [۴]

بعد از محاسبه‌ی متوسط نرخ ورود درخواست در هر وظیفه، اگر آن را با λ_w نشان دهیم با مفروض بودن مقادیر $\xi_{w,r}$ (احتمال تخصیص وظیفه‌ی w به منبع r) می‌توان نرخ ورود درخواست را برای هر منبع از رابطه‌ی (۸-۲) محاسبه کرد. اگر μ_r متوسط زمان سرویس‌دهی منبع r باشد، می‌توان مقادیر $\xi_{w,r}$ را طوری تعیین کرد که یکی از دستگاه معادلات (۹-۲) و (۱۰-۲) برقرار باشد. اگر دستگاه معادله‌ی (۹-۲) برقرار باشد در توزیع درخواست‌های ورودی سیستم توزیع عادلانه‌ای انجام داده است و اگر دستگاه معادله‌ی (۱۰-۲) برقرار باشد توزیع درخواست‌های در جهت برقراری تعادل بارکاری منبع خواهد بود.

$$\lambda_r = \sum_{w \in W \text{ and } r \in R_w} \xi_{w,r} \lambda_w \quad (۸-۲)$$

$$\lambda_{r_1} = \lambda_{r_2} = \lambda_{r_3} \dots = \lambda_{r_n}, r_i \in R \quad (۹-۲)$$

$$\frac{\lambda_{r_1}}{\mu_{r_1}} = \frac{\lambda_{r_2}}{\mu_{r_2}} = \frac{\lambda_{r_3}}{\mu_{r_3}} \dots = \frac{\lambda_{r_n}}{\mu_{r_n}}, r_i \in R \quad (۱۰-۲)$$

۳-۱-۲ مدل WF-Net

برای مدل‌سازی یک فرآیند ابزارها و روش‌های مدل‌سازی متنوعی وجود دارد. ساده‌ترین آن‌ها فلوچارت است و مواردی مانند چارت حالات^۱، EPCs^۲، نمودار فعالیت^۳ در UML، زنجیره مارکوف^۴،

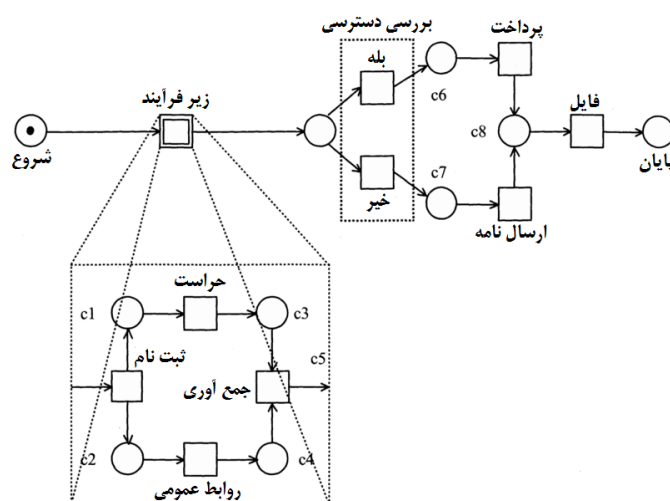
^۱ State chart

^۲ Activity diagram

^۳ Event-driven process chains

^۴ Markov chains

جبر پردازش‌ها^۱، BPMN^۲، Petri-net، WF-net^۳، EWF-net^۴ و YAWL^۵ را می‌توان نام برد. اما انتخاب یک ابزار مدل‌سازی مناسب که هم از نظر تحلیلی و تئوری پشتیبانی ریاضی قوی داشته باشد و هم از نظر گرافیکی قابل فهم و خوانا باشد کمی مشکل است. مهم‌ترین مفهوم در جریان کاری، فرآیند است. عبارت فرآیند در سیستم‌های مدیریت جریان کاری^۶ به روش اداره کردن مجموعه‌ای پرونده‌ها از یک نوع خاص گفته می‌شود. برای به سرانجام رساندن یک پرونده وظایفی باید انجام شود و این اطلاعات در فرآیند قرار دارد. علاوه بر این یک فرآیند شامل اطلاعاتی در مورد شرایط و ترتیب اجرای این وظایف است. یک فرآیند می‌تواند بخشی از یک فرآیند دیگر باشد و حتی این امکان وجود دارد که یک فرآیند شامل یک یا چند زیرفرآیند^۷ باشد (شکل (۸-۲)).



شکل (۸-۲) استفاده از یک زیرفرآیند در فرآیند

برای مدل کردن و نمایش یک فرآیند از Petri-net استفاده می‌شود. البته نوع کلاسیک Petri-net دارای مکان‌هایی به عنوان شروع (ورود^۸) و پایان (خروج^۹) نیست و در نوع مشتق شده از آن با عنوان Workflow-net (به اختصار WF-net) شامل این دو مکان است. شرایط و وظایف در فرآیند به ترتیب به شکل مکان^{۱۰}ها و انتقال^{۱۱}ها در WF-net مدل می‌شوند. به همین ترتیب مفهوم پرونده نیز در

^۱ Process Algebras

^۲ Business Process Model and Notation

^۳ Workflow-net

^۴ Extended Workflow-net

^۵ Yet Another Workflow Language

^۶ Workflow Management Systems

^۷ Sub-process

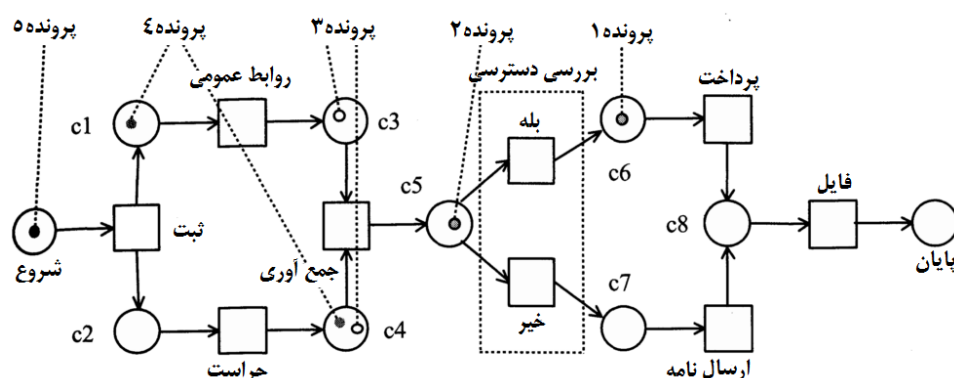
^۸ Entrance: a place without incoming arcs

^۹ Exit: a place without outgoing arcs

^{۱۰} Place

^{۱۱} Transition

WF-net به شکل توکن^۱ مدل می‌شود. در شکل (۹-۲) یک فرآیند «ثبت‌نام» در یک WF-net مدل شده است. در این فرآیند وضعیت پنج پرونده به صورت مجموعه از توکن‌ها نمایش داده شده است. به عنوان مثال برای پرونده‌ی ۱، پرونده‌ی ۲ و ۵ به ترتیب سه توکن در وضعیت‌های c6، c5 و وضعیت شروع قرار گرفته است و برای پرونده‌ی ۳ دو توکن در وضعیت‌های c4 و c3 و به طور مشابه برای پرونده‌ی ۴ نیز دو توکن در وضعیت‌های c4 و c1 نشان دهنده‌ی وضعیت پرونده در فرآیند است.



شکل (۹-۲) نگاشت مفاهیم فرآیند در WF-net [۱۲]

روش و ترتیب انجام فعالیت‌ها در فرآیند سازمانی به نوع و اطلاعات پرونده وابسته است. این بدان معناست، وقتی پرونده‌ها دارای نوع و اطلاعات مشابهی باشند در اجرا از رویه‌ی یکسانی که در فرآیند تعریف شده است پیروی می‌کنند [۱۲]. البته مدل فرآیند به صورت انتزاعی تعریف می‌شود و اجرای فرآیند با وجود یک پرونده معنی‌دار می‌گردد.

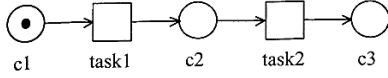
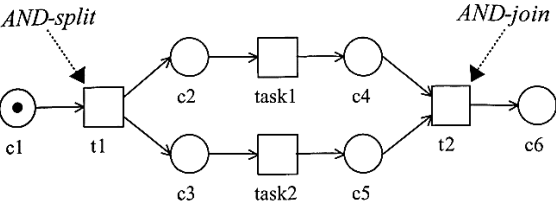
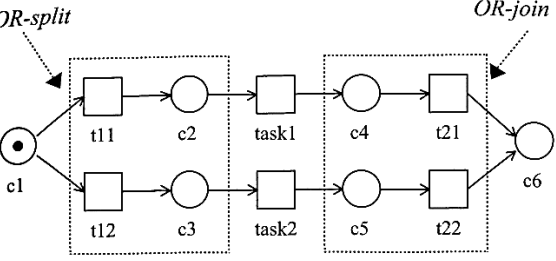
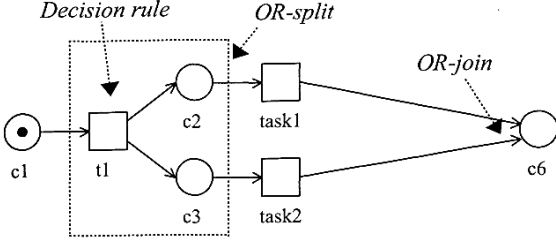
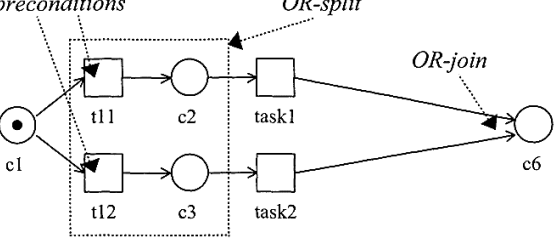
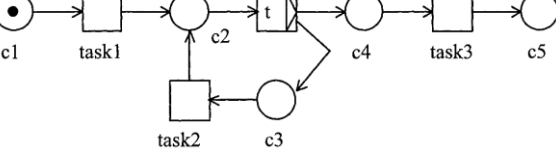
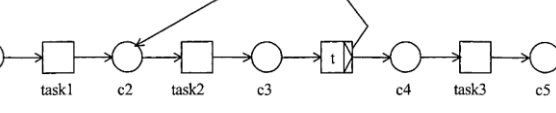
۱-۳-۱-۲ مسیریابی در فرآیند سازمانی

دوره‌ی عمر یک پرونده در طول اجرای یک فرآیند گسترده شده است و از آنجایی که فرآیند در قالب یک گراف مدل می‌شود، مسیر اجرای فرآیند برای یک پرونده‌ی خاص قابل توجه است و این مفهوم در کلمه‌ی مسیریابی^۲ یا مسیر/اجرای پرونده خلاصه می‌شود. ساده‌ترین شکل مسیریابی یک پرونده اجرای ترتیبی فعالیت‌ها است. جدول (۱-۲) انواع مختلف مسیریابی یک پرونده در فرآیند را نشان می‌دهد.

^۱ Token

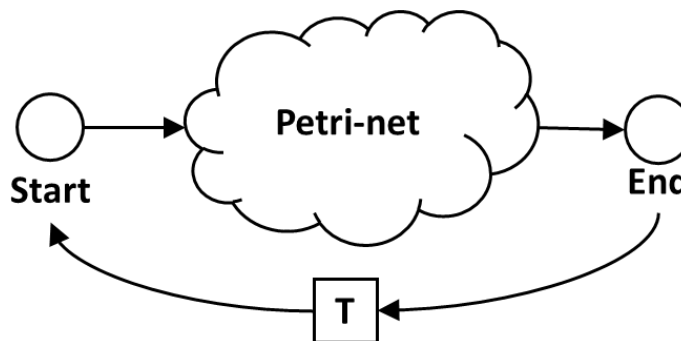
^۲ Routing

جدول (۱-۲) انواع مسیریابی پرونده در فرآیندها [۱۲].

مسیریابی	نمایش در EWF-net	توضیحات
توالی		
موازی		در این مسیریابی هم task1 و هم task2 روی پرونده اجرا می‌شود و توکن بعد از AND-split به صورت هم‌زمان در دو وضعیت c2 و c3 قرار می‌گیرد
انتخابی		در این مسیریابی فقط یکی از دو task1 یا task2 روی پرونده اجرا می‌شود
انتخابی باقاعده تصمیم‌گیری		در این نوع مسیریابی در t1 قاعده تصمیم‌گیری روی پرونده اجرا شده و سپس توکن در یکی از دو وضعیت c2 یا c3 قرار می‌گیرد و در نتیجه فقط یکی از دو task1 یا task2 روی پرونده اجرا می‌شود
انتخابی با پیش‌شرط‌ها		در این نوع مسیریابی در وضعیتی که توکن در حالت c1 قرار دارد، پیش‌شرط‌های t11 و t12 بررسی شده و توکن حداقل در یکی از دو حالت c2 و c3 قرار می‌گیرد.
تکراری		در این نوع مسیریابی task2 به صورت تکراری بر روی پرونده قابل اجرا است.
تکراری		در این نوع مسیریابی task2 به صورت تکراری بر روی پرونده قابل اجرا است و حداقل یک بار اجرا می‌شود.

۲-۳-۱-۲ درستی در WF-Net

Aalst و همکاران در مقاله‌ای که در سال ۲۰۱۱ ارائه دادند، مفهوم درستی^۱ را به عنوان یک معیار لازم برای مدل سازی فرآیندهای سازمانی بیان کردند و نشان دادند برای یک ابزار مدل سازی داشتن این خاصیت الزامی است و اثبات کردند که WF-net در صورتی این خاصیت را دارا است که Petri-net ایجاد شده مطابق با شکل (۱۰-۲) کران دار^۲ و زنده^۳ باشد. در شکل (۱۰-۲) ساختار WF-net با یک حالت انتقال که مکان پایانی را به مکان شروع وصل می کند، نشان داده شده است. در صورتی WF-net خاصیت درستی را خواهد داشت که Petri-net ایجاد شده کران دار و زنده باشد [۱۳].



شکل (۱۰-۲) یک انتقال حلقوی جهت اثبات درستی WF-Net

Aalst همچنین سه خاصیت دیگر که در Petri-net کلاسیک وجود نداشت را معرفی می کند و برای رفع این محدودیت ها نوع های توسعه یافته ای از WF-net را با عناوین EWF-net و YAWL را ارائه می دهد که سه خاصیت ذکر شده در جدول (۲-۲) را داشته باشد [۸، ۱۴].

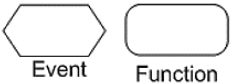
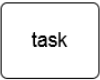
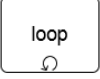
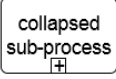
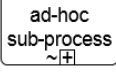
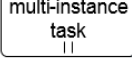
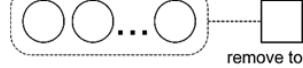
از مجموعه ای ابزارهای فوق الذکر که به طور خاص برای مدل سازی فرآیند طراحی شده اند می توان مهم ترین آن ها را در جدول (۲-۲) مورد مقایسه قرار داد. همان طور که در این جدول مشخص شده است. معیارهایی که مورد بررسی قرار گرفته اند عبارت اند از درستی، هم زمانی پیشرفته، پشتیبانی از الگوهای چند نمونه ای و الگوهای لغو درخواست.

^۱ Soundness

^۲ Live

^۳ Bounded

جدول (۲-۲) مقایسه‌ی ابزارهای مدل‌سازی فرآیند

علائم Notations	پشتیبانی از الگوهای			درستی Soundness [۱۳]	ابزارهای مدل‌سازی
	لغو Cancellation [۱۴]	چند نمونه‌ای multiple instances [۱۴]	همزمان‌سازی پیشرفته Advanced synchronization [۱۴]		
   Event Function Connectors Control Flow	ندارد	ندارد	ندارد	وابسته به طراحی	EPCs
Activites      Events       Gateways    message error time start event intermediate event end event AND XOR Event-based	ندارد	دارد	دارد	وابسته به طراحی	BPMN
  Place Transition	ندارد	ندارد	ندارد	وابسته به طراحی	Petri-net
    Place Transition Start End	ندارد	ندارد	ندارد	وابسته به طراحی	WF-Net
              Condition Atomic task Input condition Composite task Output condition Multiple instances of an atomic task Multiple instances of a composite task AND-split AND-join XOR-split XOR-join OR-split OR-join remove tokens	دارد	دارد	دارد	دارد	EWf-net
مانند EWF-net	دارد	دارد	دارد	دارد	YAWL

۲-۲ از دیدگاه هدف

یکی از تکنیک‌های افزایش پایداری در BPMS، تعادل بار کاری منابع است، مخصوصاً در مواقعی که تعداد منابع سرویس‌دهنده محدود است. این تکنیک در زمینه‌های مختلف در صنعت و علوم کامپیوتر مورد استفاده قرار می‌گیرد. به عنوان مثال، می‌توان به سیستم‌های توزیع شده و محاسبات موازی [۱۵]، محاسبات ابری [۱۶] و سیستم‌های مدیریت پایگاه داده [۱۷] اشاره کرد. از این تکنیک در سیستم‌های مدیریت جریان کاری نیز بارها استفاده شده است [۳، ۴، ۱۸-۲۳].

به صورت کلی، نحوه‌ی تخصیص وظیفه به منابع، عامل اصلی در کارایی فرآیندهای کسب و کار است. از ساده‌ترین تکنیک‌های ارائه شده می‌توان ^۱FIFO، ^۲SPT، ^۳EDD و ^۴MST را نام برد [۲۴]. مزیت این روش‌های پایه‌ای در این است که به سادگی آن‌ها را می‌توان در هر سیستمی مورد استفاده قرار داد یا با روش‌های پیچیده‌تر دیگر ترکیب کرد. عامل دیگری که در کارایی فرآیندهای کسب و کار مؤثر است، استفاده از توانایی و تخصص منابع در زمان تخصیص وظایف است [۲۵، ۲۶]، این رویکرد به طور غیرمستقیم تلاش می‌کند با توجه به نقش منابع، تخصیص وظایف به هر منبع را اولویت‌بندی کند که در مجموع باعث بهتر شدن عملکرد الگوریتم تخصیص در انتخاب منبع مناسب می‌شود ولی باعث کاهش انعطاف‌پذیری الگوریتم تخصیص وظیفه در انتخاب آزادانه منابع می‌گردد.

۲-۲-۱ کاهش چرخه‌ی زمانی

مهم‌ترین هدفی که ردپای آن در بسیاری از تحقیقات دیده شده است، کاهش چرخه‌ی زمانی درخواست‌ها در فرآیندهای سازمانی است [۳، ۴، ۱۸-۲۳، ۲۷]. Liu و همکاران [۲۷] با تحلیلی که روی پایگاه داده‌ی واقعی BPI Challenge 2012 انجام دادند، توانستند بین منابع نوعی ارتباط اجتماعی تعریف کنند. این ارتباط اجتماعی بر اساس یک تحلیل آماری بین منبع کار جاری یک درخواست و منابع کارهای قبلی همان درخواست، تعریف شد. آن‌ها با تغییر در الگوریتم تخصیص منابع که

^۱ First In First Out

^۲ Earliest Due Date

^۳ Shortest Processing Time

^۴ Minimum Slack Time

پارامترهای آن بر اساس ارتباط اجتماعی بین منابع با یک سیستم یادگیری تقویتی تنظیم شد. ایده‌ی خود را مورد آزمایش قرار دادند و توانستند چرخه‌ی زمانی درخواست‌ها را روی پایگاه داده‌ی واقعی BPI Challenge 2012 با ۵۳ درصد کاهش بهبود دهند. همچنین آن‌ها روش خود را روی یک پایگاه داده شبیه‌سازی شده مورد آزمایش قرار دادند که با کاهش ۴۰ درصدی نشان دادند که استفاده از ارتباط اجتماعی منابع می‌تواند در کاهش چرخه‌ی زمانی درخواست مؤثر باشد.

در تحقیقی که Liu و همکاران انجام دادند، ارتباط اجتماعی^۱ بین منابع انسانی مورد توجه قرار گرفت و در اعمال ضرایب یادگیری تقویتی در فرآیند تصمیم‌گیری مارکوف از ضریب ارتباط اجتماعی استفاده شد که باعث بهبود عملکرد الگوریتم تخصیص منبع در جهت کاهش چرخه‌ی زمانی توکن‌ها شد. در این تحقیق ارتباط اجتماعی بین منبع تخصیص‌یافته به واحد کاری با منابع قبلی که روی آن پرونده فعالیت انجام داده‌اند به صورت ضریب SR تعریف شد و این ضریب از روی متوسط زمان انجام فعالیت روی پرونده‌ی مشترک و متوسط زمان انجام فعالیت، مستقل از پرونده محاسبه شده در رابطه‌ی (۱۱-۲) روش محاسبه‌ی ضریب SR آمده است. در این معادله t_{r_2, r_1} متوسط زمان فعالیت‌هایی است که منبع r_2 انجام داده است مشروط به اینکه روی آن پرونده‌ی r_1 قبلاً فعالیت انجام داده است و t متوسط زمان انجام فعالیت تخصیص‌یافته شده به r_2 مستقل از منابع دیگر است [۲۷].

$$SR_{r_1, r_2} = \frac{t_{r_2, r_1} - t}{t} \quad (11-2)$$

اگر مقدار رابطه‌ی فوق منفی باشد بیانگر تخصیص خوب است که باعث کاهش زمان انجام فعالیت و در نتیجه باعث کاهش چرخه‌ی زمانی پرونده می‌شود و اگر این مقدار مثبت باشد بیانگر تخصیص بد فعالیت است و بهتر است پارامترهای الگوریتم یادگیری در جهتی تنظیم شود که این تخصیص با احتمال کم‌تری انجام شود. بر اساس رابطه‌ی (۱۱-۲) می‌توان روابط ذیل را به دست آورد [۲۷].

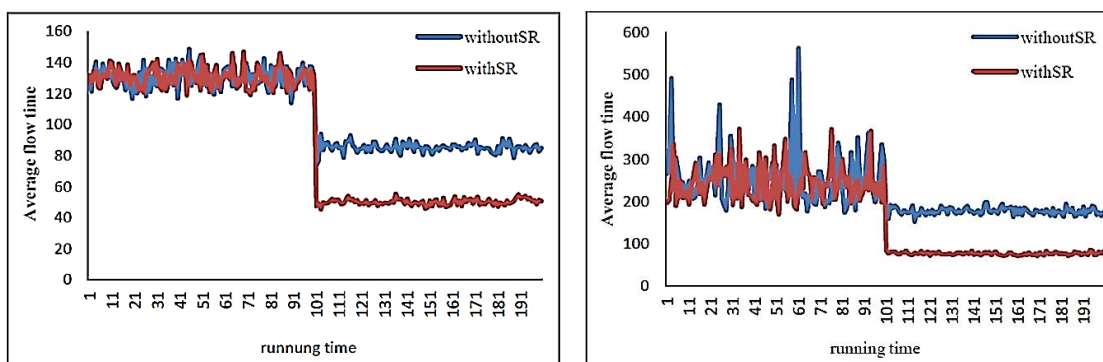
$$\begin{aligned} t_{r_3, r_2} &= (1 + SR_{r_3, r_2}) \times t \\ t_{r_3, r_1} &= (1 + SR_{r_3, r_1}) \times t \end{aligned} \quad (12-2)$$

^۱ Social Relation

$$t_{r_3, r_2, r_1} = \frac{t_{r_3, r_2} + t_{r_3, r_1}}{2} = \left(1 + \frac{SR_{r_3, r_2} + SR_{r_3, r_1}}{2}\right) \times t \quad (13-2)$$

$$t_{r_n, r_{n-1}, r_{n-2}, \dots, r_2, r_1} = \left(1 + \frac{1}{n-1} \sum_{i=1}^{n-1} SR_{r_n, r_i}\right) \times t \quad (14-2)$$

Liu و همکاران برای محاسبه‌ی متوسط زمان اجرای فعالیت تخصیص یافته به منبع r_n که منابع دیگر در پرونده‌ی مربوط به آن فعالیت داشته‌اند، از رابطه‌ی (۱۴-۲) استفاده کرده‌اند. آن‌ها در این تحقیق از یک پایگاه داده‌ی واقعی^۱ BPI Challenge 2012 استفاده کردند و برای اعمال ضریب ارتباط اجتماعی در پایگاه داده از شبیه‌سازی استفاده کردند و نتایج را روی هر دو پایگاه داده به دست آوردند در شکل (۱۱-۲) نتیجه‌ی استفاده از تکنیک یادگیری تقویتی و ارتباط اجتماعی منابع را در کاهش چرخه‌ی زمانی نشان می‌دهد. در این آزمایش ۲۰۰ پرونده به‌عنوان ورودی فرآیند در نظر گرفته شده است. ۱۰۰ پرونده‌ی اول با الگوریتم تخصیص منبع به‌صورت تصادفی به منابع تخصیص داده شده است و ۱۰۰ پرونده‌ی بعدی با الگوریتم یادگیری تقویتی (با هدف چرخه‌ی زمانی کمینه) منبع مناسب انتخاب شده است. این آزمایش با در نظر گرفتن ارتباط اجتماعی منابع تکرار شد و اثر هر دو تکنیک استفاده از یادگیری تقویتی و ارتباط اجتماعی منابع در انتخاب منبع مناسب در شکل (۱۱-۲) مشهود است.



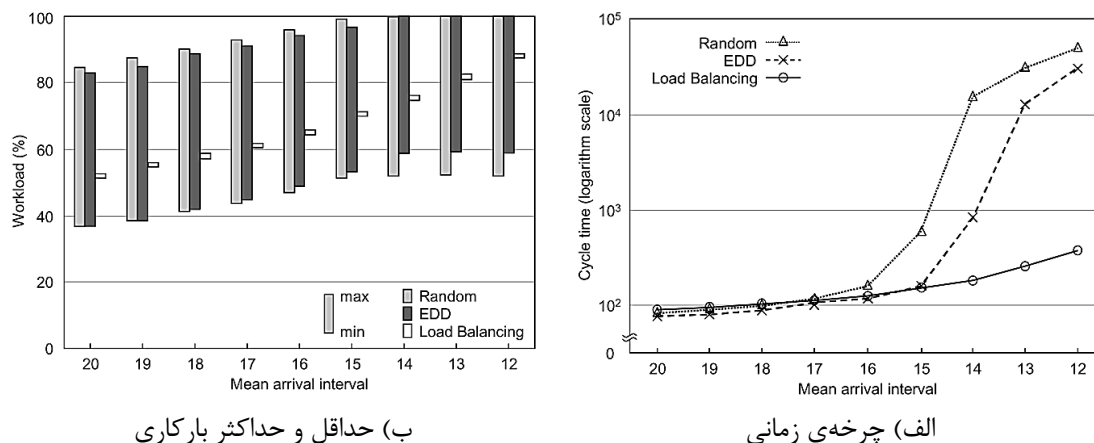
(ب) پایگاه داده‌ی شبیه‌سازی

(الف) پایگاه داده‌ی واقعی

شکل (۱۱-۲) نتایج الگوریتم یادگیری تقویتی با در نظر گرفتن ارتباط اجتماعی برای ۲۰۰ بار اجرا [۲۷]

^۱ <http://www.win.tue.nl/bpi/2012/challenge>

در مقاله‌ی ارائه شده توسط Byung-Hyun Ha و همکاران [۳]، یک الگوریتم تخصیص وظیفه به منبع انسانی با هدف ایجاد تعادل بارکاری معرفی شد. الگوریتم ارائه شده بر روی مدل شبکه‌ی صف^۱ اجرا شد؛ و نتایج آن با روش‌های اصلی تخصیص تصادفی و EDD^۲ مقایسه شد و نشان داده شد علاوه بر ایجاد تعادل بارکاری در کل سیستم با افزایش حجم درخواست‌های ورودی، چرخه‌ی زمانی درخواست‌ها کم‌تر افزایش پیدا کرده است این نتیجه در شکل (۲-۱۲) نشان داده شده است.



شکل (۲-۱۲) مقایسه‌ی چرخه‌ی زمانی و میزان تعادل بارکاری سه الگوریتم پایه [۳]

در راستای هوشمندسازی الگوریتم تخصیص وظایف به منبع، محققان بسیاری تلاش کردند با استفاده از الگوریتم‌های تکاملی مانند الگوریتم ژنتیک، مسئله‌ی تخصیص وظایف به منابع را حل کنند یا بهبود دهند [۲۸-۳۰]. به عنوان مثال Debby و Chan در پایان‌نامه کارشناسی ارشد خود، یک روش تخصیص وظایف به منابع بر پایه‌ی الگوریتم ژنتیک ارائه کرد. در این تحقیق برای بهینه‌سازی زمان اجرای فرآیند، قیدهایی پیشنهاد شده است، زمان میانگین نمونه‌ها در فرآیند اندازه‌گیری شد و یک روش برای محاسبه‌ی میانگین زمان پاسخ پرونده‌ها در فرآیند پیشنهاد شد و یک الگوریتم ژنتیک بهبودیافته برای تخصیص منابع ارائه گردید. نتایج آزمایشگاهی نشان می‌دهد که الگوریتم پیشنهادی آن‌ها عملکردی خوبی نسبت به روش‌های دیگر داشت [۳۱].

^۱ Queuing network model

^۲ Earliest Due Date

Xu و همکاران [۲۱] با ارائه‌ی الگوریتم تخصیص منابع، هزینه‌ی اجرای فرآیند سازمانی را با در نظر گرفتن محدودیت زمانی، کاهش دادند. به‌طور مشابه در مطالعات Nisafani و همکاران [۲۲] برای انتخاب منبع قاعده‌ای شامل چندین پارامتر ارائه دادند. آن‌ها پارامترهای بارکاری منابع، اندازه صف، ساعت کاری منابع در طول روز و میزان حجم درخواست در ساعات مختلف روز را در تعریف قاعده‌ی انتخاب منبع مورد توجه قرار دادند سپس این پارامترها را در یک شبکه‌ی بیز^۱ ترکیب کردند و یک مدل واحد، به‌عنوان «قاعده‌ی انتخاب بیز^۲» برای انتخاب منابع در فرآیندهای کسب‌وکار ارائه دادند. آزمایش‌های آن‌ها نشان داد، این مدل به‌طور قابل‌توجهی عملکرد سیستم را بهبود بخشیده است. در ادامه‌ی مطالعات Nisafani، Wibisono و همکاران [۲۳] ایده‌ی مشابهی را مورد آزمایش قرار دادند و الگوریتم تخصیص منبع دیگری برای مدیریت منابع در BPMS ارائه دادند. آن‌ها همان ایده‌ی قبلی Nisafani را گسترش دادند و با استفاده از مدل بیز خام^۳ قاعده انتخاب منبع^۴ NBSR را ارائه کردند. در تحقیق انجام شده توسط Kumar و همکاران [۳۲] یک روش جدید و پیچیده برای توزیع وظایف ارائه شد که در حال حاضر در بسیاری از سیستم‌های مدیریت جریان کاری^۵ مورد استفاده قرار می‌گیرد. در مقایسه با سیستم‌های قدیمی‌تر که از تکنیک‌های ضعیف‌تر استفاده می‌کنند روش ارائه‌شده‌ی ایشان به‌عنوان انقلابی در دستیابی به کیفیت و کارایی به حساب می‌آید.

Narahari و همکاران [۳۳] با استفاده از مدل‌سازی صف در BPMS و بهره‌گیری از تکنیک‌های شبیه‌سازی توانستند برای کاهش چرخه‌ی زمانی با استفاده از تجزیه و تحلیل مدل صف روش‌های مختلفی به دست آورند. Son و Kim [۳۴] با مدل‌سازی هر وظیفه در فرآیند کسب‌وکار به‌عنوان یک سرویس‌دهنده صف، با توجه به زمان اتمام اجرای فرآیند توانستند روشی برای محاسبه‌ی ظرفیت هر منبع ارائه دهند. Chang و همکاران [۳۵] روش‌های شناسایی مسیر بحرانی در فرآیندهای کسب‌وکار

^۱ Bayesian Network

^۴ Naïve Bayes Selection Rule

^۲ Bayesian Selection Rule

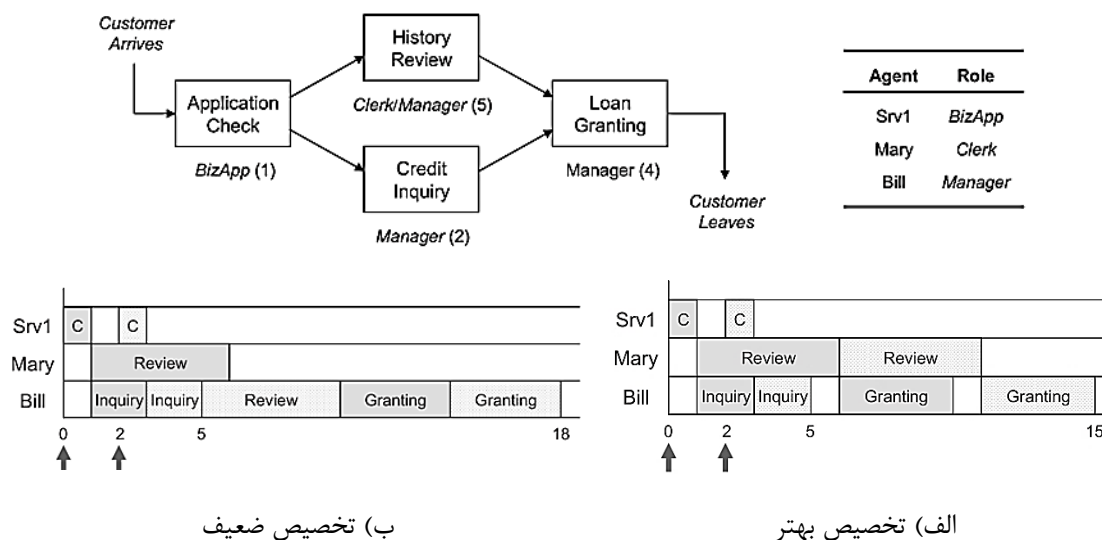
^۵ Workflow management systems

^۳ Naïve Bayes Model

را مورد مطالعه قرار دادند. شبکه‌ی جریان کاری تصادفی^۱ (SWN) یکی دیگر از مدل‌هایی است که به صورت گسترده توسط محققان مورد استفاده قرار گرفته است. این مدل بر اساس مبانی نظری خوبی تعریف شده است [۳۶-۳۸]. SWN هم‌ارز یک شبکه‌ی پتری تصادفی نامحدود است که برای مدل‌سازی جریان کاری مورد استفاده قرار می‌گیرد و نتایج تجزیه و تحلیل آن در فرآیندهای کسب و کار قابل استناد است.

۲-۲-۲ تعادل بارکاری منابع

از آنجایی که تعادل بارکاری در شرایط بار سنگین باعث کاهش چرخه‌ی زمانی فرآیندها می‌شود [۳]، ظرفیت و توانایی هر منبع برای انجام وظیفه‌اش عامل محدودکننده است، با توجه به اینکه در سازمان‌ها بیش‌تر فعالیت‌ها توسط منابع انسانی انجام می‌شود. زمان‌بندی فرآیندها به گونه‌ای که حجم کاری آن‌ها روی هم قرار نگیرد و از طرفی به چرخه‌ی زمانی هر فرآیند کم‌ترین آسیب را تحمیل کند، یک مسئله‌ی مشکل است [۴].



شکل (۲-۱۳) تخصیص بهتر علاوه بر ایجاد تعادل در بارکاری باعث کاهش چرخه‌ی زمانی شده است [۳].

^۱ Stochastic Workflow Net

در برخی از تحقیقات، تعادل بارکاری منابع به عنوان هدف اصلی تحقیق در نظر گرفته شده است. در این تحقیقات با استفاده از تئوری صف و مدل سازی فرآیندهای کسب و کار در شبکه‌ی صف^۱، الگوریتم‌های هوشمندی جهت تخصیص منابع ارائه شده است به طوری که تعادل بارکاری منابع را اولویت اصلی خود قرار داده‌اند. تحقیقی که به وسیله Byung-Hyun Ha و همکاران [۳] انجام شد، نشان داد ایجاد تعادل در بارکاری منابع باعث افزایش پایداری سیستم می‌شود. Byung-Hyun Ha و همکاران در تحقیق دیگری [۴]، الگوریتمی جهت تعادل بارکاری منابع در BPMS ارائه کردند، آن‌ها یک مدل تحلیلی بر اساس داده‌های ذخیره شده در سابقه‌ی رویدادها ارائه دادند، سپس با استفاده از مدل تحلیلی خود و ترکیب آن با شبکه‌ی صف توانستند روشی رسمی (فرمال) برای ایجاد تعادل بارکاری منابع ارائه دهند، آن‌ها روش خود را به کمک شبیه سازی فرآیندهای کسب و کار مورد ارزیابی قرار دادند.

در تحقیق مشابهی که توسط Larbi و همکاران [۵] انجام شد. برای حل مسئله‌ی زمان بندی روی ماشین‌های موازی همسان از ایجاد تعادل بارکاری روی ماشین‌های موازی استفاده شد. آن‌ها برای ارزیابی روش پیشنهادی خود، از یک نگاشت بر روی شبکه‌ی پتری استفاده کردند و گزارش‌های آن‌ها نشان داد، تعادل بارکاری عملکرد خوبی در پایداری سیستم دارد. Huang و همکاران [۱۰] با ارائه‌ی یک مدل جامع بر پایه‌ی یادگیری تقویتی توانستند یک تابع هزینه برای الگوریتم تخصیص منابع تعریف کنند به طوری که با تعریف یک تابع هزینه مناسب برای تعادل بارکاری منابع در مدل ارائه شده‌ی آن‌ها می‌توان از مدل تعاملی و هوشمند در ایجاد تعادل بارکاری منابع بهره برد.

در بسیاری از تحقیقات انجام شده از مدل‌های تصادفی به عنوان مدلی جهت تحلیل فرآیندهای کسب و کار استفاده می‌شود. این تحقیقات نشان داده است که مدل‌های تصادفی به خوبی برای مقاصد مختلف قابل استفاده هستند. گراف زمانی احتمالاتی^۲ که بر پایه‌ی رویدادهای محتمل در فرآیندهای کسب و کار بنا شده است، نشان داد که می‌تواند تحلیل بهتری از عملکرد منابع داشته باشد، در نتیجه

^۱ Queueing network

^۲ Probabilistic Timed Graph

مدیریت بهتری از زمان‌بندی منابع ارائه می‌دهد [۳۹]. Xie و همکاران [۶] در تکمیل تحقیق Byung-Hyun Ha و همکاران [۴] روش آن‌ها را گسترش دادند و توانستند با ترکیب نظریه‌ی صف و مدل‌های تصادفی، روش بهتری به‌عنوان «تخصیص تصادفی پویا با هدف تعادل بارکاری»^۱ (DSWB) جهت تخصیص وظایف ارائه دهند. ایشان با اضافه کردن یک پارامتر جدید به شبکه‌ی صف، تخمین بهتری از چرخه‌ی زمانی ارائه دادند و با جابجایی واحدهای کاری در صف اختصاصی (لیست کاری) منابع تعادل بارکاری منابع را افزایش دادند و هم‌زمان چرخه‌ی زمانی درخواست‌ها را بهبود بخشیدند. ایده‌ی اصلی آن‌ها در کاهش بارکاری منابع با جابجایی واحدهای کاری، از منابع با بارکاری بالا به منابع غیرفعال است.

^۱ Dynamic Stochastic task allocation approach for Workload Balancing

فصل ۳ – توسعه‌ی مدل MDPs و ارائه‌ی مدل پیشنهادی

فرآیندهای سازمانی و کاستی‌های مدل فرآیندهای تصمیم‌گیری مارکوف
ارائه‌ی مدل پیشنهادی و مسئله‌ی تخصیص منابع
استفاده از یادگیری تقویتی در حل مدل پیشنهادی

همان‌طور که در سابقه‌ی تحقیق اشاره شد برای مدل‌سازی اجرای فرآیندهای سازمانی سه مدل تحلیلی وجود دارد. (۱) مدل MDPs که برای حل آن از تکنیک بهینه‌سازی پویا و یادگیری تقویتی استفاده می‌شود، (۲) مدل تحلیلی منطبق بر تئوری صف، در این روش مسئله‌ی اجرای فرآیندهای سازمانی در یک شبکه‌ی صف مدل می‌شود و (۳) مدل شبکه‌ی جریان کاری (WF-net) به‌عنوان یک توسعه از شبکه‌ی پتری. مدل تحلیلی شبکه صف با فرض محدودیت‌هایی در اجرای فرآیندهای سازمانی به‌صورت تحلیلی قابل حل است ولی در شرایطی که محدودیت برای الگوهای صف‌بندی و توزیع آماری مدت انجام فعالیت وجود نداشته باشد، مدل به‌صورت تحلیلی قابل حل نیست. اما از مزایای مدل شبکه‌ی صف امکان تعریف برخی از الگوهای صف‌بندی، توزیع آماری مدت انجام فعالیت، توزیع آماری احتمال حضور منابع است. در حالی که در مدل MDPs فقط احتمال رخ دادن حالت بعدی از روی حالت قبلی و عملی (Action) که انتخاب شده است قابل تعریف است. WF-net در پیشینه‌ی تحقیق به عنوان ابزاری جهت طراحی مدل فرآیند استفاده شده است و کم‌تر برای مدل‌سازی اجرای فرآیند استفاده شده است. در این فصل مدل MDPs را توسعه می‌دهیم و مدل جدیدی را تحت عنوان $XMDPs^1$ معرفی می‌کنیم تا انطباق بیش‌تری با مسئله‌ی اجرای فرآیندهای سازمانی بخصوص مسئله‌ی تعادل و یکنواختی بارکاری (مسئله‌ی مطرح شده در این رساله) داشته باشد.

۳-۱ مدل پیشنهادی $XMDPs$

مدل توسعه‌یافته‌ی فرآیندهای تصمیم‌گیری مارکوف (مدل پیشنهادی) را می‌توان به‌صورت چندتایی $M = \langle S, A, P, f, C, T, \mu, \Delta s \rangle$ تعریف کرد:

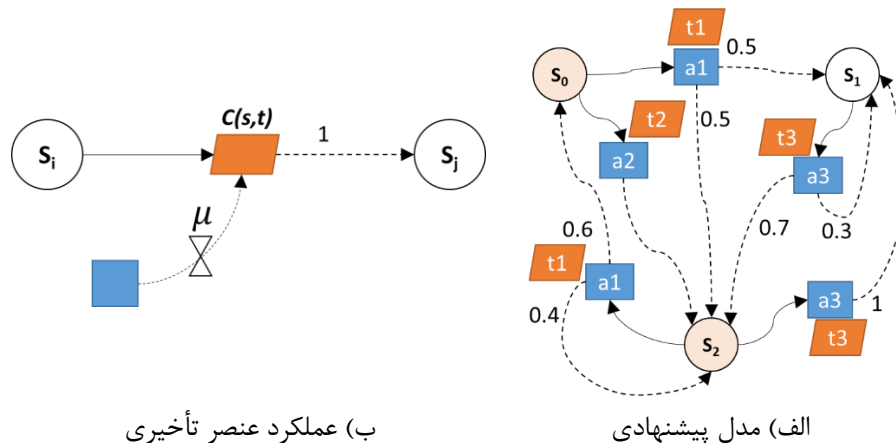
- S مجموعه‌ی حالت‌های یا وضعیت‌های سیستم است.
- A مجموعه‌ی کل عملیاتی است که روی S قابل تعریف است و به ازای هر وضعیت $s \in S$ ، مجموعه‌ی $A_s \subseteq A$ عملیات مجاز در وضعیت s است.

¹ eXtended Markov Decision Processes

- $P: S \times A \times S \rightarrow [0,1]$ تابع احتمال شرطی وضعیت بعدی سیستم مشروط به انتخاب یک عمل مشخص است به بیان دیگر $P(s'|s,a)$ احتمال رخ دادن حالت s' به عنوان حالت جدید سیستم است مشروط به اینکه در وضعیت s یک عمل $a \in A_s$ اتفاق افتاده باشد.
- T مجموعه‌ی Triggerهای تأخیری است که به ازای هر عمل انجام شده با تأخیر زمانی مشخص باعث تغییر وضعیت سیستم می‌شود.
- $f: S \rightarrow \mathbb{R}^+$ تابع برآزندگی وضعیت سیستم است.
- $C: S \times (A \cup T) \times S \rightarrow \mathbb{R}^+$ تابع هزینه (یا پاداش) است. به ازای هر عملی یا هر Trigger که در سیستم رخ می‌دهد وضعیت سیستم از حالت s به حالت s' تغییر می‌کند. اگر مقدار $c(s, t, s') = f(s) - f(s'), t \in T$ بزرگ‌تر از صفر باشد، مقدار $c(s, t, s')$ هزینه‌ی تحمیل شده به سیستم است و اگر این مقدار کمتر از صفر باشد، به عنوان پاداش به سیستم در نظر گرفته می‌شود. برای تغییر وضعیت‌هایی که یک عمل باعث آن شده است تابع هزینه می‌تواند مستقل از تابع برآزندگی حالت تعریف شود.
- $\mu: S \times A \times S \rightarrow \mathbb{R}^+$ تابع تولید زمان تأخیر رخ دادن Trigger، بعد از انجام عمل است.
- Δs متوسط فاصله‌ی زمانی بین رخ دادن دو عمل است.

مدل پیشنهادی نسبت به مدل MDPs که در بخش ۱-۱-۲ شرح داده شد؛ دارای عناصر جدیدی مانند مجموعه‌ی T و تابع μ است. وقتی Huang و همکاران [۱۰] الگوریتم تخصیص وظایف به منابع را بر اساس مدل MDPs طراحی کردند و آن را تحت عنوان روش RLRAM منتشر کردند. قرار دادن یک واحد کاری در محل مشخصی از لیست کاری منبع، به عنوان عمل (Action) در مدل MDPs تعریف کردند. از آنجایی که در اجرای فرآیندهای سازمانی، بارکاری منابع بخشی از وضعیت سیستم است و تغییرات بارکاری در زمان تخصیص واحد کاری به منبع و زمان تکمیل فعالیت مربوطه رخ می‌دهد. در نظر نگرفتن کاهش بارکاری در زمان تکمیل فعالیت، بر نتایج حاصل از اجرای فرآیند سازمانی منطبق بر مدل MDPs تأثیر نامطلوب می‌گذارد. برای حل این چالش به مدل پیشنهادی مجموعه‌ی T و تابع μ اضافه شده است، در نتیجه تطبیق بیش‌تری بین اجرای فرآیندهای سازمانی و مدل پیشنهادی وجود دارد. به نحوی که هر رویداد «تخصیص» واحد کاری به منبع با مجموعه‌ی عمل (action) منطبق می‌شود

و هر رویداد «تکمیل فعالیت» با مجموعه‌ی Triggerهای تأخیری منطبق می‌گردد و تابع μ بیانگر مجموع مدت زمان انتظار در صف و سرویس‌دهی به هر واحد کاری است. سایر بخش‌های مدل پیشنهادی با مدل MDPs تفاوت چندانی ندارد و همچنان تابع هزینه و وضعیت‌های مدل قابل تعریف است.



شکل (۱-۳) مدل پیشنهادی XMDPs و عملکرد عنصر trigger
(الف) مدل پیشنهادی (ب) عملکرد عنصر تأخیری

همان‌طور که در شکل (۱-۳) الف) نشان داده شده است در مدل پیشنهادی به ازای هر عمل یک Trigger وجود دارد وقتی در یک وضعیت مشخص یک action رخ می‌دهد وضعیت سیستم با احتمال مشخصی تغییر می‌کند و هزینه‌ی $c(s, a, s')$ به سیستم به صورت آنی تحمیل می‌شود. همان‌طور که در شکل (۱-۳) ب) آمده است، بعد از انتخاب یک عمل، Trigger مربوط به آن قطعاً با تأخیر μ رخ خواهد داد و وضعیت سیستم را از حالت جاری s_i به حالت s_j تغییر خواهد داد و هزینه‌ی $c(s, t, s')$ را با تأخیر به سیستم تحمیل خواهد کرد. مسئله‌ی فرآیندهای تصمیم‌گیری مارکوف به صورت مشابه برای مدل پیشنهادی قابل تعریف است و هدف، یافتن سیاستی جهت تصمیم‌گیری در انتخاب عمل مناسب در هر مرحله است به طوری که مجموعه‌ی هزینه‌های آنی و تأخیری، حداقل شود.

۳-۲ حل مسئله‌ی فرآیندهای تصمیم‌گیری مارکوف با مدل XMDPs

همان‌طور که در بخش قبل دیدیم با اضافه شدن عناصر جدید به مدل MDPs یک مدل پیشنهادی معرفی شد. مدل معرفی شده نسبت به مدل MDPs دارای پیچیدگی بیشتری است. در این مدل Triggerها در انجام هر عمل به وجود می‌آیند و با تأخیر زمانی مشخصی وضعیت سیستم را تغییر می‌دهد. به بیان بهتر زمانی که Trigger به عنوان یک تغییر دهنده حالت عمل می‌کند وضعیت جدید

سیستم وابسته به عملی (action) است که μ واحد زمانی قبل (نه صرفاً عمل قبلی) رخ داده است. به بیان دیگر وضعیت جدید سیستم به مجموعه‌ای از عملیات که در گذشته سیستم رخ داده است وابسته است. اگر به ازای هر وضعیت $s \in S$ و هر عمل $a \in A$ در مدل پیشنهادی یک مقدار Q-Value در نظر بگیریم برای حل مسئله‌ی XMDPs به استفاده از یادگیری تقویتی می‌توانیم از الگوریتم ذیل جهت به‌روزرسانی مقادیر Q-Value‌ها استفاده کنیم.

Q-Learning Algorithm for XMDPs

Input: n as integer, M as XMDPs = $\langle S, A, P, f, c, T, \mu, \Delta s \rangle$

```

main( $M, n$ ){
     $Triggers = \emptyset$ 
     $\forall v \in QValues, v = 0$ ;
    foreach( $s \in S$ )
         $QUpdate(s, 0, n)$ 
}

Function  $QUpdate(s, curTime, n)$  {
    if ( $n > 0$ ) {
        foreach( $\forall s' | P(s' | (s, a)) > 0$ ) {
             $t_a$  as  $Trigger = new()$ 
             $Triggers = Triggers \cup \{ \langle t_a, s, a, s', curTime \rangle \}$ 
             $QValue(s, a) = (1 - \alpha_n)QValue(s, a)$ 
             $+ \alpha_n (c(s, a, s') + \gamma P(s' | (s, a)) \times QUpdate(s', curTime + \Delta s, n - 1))$ 
        }
        foreach( $\langle t_a, s_a, a, s', actionTime \rangle \in Triggers$ )
            if( $curTime > actionTime + \mu(s_a, a, s')$ ){
                 $Triggers = Triggers - \{ \langle t_a, s_a, a, s', actionTime \rangle \}$ 
                 $QValue(s_a, a) = (1 - \beta_n)QValue(s_a, a)$ 
                 $+ \beta_n \left( f(s_a) - \sum_{s' \in S} P(s' | (s, t_a)) f(s') \right)$ 
            }
        }
    }
    return  $\min_{s' \in S, a \in A_s} (P(s' | (s, a)) \times Qvalue(s, a))$ 
}

```

شکل (۳-۲) الگوریتم یادگیری تقویتی برای مدل پیشنهادی XMDPs

الگوریتم فوق شامل دو بخش (۱) بدنه اصلی الگوریتم یادگیری و (۲) تابع بهروزرسانی Q-Value ها است. در بدنه اصلی مجموعه‌ی *Triggers* و ماتریس *QValues* با مقادیر تصادفی مقداردهی می‌شود و به ازای هر وضعیت $s \in S$ تابع بهروزرسانی به صورت $QUpdate(s, 0, n)$ فراخوانی می‌شود. تابع بهروزرسانی $QUpdate$ به صورت بازگشتی در n مرحله مقادیر Q-Value ها را بهروزرسانی می‌کند و در هر مرحله از پیشروی تابع $QUpdate$ مقدار n کاهش و زمان به میزان Δs افزایش می‌یابد. در هر مرحله مقدار Q-Value برای تمام Action خروجی از وضعیت s از روی مقدار بهروز شده‌ی Q-Value وضعیت s' بهروز می‌شود. به ازای هر Action نیز یک عنصر تأخیری t_a ایجاد می‌شود و در مجموعه‌ی *Triggers* قرار می‌گیرد. اثر عنصر تأخیری *Triggers* در تابع $QUpdate$ با تأخیر $\mu(s_a, a, s')$ واحد زمان با افزایش مقدار $curTime$ در اجراهای بعدی تابع بررسی می‌شود و در زمان مناسب مقدار $f(s) - \sum_{s' \in S} P(s'|s, t_a)f(s')$ را با ضریب β_n در مقدار $QValue(s_a, a)$ تأثیر خواهد گذاشت. ضرایب α_n و β_n نرخ یادگیری هستند که مقدار مثبت کم‌تر از یک دارند و به ترتیب میزان اثرپذیری Q-Value ها را از مقادیر Q-Value های وضعیت‌های بعدی و عناصر تأخیری را تنظیم می‌کنند.

۳-۳ استخراج مدل XMDPs از مدل فرآیند

استخراج مدل XMDPs از یک فرآیند سازمانی مستلزم استخراج کلیه‌ی حالات فرآیند که شامل وضعیت بارکاری و لیست کاری تمام منابع، وضعیت هر پرونده در فرآیند و مجموعه‌ی اطلاعات موجود در پرونده است که مسیر حرکت آن را مشخص می‌کند. تعداد حالات مدل XMDPs تابعی از نرخ ورود درخواست، نرخ سرویس‌دهی منابع، الگوریتم تخصیص وظایف، الگوهای تخصیص منابع، الگوهای داده و ساختار کنترلی فرآیند سازمانی است. در حالت کلی تعداد حالات فرآیند سازمانی نامحدود است. البته جهت استفاده از مدل XMDPs و حل آن با استفاده از یادگیری تقویتی لزومی به استخراج کل حالات فرآیند سازمانی نیست. مدل استخراج شده XMDPs مانند یک گراف با ساختاری شبیه درخت است

که ریشه‌ی این درخت حالت جاری فرآیند است و هر سطح از این درخت بیانگر حالات مختلف بعدی است که با الگوریتم تخصیص یا رویدادهای موجود در فرآیند رخ می‌دهد. بنابراین با توجه به ظرفیت حافظه و توان پردازشی موجود می‌توان به دفعات دلخواه، سطوح درخت را در استخراج مدل XMDPs افزایش داد و الگوریتم یادگیری تقویتی را روی آن اجرا کرد.

۳-۳-۱ استخراج دنباله‌ی حالات مدل فرآیند

جهت ایجاد مدل XMDPs بررسی تعداد دفعات انتقال بین هر دو حالت از فرآیند سازمانی نیز باید محاسبه شود که به کمک آن می‌توان مقادیر مؤلفه‌ی P را در مدل XMDPs تخمین زد. برای این منظور از یک الگوریتم تصادفی در تخصیص وظایف و یک شبیه‌ساز برای ایجاد مدل XMDPs و محاسبه‌ی مقادیر P استفاده می‌کنیم. اگر بخواهیم از الگوریتم تخصیص وظایف که بر پایه‌ی مدل XMDPs و یادگیری تقویتی طراحی شده است، استفاده کنیم. کافی است از وضعیت جاری سیستم یک دنباله‌ی منحصربه‌فرد ایجاد شود که بیانگر یک حالت از مدل XMDPs است. سپس با استفاده از یک الگوریتم تصادفی در تخصیص وظایف و یک الگوریتم تکاملی در یک شبیه‌ساز، به ازای هر با تکرار الگوریتم تکاملی در شبیه‌ساز دنباله‌ای از حالات ممکن از مدل XMDPs حاصل می‌شود. تکرارهای الگوریتم تکاملی تا جایی ادامه می‌یابد تا به مجموعه حالات یافته شده عضو جدیدی اضافه نشود. به عنوان مثال در جدول (۳-۱) یک اجرا از الگوریتم تکاملی در شبیه‌ساز را نشان می‌دهد. در این اجرا به ازای سه رویداد «تخصیص»، «تکمیل» و «ورود درخواست جدید»، حالت مدل فرآیند که شامل لیست کاری سه منبع و تعداد توکن‌ها در ۴ مکان غیر پایانی است را نشان می‌دهد. بنابراین برای هر حالت یک عبارت مانند $A1B2C1-1,1-4,1$ به عنوان شناسه حالت تعریف می‌شود که بدین معناست، برای منابع A ، B و C به ترتیب ۱، ۲ و ۱ توکن در لیست کاری آن‌ها وجود دارد و در مکان ۱ و ۴ در مدل فرآیند نیز یک توکن در انتظار تخصیص است. در این اجرا، ۱۰ رویداد رخ می‌دهد و دنباله‌ی حالات مدل فرآیند به دست می‌آید. به ازای هر اجرا در شبیه‌ساز یک دنباله‌ی جدید از حالات مدل فرآیند ایجاد می‌شود. تعداد اجراهای شبیه‌ساز آن قدر ادامه می‌یابد تا در دنباله‌ی حالات ایجاد شده حالت جدیدی مشاهده نشود.

جدول (۱-۳) یک مثال از استخراج دنباله‌ی حالات مدل فرآیند

ردیف	حالت و شناسه‌ی حالت	ردیف	حالت و شناسه‌ی حالت
حالت جاری	شناسه‌ی حالت: A1C1-1,1-2,1-4,1	۱- تخصیص توکن مشکی به منبع C	A1C2-1,1-4,1
۲- تخصیص توکن زرد به منبع B	A1B1C2-1,1	۳- تکمیل فعالیت مربوط به توکن سفید	A1B1C1-1,1-4,1
۴- تخصیص توکن سبز به منبع A	A2B1C1-4,1	۵- تکمیل فعالیت مربوط به توکن قرمز	A1B1C1-4,1
۶- تکمیل فعالیت مربوط به توکن زرد	A1C1-4,1	۷- ورود توکن جدید به فرآیند با رنگ قرمز	A1C1-1,1-4,1
۸- تکمیل فعالیت مربوط به توکن سبز	C1-1,1-2,1-4,1	۹- تخصیص توکن قرمز به منبع A	A1C1-2,1-4,1
۱۰- ورود توکن جدید به فرآیند با رنگ زرد	A1C1-1,1-2,1-4,1	وضعیت مدل فرآیند بعد از رویداد دهم مانند حالت جاری مدل فرآیند شده است. این مثال نشان می‌دهد دنباله‌ی حالات ایجاد شده می‌توان دارای حلقه نیز باشد.	

۲-۳-۳ استخراج مدل XMDPs از مجموعه‌ی دنباله‌ی حالات

در بخش ۱-۳-۳ روش ایجاد مجموعه‌ی دنباله‌ی حالات فرآیند به کمک یک الگوریتم تکاملی در اجراهای مختلف شبیه‌ساز با یک الگوریتم تصادفی در تخصیص وظایف شرح داده شد. در این بخش روش ایجاد مدل XMDPs را از مجموعه‌ی دنباله‌ی حالات به دست آمده شرح می‌دهیم.

اگر مجموعه‌ی $L = \{t_1, t_2, t_3, \dots, t_N\}$ دنباله‌های ایجاد شده از اجرای شبیه‌ساز توسط الگوریتم تکاملی باشد و به ازای هر $t_i \in L$ دنباله‌ی $t_i = \langle s_1, e_1, s_2, e_2, s_3, e_3, \dots, s_M \rangle$ بیانگر حالات گذار دیده شده و رویدادهای رخ داده از اجرای i ام شبیه‌ساز باشد با بررسی جفت حالات متوالی می‌توان مدل XMDPs را ایجاد کرد. در دنباله‌ی حالات گذار دو عنصر s و e وجود دارد هر عنصر s شناسه‌ی منحصر به فرد یک حالت در مدل فرآیند است و هر عنصر e یک رویداد در اجرای فرآیند سازمانی است که باعث تغییر حالت فرآیند می‌شود. $e.Type$ نوع رویداد را مشخص می‌کند. در استخراج مدل XMDPs نوع این رویدادها یکی از سه مقدار ورود توکن (ورود درخواست)، تخصیص یا تکمیل را خواهد گرفت که به ترتیب با عبارات New ، $Allocate$ و $Complete$ مقداردهی می‌شود. $e.Time$ زمان رخ دادن رویداد را مشخص می‌کند و $e.Token$ شناسه‌ی توکنی است که رویداد روی آن رخ داده است. با توجه به مفروضات فوق روش محاسبه‌ی هر یک از مؤلفه‌های مدل $M = \langle S, A, P, f, C, T, \mu, \Delta S \rangle$ به صورت ذیل خواهد بود.

- مجموعه‌ی حالات مدل شامل مجموعه‌ی کلیه‌ی حالات مشاهده شده در تمام دنباله‌های حالات گذار است به عبارت دیگر $S = \{s | \forall s \in t, \forall t \in L\}$
- مجموعه‌ی A برابر است با مجموعه‌ی کلیه‌ی رویدادهای تخصیص دیده شده در تمام دنباله‌های حالات گذار است به عبارت دیگر

$$A = \{e | \forall e \in t, \forall t \in L, e.Type = Allocate\} \quad (۱-۳)$$

- مجموعه‌ی T برابر است با مجموعه‌ی کلیه‌ی رویدادهای تخصیص دیده شده در تمام دنباله‌های حالات گذار است به عبارت دیگر

$$T = \{e | \forall e \in t, \forall t \in L, e.Type = Complete\} \quad (۲-۳)$$

- با محاسبه‌ی میانگین زمان بین دو رویداد ورود می‌توان مقدار Δs را تخمین زد در نتیجه

$$\begin{aligned}\Delta s &= avg(e_i.time - e_j.time), \forall e_i, e_j, e_k \in t, \forall t \in L, e_i.type \\ &= e_j.type = e_k.type = Allocate, e_i.time > e_j.time \\ &, \nexists e_k | e_i.time > e_k.time > e_j.time\end{aligned}\quad (3-3)$$

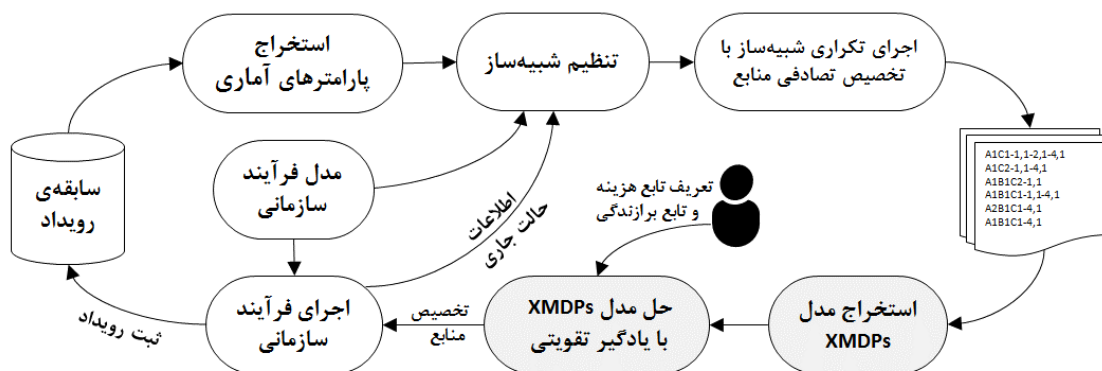
- برای محاسبه‌ی مقادیر مؤلفه‌ی μ کافی است در دنباله‌ی حالات، توالی‌های چهارتایی $\langle s_i, e_i, s_{i+1}, e_j \rangle$ به ازای تمام $t \in L$ به‌طوری‌که $e_i.token = e_j.token$ و $e_i.type = Allocate$ و $e_j.type = complete$ باشد. در این حالت مقدار هر مؤلفه‌ی μ به‌صورت $\mu(s_i, e_i, s_{i+1}) = avg(e_j.time - e_i.time)$ محاسبه می‌شود.
- برای محاسبه‌ی مؤلفه‌ی P نیز از رابطه‌ی ذیل استفاده می‌کنیم.

$$P(s_{i+1} | (s_i, e_i)) = \frac{|\{(s_i, e_i, s_{i+1}) | \langle s_i, e_i, s_{i+1} \rangle \in subSeq(t), \forall t \in L\}|}{|B\{(s_i, e_i) | \langle s_i, e_i \rangle \in subSeq(t), \forall t \in L\}|}\quad (4-3)$$

۳-۴ آزمایش مدل پیشنهادی XMDPs

همان‌طور که در بخش ۳-۳ شرح داده شد برای ایجاد مدل XMDPs لازم است ابتدا مجموعه‌های S و A که به ترتیب وضعیت‌های گذار و عملیات مجاز در هر وضعیت است، مشخص شود. این اطلاعات هم با بررسی و تحلیل فرآیند سازمانی حاصل می‌شود و هم با شمارش تعداد حالات دیده شده در دنباله‌ی حالات گذار میسر است. به‌طور عام، فرآیند سازمانی با یکی از روش‌های مدل‌سازی فرآیند، مانند WF-net مدل می‌شود. از آنجایی که WF-net یک نوع توسعه‌یافته‌ی Perti-net است. استخراج حالات گذار آن با استفاده از تکنیک‌های شبیه‌سازی و بررسی سابقه‌ی رویدادهای ایجاد شده، امکان‌پذیر است. برای ایجاد مدل XMDPs از روی سابقه‌ی رویدادهای اجرای فرآیند سازمانی لازم است. اطلاعات درخواست‌های ورودی و Trace‌ها در یک شبیه‌ساز فرآیند سازمانی باز اجرا^۱ (اجرای مجدد درخواست مطابق با اجرای واقعی) شود و به هر وضعیت سیستم یک شناسه‌ی منحصر به فرد نسبت داده شود و دنباله‌ای از حالات گذار فرآیند در باز اجرای فرآیند ایجاد شود. با شمارش چندتایی‌های متوالی می‌توان مؤلفه‌های مدل XMDPs را به دست آورد.

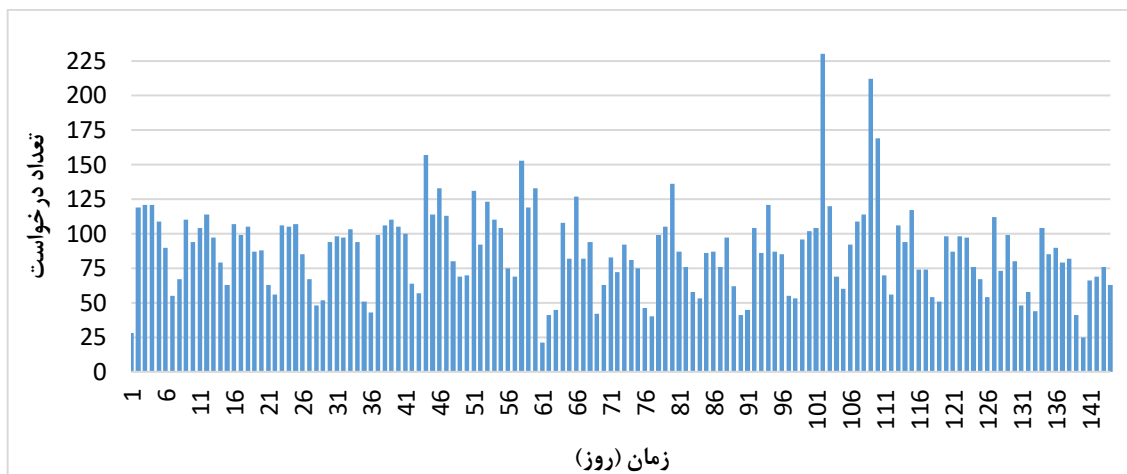
^۱ Replay



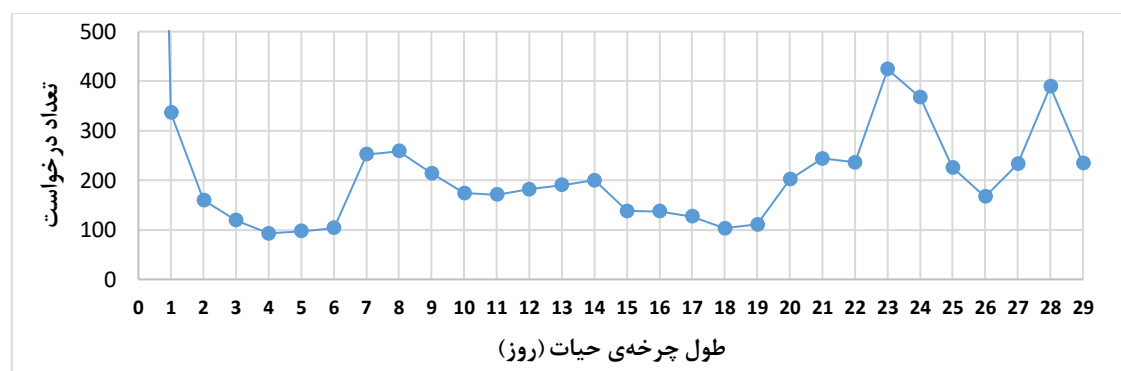
شکل (۳-۳) روش استفاده از مدل پیشنهادی در تخصیص منابع

تابع هزینه‌ی هر Action و برازندگی هر حالت بر اساس سیاست سازمان تعریف و تنظیم می‌گردد. با مشخص شدن تمام مؤلفه‌های مدل XMDPs، امکان حل آن با استفاده از یادگیری تقویتی امکان‌پذیر است. بعد از حل مدل XMDPs، یک عامل تصمیم‌گیری به وجود می‌آید که امکان انتخاب منبع مناسب جهت تخصیص واحد کاری با آن میسر می‌شود به‌طوری‌که این انتخاب در جهت برآورده کردن سیاست سازمان است. در شکل (۳-۳) روش استفاده از مدل پیشنهادی XMDPs آمده است. قبل از تصمیم‌گیری جهت تخصیص یک منبع به یک واحد کاری با بررسی اطلاعات ثبت شده در سابقه‌ی رویداد و اطلاعات مدل فرآیند، یک شبیه‌ساز فرآیند سازمانی تعریف و تنظیم می‌شود و با شروع از حالت جاری فرآیند، این شبیه‌ساز چندین بار اجرا می‌شود تا مجموعه‌ی دنباله‌ی حالات گذار فرآیند به وجود بیاید. سپس از روی این مجموعه، مدل XMDPs استخراج می‌شود و با الگوریتم یادگیری تقویتی ارائه شده در این فصل مدل XMDPs حل می‌شود و از مدل حل شده جهت تخصیص مناسب منبع استفاده می‌شود.

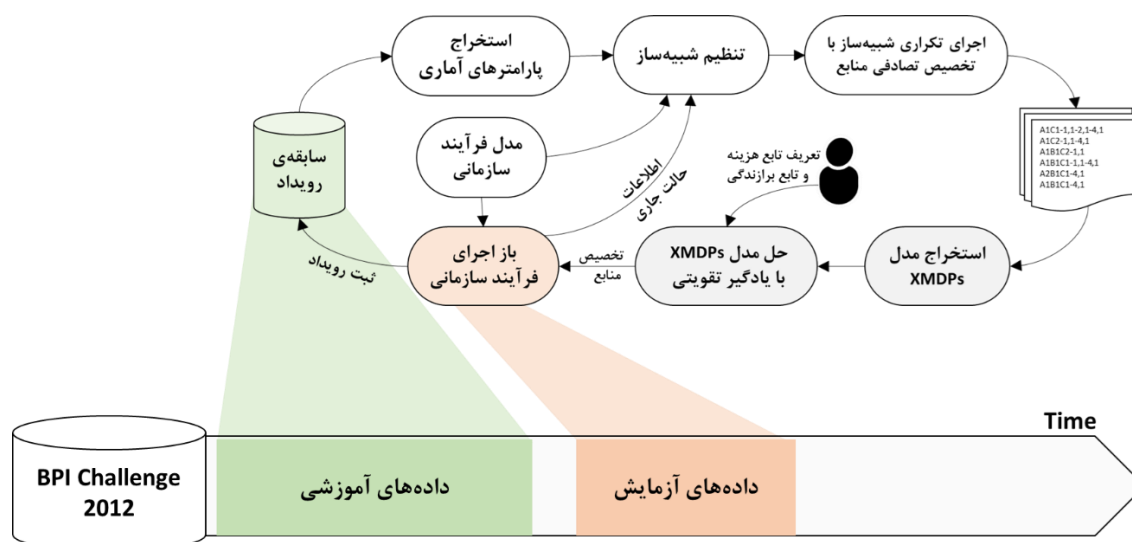
برای آزمایش مدل پیشنهادی از پایگاه داده‌ی BPI Challenge 2012 استفاده می‌کنیم. این پایگاه داده توسط یک موسسه‌ی مالی در کشور هلند ایجاد شده است که شامل ۵ وظیفه، ۵۵ منبع و ۱۳۰۸۷ درخواست است که در مجموع شامل ۲۶۲۲۰ رویداد ذخیره شده است. این رویدادها در مجموع در شش ماه (سه ماه آخر ۲۰۱۱ و سه ماه اول ۲۰۱۲) جمع‌آوری شده است (شکل (۳-۴)). طول چرخه‌ی زمانی درخواستها در این پایگاه داده از ۳۰ روز تجاوز نکرده است (شکل (۳-۵)) و متوسط طول چرخه‌ی زمانی درخواستها حدود ۱۰ روز است.



شکل (۳-۴) تعداد درخواست‌های ورودی به تفکیک روز در پایگاه داده‌ی BPI Challenge 2012



شکل (۳-۵) تعداد درخواست‌ها به تفکیک طول چرخه‌ی زمانی در پایگاه داده‌ی BPI Challenge 2012



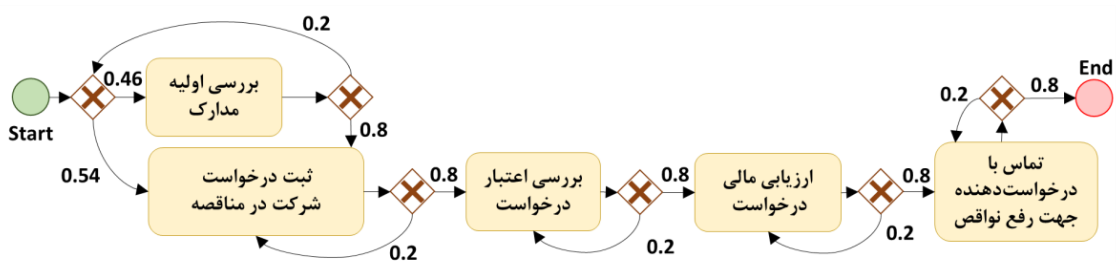
شکل (۳-۶) داده‌های آموزش و آزمایش مدل پیشنهادی XMDPs

در آزمایش مدل پیشنهادی، داده‌های پایگاه داده‌ی BPI Challenge 2012 را بر اساس جدول (۳-۲) به ۵ قسمت تقسیم می‌کنیم در هر قسمت، از اطلاعات ۳ ماه از پایگاه داده استفاده می‌شود. از ۲ ماه برای آموزش (به جز تکرار اول) و از یک ماه برای آزمایش مدل استفاده می‌شود.

جدول (۲-۳) تقسیم‌بندی پایگاه داده‌ی BPI Challenge 2012 جهت آزمایش مدل پیشنهادی

تکرار آزمایش	داده‌های آموزشی			داده‌های آزمایش		
	ماه	سال	تعداد درخواست	ماه	سال	تعداد درخواست
۱	۱۰	۲۰۱۱	۲۳۸۵	۱۱	۲۰۱۱	۲۹۴۳
۲	۱۰	۲۰۱۱	۲۳۸۵	۱۲	۲۰۱۱	۲۱۲۶
	۱۱	۲۰۱۱	۲۹۴۳			
۳	۱۱	۲۰۱۱	۲۹۴۳	۱	۲۰۱۲	۲۷۳۸
	۱۲	۲۰۱۱	۲۱۲۶			
۴	۱۲	۲۰۱۱	۲۱۲۶	۲	۲۰۱۲	۲۸۷۴
	۱	۲۰۱۲	۲۷۳۸			
۵	۱	۲۰۱۲	۲۷۳۸	۳	۲۰۱۲	۲۱
	۲	۲۰۱۲	۲۸۷۴			

به ازای هر بار تکرار آزمایش، با تحلیل و آنالیز آماری سابقه‌ی رویدادهای ثبت شده در ۲ ماه داده‌های آموزشی (جدول (۲-۳)) و مدل فرآیند (شکل (۷-۳)) و توابع برازندگی و هزینه که به ترتیب در رابطه‌های (۵-۳) و (۶-۳) آمده است، مدل XMDPs حاصل می‌شود. سپس بر اساس الگوریتم یادگیری تقویتی ارائه شده در این فصل، مدل حل می‌شود و برای آزمایش مدل حل شده از داده‌های آزمایشی استفاده می‌کنیم. برای تثبیت نتایج، آزمایش بر اساس جدول (۲-۳) پنج بار تکرار می‌شود و برای مقایسه‌ی نتایج از روش پیشنهادی Huang و همکاران [۱۰] تحت عنوان RL RAM نیز استفاده می‌کنیم.



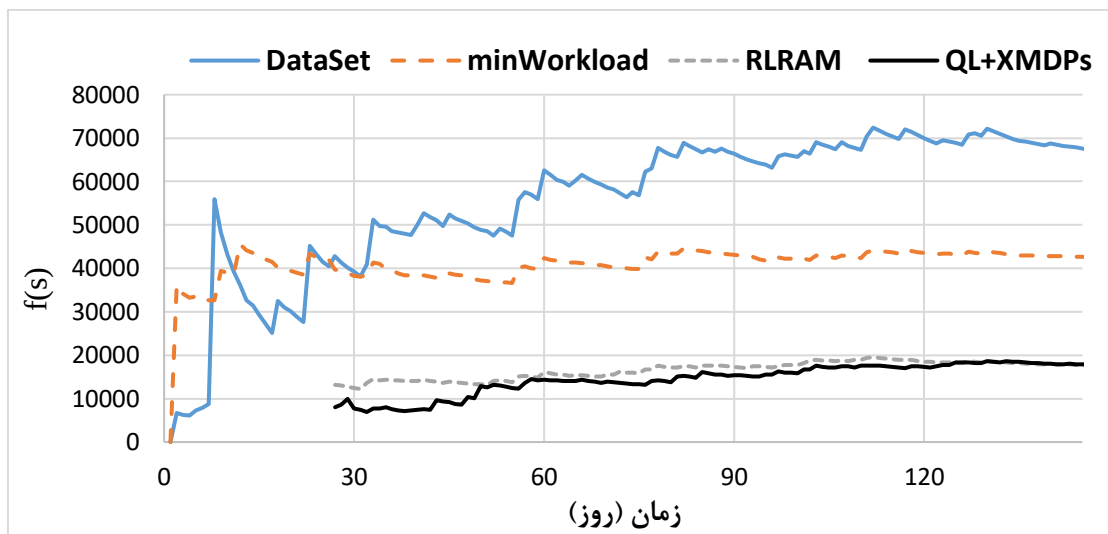
شکل (۷-۳) فرآیند موسسه‌ی مالی هلند [۲۷]

$$f(s) = \min \left\{ \alpha \left(\frac{\sum_{r \in R} \sum_{i=0}^N (ld_r(t_i) - EL(t_i))^2}{N|R|} \right) + \right. \\ \left. (1 - \alpha) \left(\frac{\sum_{r \in R} \sum_{i=0}^N (ld_r(t_i) - \mu_r)^2}{N|R|} \right) \right\} \quad (۵-۳)$$

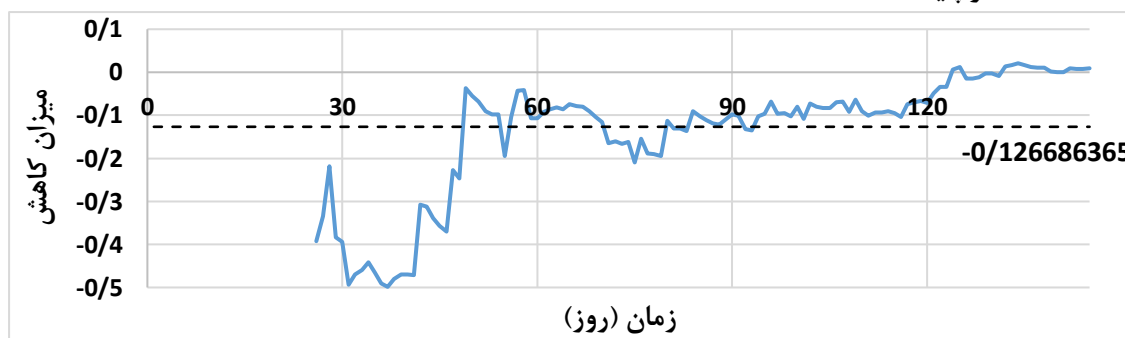
$$\alpha = 0.5(\text{default value})$$

$$c(s, a, s') = f(s') - f(s) \quad (۶-۳)$$

همان‌طور که در بخش‌های ۷-۲-۱ و ۸-۲-۱ روش محاسبه‌ی تعادل بارکاری و یکنواختی بارکاری منابع شرح داده شد. برای برقرار تعادل بارکاری و ایجاد یکنواختی بارکاری لازم است مقدار رابطه‌ی (۵-۳) در هر وضعیت $s \in S$ در مدل XMDPs کمینه شود بنابراین تابع برازندگی هر وضعیت را مطابق با رابطه‌ی (۵-۳) تعریف می‌کنیم و تابع هزینه انجام هر $a \in (A \cup T)$ مطابق با رابطه‌ی (۶-۳) تعریف می‌شود. در شکل (۸-۳) نتایج اجرای الگوریتم تخصیص منابع، بر اساس الگوریتم یادگیری تقویتی ارائه شده در این فصل آمده است. جهت مقایسه‌ی الگوریتم پیشنهادی، آزمایش مربوطه بر اساس داده‌های فهرست شده در جدول (۲-۳) با الگوریتم RL RAM و تابع برازندگی رابطه‌ی (۳-۱) انجام می‌شود. تنها تفاوت روش پیشنهادی با الگوریتم RL RAM در مدل‌سازی فضای مسئله است. در روش پیشنهادی از مدل پیشنهادی XMDPs استفاده شده است درحالی‌که در الگوریتم RL RAM از مدل MDPs استفاده می‌شود. جهت مشاهده‌ی تأثیر یادگیری تقویتی در روش پیشنهادی و الگوریتم RL RAM یک بار آزمایش با الگوریتم تخصیص منابع MinWorkload انجام شده است. در این روش از تکنیک یادگیری تقویتی استفاده نمی‌شود و تخصیص بر اساس بارکاری منابع در حالت جاری سیستم انجام می‌شود و واحد کاری به منبع با کم‌ترین بارکاری تخصیص داده می‌شود. در شکل (۸-۳) علاوه بر الگوریتم‌های ذکر شده مقدار تابع برازندگی بر اساس وضعیت بارکاری ثبت شده در پایگاه داده‌ی BPI Challenge 2012 تحت عنوان DataSet نیز نشان داده شده است که میزان کاهش تابع برازندگی را نسبت به وضعیت فعلی پایگاه داده نشان می‌دهد. فاصله الگوریتم MinWorkload با الگوریتم پیشنهادی نشان می‌دهد نرخ درخواست‌ها نسبت به تعداد منابع کم بوده است و فاصله الگوریتم MinWorkload با DataSet بیانگر تخصیص نامناسب انجام شده در پایگاه داده است. هرچه نرخ ورودی درخواست‌ها در این پایگاه داده افزایش یابد فاصله‌ی الگوریتم MinWorkload از الگوریتم‌هایی که بر اساس یادگیری تقویتی تخصیص منابع را انجام داده‌اند، کاهش می‌یابد و این جمله بیانگر تأثیر مطلوب تکنیک یادگیری تقویتی در شرایط پایدار سیستم با نرخ ورود کم است.



شکل (۳-۸) نتایج مقایسه‌ی الگوریتم یادگیری تقویتی بر اساس مدل‌های (RLRAM)MDPs و XMDPs با الگوریتم پایه‌ی تخصیص بارکاری به منبع با کم‌ترین بارکاری (MinWorkload) در بهینه‌سازی برقراری تعادل بارکاری و یکنواختی بارکاری منابع روی پایگاه داده‌ی BPI Challenge 2012 در مقایسه با الگوریتم تخصیص استفاده شده در پایگاه داده (DataSet)



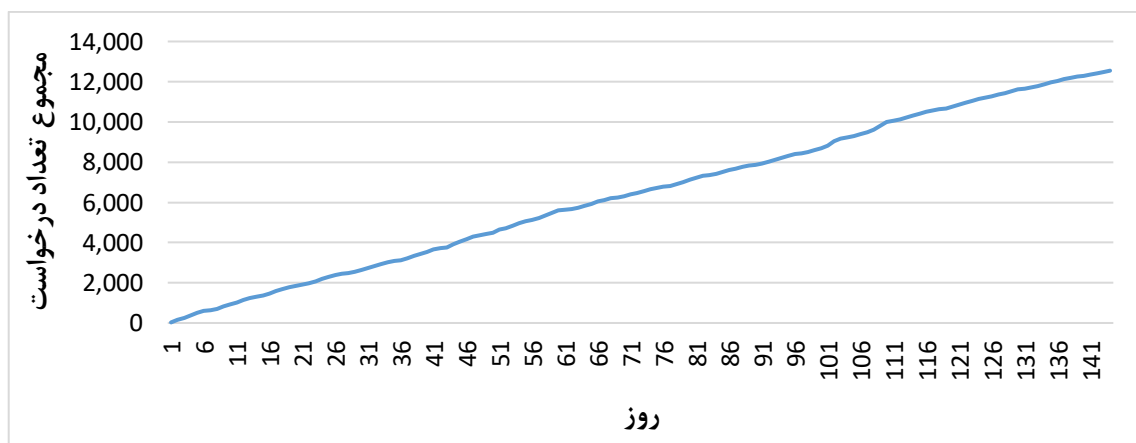
شکل (۳-۹) میزان کاهش $f(s)$ در الگوریتم QL+XMDPs نسبت به الگوریتم RLRAM

در شکل (۳-۹) میزان کاهش تابع برازندگی (۳-۵) در الگوریتم QL+XMDPs را نسبت به الگوریتم RLRAM نشان می‌دهد. از آنجایی که تنها تفاوت روش QL+XMDPs با RLRAM در استفاده از مدل MDPs است. می‌توان نتیجه گرفت که مدل پیشنهادی XMDPs مدل مناسب‌تری برای مسئله‌ی تخصیص منابع در اجرای فرآیندهای سازمانی است. در شکل (۳-۹) میانگین میزان کاهش نیز محاسبه شده است که مقدار ۱۲/۷ درصدی را نشان می‌دهد که نشان دهنده‌ی میزان بهبود مدل پیشنهادی XMDPs نسبت به مدل MDPs در حل مسئله‌ی یکنواختی و تعادل بارکاری منابع است. در بازه‌ی ۳۰ الی ۶۰ روز در شکل (۳-۹) میزان کاهش بیش‌تری را نشان می‌دهد که ممکن است به داده‌های آموزشی تکرار اول آزمایش مربوط باشد. زیرا در تکرار اول از داده‌های آموزشی یک ماه گذشته استفاده شده و

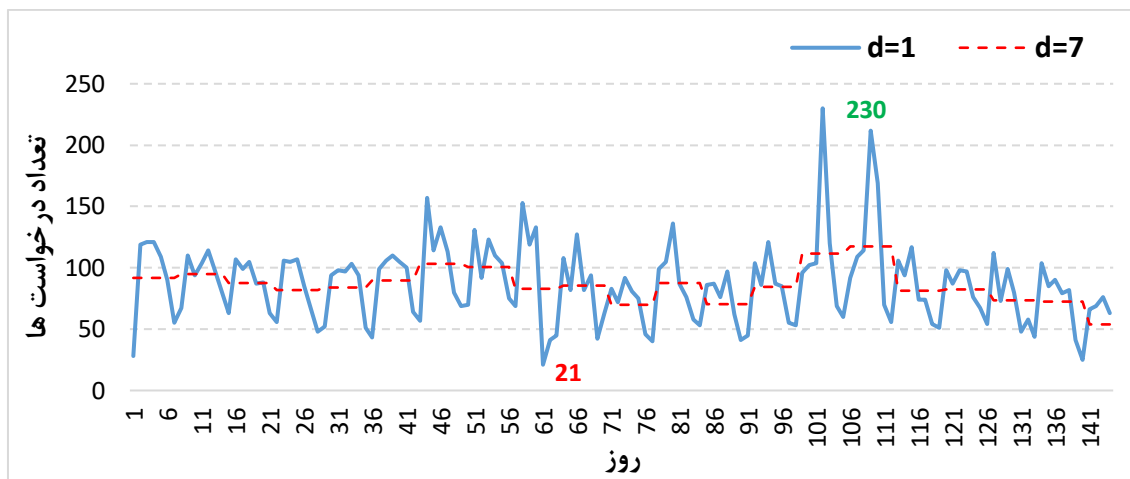
در تکرارهای بعدی از داده‌های آموزشی دو ماه گذشته استفاده شده است. بنابراین می‌توان نتیجه گرفت در استفاده از سابقه‌ی رویداد باید از داده‌های گذشته‌ی نزدیک‌تر استفاده کرد. استفاده از سابقه‌ی رویدادهای قدیمی‌تر می‌تواند اثر نامطلوب در الگوریتم یادگیری داشته باشد. استفاده از بازه‌ی مناسب از گذشته‌ی سیستم نیز می‌تواند به عنوان یک مسئله‌ی کاربردی در راستای تحقیقات انجام شده در این رساله می‌تواند مطرح شود.

۳-۴-۱ مشاهده‌ی اثر افزایش بازه‌ی یکنواختی در بهبود یادگیری

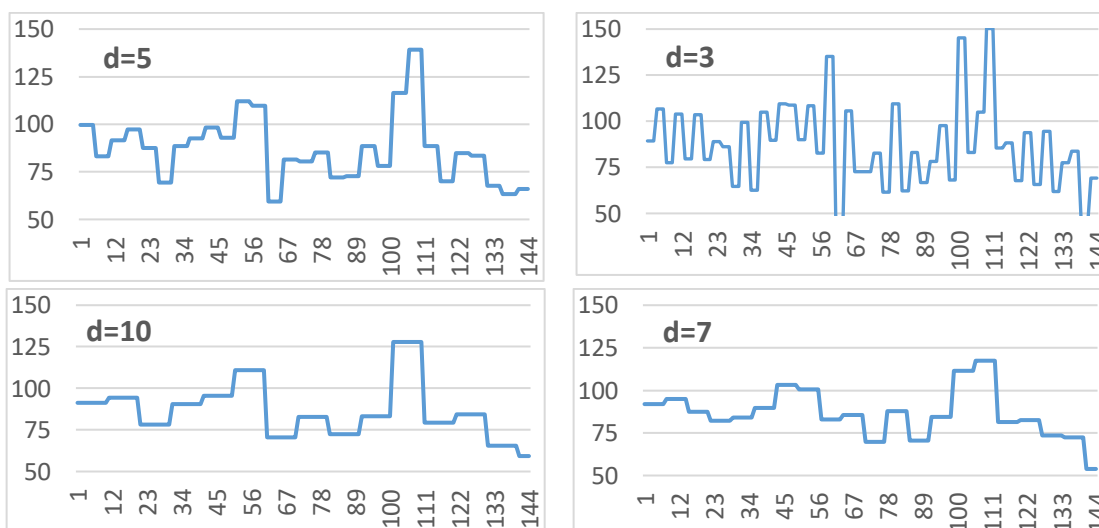
یکی از مهم‌ترین چالش‌هایی که در ایجاد یکنواختی بارکاری منابع وجود دارد غیر یکنواخت بودن آهنگ ورود درخواست‌ها به فرآیند است. وجود نوسان در تعداد درخواست‌های ورودی در هر روز می‌تواند باعث ایجاد نوسان در بارکاری منابع شود. بررسی پایگاه داده‌ی BPI Challenge 2012 نشان داد تغییرات تعداد درخواست‌ها در کل دوره‌ای که پایگاه داده ایجاد شده است به صورت یکنواخت بوده است. این موضوع در شکل (۳-۱۰) که تعداد تجمعی روزانه‌ی درخواست‌ها را نشان می‌دهد و به صورت خطی افزایش یافته است، مشهود است ولی تعداد درخواست‌ها به صورت روزانه نوسان زیادی (بین ۲۱ تا ۲۳۰ درخواست) دارد. شکل (۳-۱۱) تغییرات تعداد درخواست‌ها را به صورت روزانه با برچسب $d=1$ در پایگاه داده‌ی BPI Challenge 2012 را نشان می‌دهد.



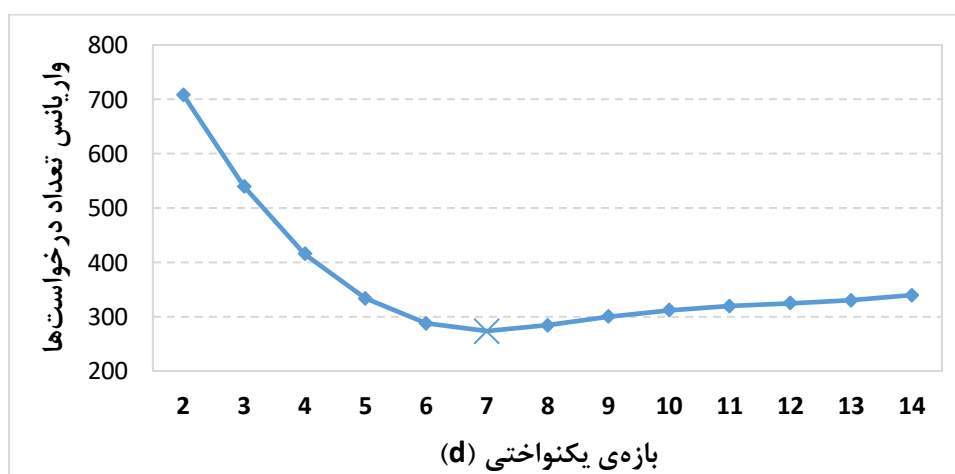
شکل (۳-۱۰) مجموع روزانه‌ی تعداد درخواست‌ها از ابتدای دوره در پایگاه داده‌ی BPI Challenge 2012



شکل (۳-۱۱) تعداد درخواست‌ها به تفکیک روز در پایگاه داده‌ی BPI Challenge 2012



(الف) نمودار تعداد درخواست‌ها با بازه‌های یکنواختی متفاوت



(ب) محاسبه‌ی واریانس تعداد درخواست‌ها برای بازه‌های یکنواختی بین ۲ الی ۱۴

شکل (۳-۱۲) واریانس تعداد درخواست‌ها برای بازه‌های یکنواختی متفاوت

اگر بخواهیم اثر نوسان زیاد تعداد درخواست‌ها را در یکنواختی بارکاری منابع کاهش دهیم می‌توانیم تعداد درخواست‌ها را در بازه‌های زمانی بزرگ‌تری محاسبه کنیم و تابع برازندگی را برای یکنواختی بارکاری منابع برای بازه‌های بزرگ‌تر تعریف کنیم. تعریف بازه‌ی یکنواختی بزرگ‌تر به معنای یکنواختی بارکاری با درجه‌ی سختی کم‌تر است. به عنوان مثال به‌جای ایجاد یکنواختی در بارکاری منابع در هر روز می‌توان یکنواختی بارکاری منابع را در هفته در نظر گرفت. این موضوع باعث می‌شود الگوریتم یادگیری تقویتی آزادی بیش‌تری در انتخاب منبع مناسب داشته باشد و ممکن است اثر یکنواختی بهتری در بارکاری منابع داشته باشد. برای انتخاب یک بازه‌ی مناسب، واریانس تعداد درخواست‌ها را برای بازه‌های یکنواختی ۲ الی ۱۴ روزه روی پایگاه داده مذکور محاسبه کردیم. شکل (۳-۱۲الف) نمودار تعداد درخواست‌ها برای بازه‌های یکنواختی ۳، ۵، ۷ و ۱۰ را نشان می‌دهد برای هر یک از این نمودارها واریانس تغییرات تعداد درخواست‌ها را محاسبه می‌کنیم، همان‌طور که در شکل (۳-۱۲ب) مشخص است واریانس تعداد درخواست‌ها برای بازه‌ی یکنواختی ۷ روزه کم‌ترین مقدار را دارد، بنابراین الگوریتم QL+XMDPs را با تابع برازندگی $g(s)$ که در رابطه (۳-۹) آمده است مورد آزمایش مجدد قرار دادیم. این تابع مشابه تابع برازندگی $f(s)$ در رابطه‌ی (۳-۵) است با این تفاوت که پارامتر جدید d به عنوان بازه‌ی یکنواختی به آن اضافه شده است. در این آزمایش یک‌بار پارامتر d را با مقدار ۷ و یک‌بار با مقدار یک تنظیم می‌شود.

$$ld_r^d(t_j) = \frac{\sum_{i=jd}^{(j+1)d-1} ld_r(t_i)}{d} \quad (۷-۳)$$

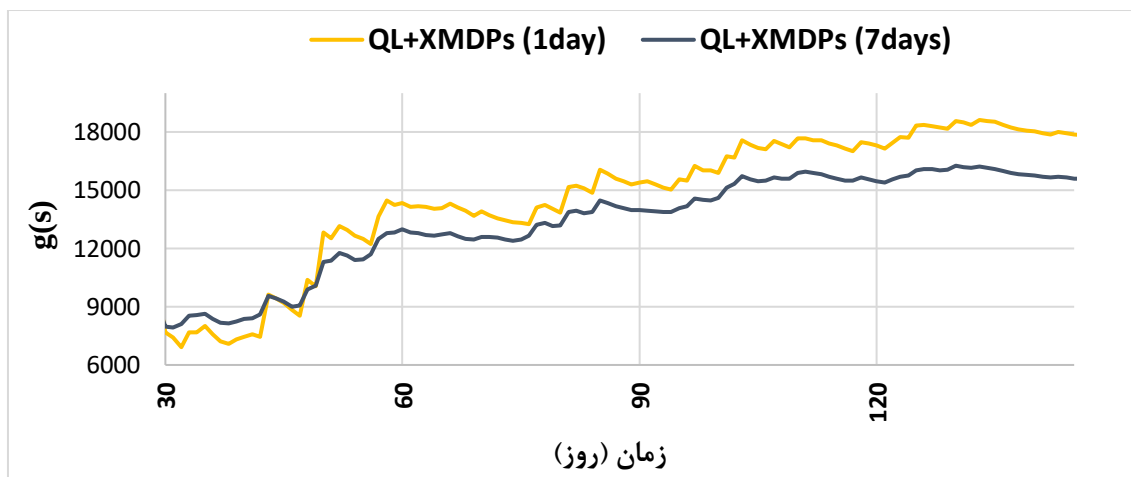
$$EL^d(t) = \sum_{r \in R} ld_r^d(t) / |R| \quad (۸-۳)$$

$$g(s) = \min \left\{ \alpha \left(\frac{\sum_{r \in R} \sum_{j=0}^{\lfloor N/d \rfloor} (ld_r^d(t_j) - EL^d(t_j))^2}{[N/d] |R|} \right) + \right. \quad (۹-۳)$$

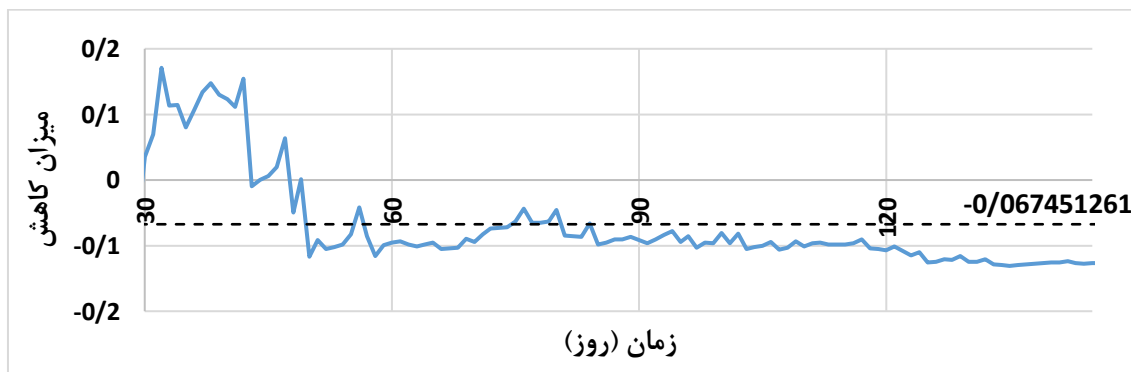
$$\left. (1 - \alpha) \left(\frac{\sum_{r \in R} \sum_{j=0}^{\lfloor N/d \rfloor} (ld_r^d(t_j) - \mu_r)^2}{[N/d] |R|} \right) \right\}$$

$$c(s, a, s') = g(s') - g(s) \quad (۱۰-۳)$$

نتیجه‌ی اجرای مجدد الگوریتم با داده‌های آموزشی و آزمایشی جدول (۳-۲) و مقادیر متفاوت پارامتر d نشان می‌دهد. افزایش بازه‌ی محاسبه‌ی بارکاری به طوری که واریانس تعداد درخواست‌ها کمینه باشد می‌تواند اثر مطلوب بر تعادل و یکنواختی بارکاری منابع داشته باشد. شکل (۳-۱۳الف) مقایسه‌ی اجرای الگوریتم QL+XMDPs را برای دو مقدار ۷ و ۱ برای پارامتر d در تابع برازندگی رابطه‌ی (۳-۹) را نشان می‌دهد. همان‌طور که در شکل (۳-۱۳ب) مشهود است الگوریتم پیشنهادی با پارامتر $d=7$ (QL+XMDPs 7days) ۶/۷ درصد بهتر نسبت به الگوریتم QL+XMDPs با پارامتر $d=1$ توانسته است تعادل و یکنواختی در بارکاری منابع ایجاد کند.



(الف) مقایسه‌ی بازه‌ی یکنواختی در الگوریتم QL+XMDPs



(ب) میزان کاهش $g(s)$ در الگوریتم QL+XMDPs (7days) نسبت به QL+XMDPs (1days)

شکل (۳-۱۳) تغییر در تابع محاسبه‌ی بارکاری و اثر آن در یکنواختی و تعادل بارکاری منابع

۳-۵ جمع‌بندی و نتیجه‌گیری

در این فصل مدل فرآیندهای تصمیم‌گیری مارکوف (MDPs) توسعه داده شد و مدل فرآیندهای تصمیم‌گیری توسعه‌یافته‌ی مارکوف (XMDPs) بر پایه‌ی یادگیری تقویتی ارائه گردید. مدل پیشنهادی XMDPs انطباق بیشتری با مسئله‌ی تخصیص منابع در اجرای فرآیندهای سازمانی دارد. تفاوت اصلی مدل پیشنهادی XMDPs با مدل MDPs در وجود عناصر تأخیری Trigger در مدل توسعه‌یافته است این عنصر باعث می‌شود بعد از رخ دادن یک عمل (Action) در مدل با تأخیری مشخص (قابل محاسبه به‌صورت آماری) یک Trigger رخ دهد و وضعیت سیستم را تغییر دهد با اضافه شدن این عنصر در مدل پیشنهادی لازم شد در الگوریتم یادگیری تقویتی یک پارامتر زمان نیز اضافه شود. برای مقایسه‌ی مدل پیشنهادی با مدل MDPs، روش RLRAM یک‌بار بر پایه‌ی مدل پیشنهادی و یک‌بار بر پایه‌ی مدل MDPs روی پایگاه داده‌ی BPI Challenge 2012 مورد آزمایش قرار گرفتند و نتایج نشان داد مدل پیشنهادی ۱۲/۷ درصد بهتر از مدل MDPs در برقراری تعادل بارکاری و یکنواختی منابع مؤثر است. در انتهای این فصل پارامتر جدیدی تحت عنوان بازه یکنواختی درخواست ورودی به تابع برازندگی اضافه شد و تنظیم این پارامتر بر اساس مقدار کمینه‌ی واریانس تعداد درخواست‌ها، می‌تواند اثر مطلوب ۶/۷ درصدی در بهبود عملکرد الگوریتم داشته باشد.

فصل ۴ – کاهش چرخه‌ی زمانی در تخصیص متعادل بارِ کاری

معرفی مفهوم شباهت وظایف و فاصله‌ی وابستگی وظایف

ارائه‌ی الگوریتم تکمیلی CPDT

ارائه‌ی الگوریتم تکمیلی CPDT2

در تحقیقی که توسط Liu و همکاران انجام شد [۲۷]، به نوعی ارتباط اجتماعی^۱ بین منابع انسانی اشاره شد. با اعمال ارتباط اجتماعی منابع در یادگیری تقویتی در فرآیند تصمیم‌گیری مارکوف، نویسندگان توانستند بهبود قابل توجهی در عملکرد الگوریتم تخصیص منبع در جهت کاهش چرخه‌ی زمانی داشته باشند. در این فصل، پایگاه داده BPI Challenge 2012^۲ مورد بررسی قرار می‌گیرد و تلاش می‌شود با الهام از تحقیق Liu و همکاران، وابستگی بین وظایف در پایگاه داده بررسی شود. این وابستگی مربوط به وظایفی بود که هم‌زمان توسط یک منبع مشخص اجرا می‌شدند. روش استخراج و محاسبه‌ی وابستگی بین وظایف از روی سابقه‌ی رویدادهای ثبت‌شده در ادامه شرح داده خواهد شد. سپس، بعد از تخصیص وظایف توسط یک الگوریتم پایه، از الگوریتم تکمیلی پیشنهادشده جهت متعادل‌سازی مجدد بارکاری منابع استفاده می‌شود. این الگوریتم با در نظر گرفتن وابستگی تعریف‌شده بین وظایف، وظیفه‌های وابسته را بین لیست کاری منابع جابجا می‌کند تا توزیع مناسب‌تری از وظیفه‌ها حاصل شود. در توزیع حاصل‌شده امکان اجرای هم‌زمان وظایف برای منابع مربوطه به وجود می‌آید. به عبارت دیگر، تکنیک اجرای هم‌زمان وظایف وابسته^۳ (CPDT) ایده‌ی اصلی این فصل است. این تکنیک بر اساس تحلیل آماری سابقه‌ی رویدادها بنا شده است تا بتواند از اطلاعات سابقه‌ی رویدادهای ثبت‌شده، استفاده کند و وظایف را به شکل بهتری بین منابع توزیع کند و باعث کاهش چرخه‌ی زمانی درخواست‌ها شود. در این فصل، الگوریتم ارائه‌شده‌ی Byung-Hyun Ha و همکاران [۴] توسعه داده می‌شود و با الگوریتم DSWB که بر اساس شبکه‌ی صف بنا شده است، مقایسه می‌شود. تفاوت کلیدی الگوریتم ارائه‌شده در این تحقیق با روش‌های قبلی [۴، ۶] در استفاده آماری از سابقه‌ی رویدادها است که در این فصل به صورت استخراج فاصله‌ی وابستگی وظایف تعریف شده است و این فاصله‌ی وابستگی در الگوریتم ارائه‌شده نقش اصلی را دارد. در تحقیقی که به وسیله Xingmei و همکاران [۲۷] انجام شد نوعی ارتباط اجتماعی بین منابع وظیفه‌ی جاری یک درخواست با منابعی که روی همان درخواست کار انجام داده

^۱ Social Relation

^۲<http://data.4tu.nl/repository/uuid:3926db30-f712-4394-aebc-75976070e91f>

^۳ Concurrent Performing of Dependent Tasks

بودند تعریف شد و آن‌ها ارتباط اجتماعی بین منابع را بر اساس اطلاعات ثبت‌شده در سابقه‌ی رویدادها به دست آوردند. الگوریتم‌های ارائه‌شده در این بخش را می‌توان مشابه با تحقیق Xingmei و همکاران دانست با این تفاوت که آن‌ها از تحلیل آماری سابقه‌ی رویدادها، ارتباط اجتماعی بین منابع را به دست آوردند درحالی‌که در این بخش، فاصله‌ی وابستگی بین وظایف از سابقه‌ی رویدادها استخراج می‌شود.

۴-۱ مفاهیم و تعاریف

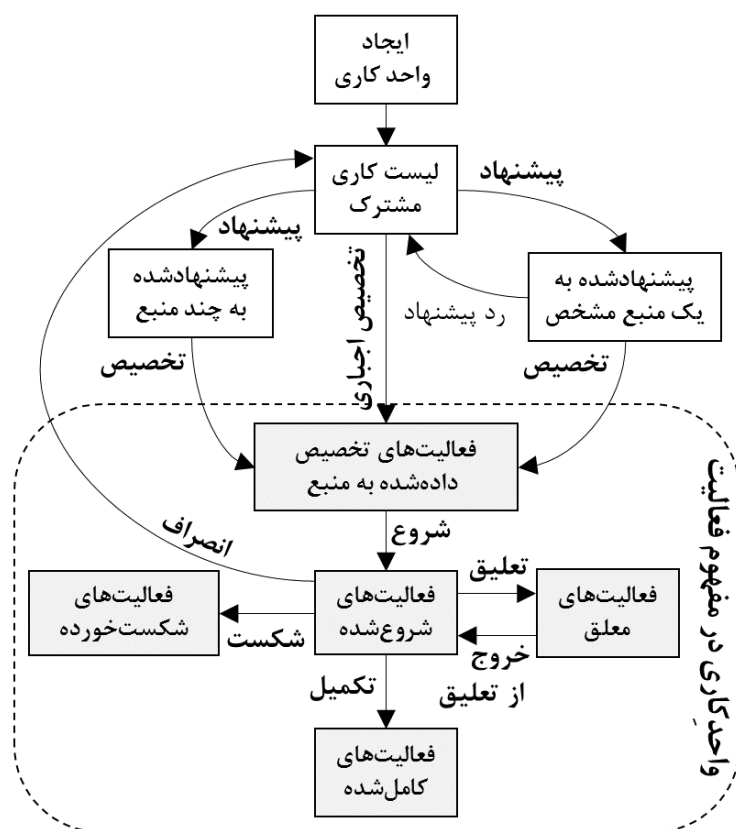
۴-۱-۱ مدل تحلیلی فرآیند

برای معرفی روش پیشنهادی و آشنایی با فضای مسئله‌ی تخصیص وظایف، لازم است ابتدا مدل تحلیلی فرآیند ارائه شود. مدل تحلیلی فرآیند به‌صورت چندتایی $p = \langle \varphi, T, F, R, U, \mu, \delta \rangle$ بیان می‌شود که اجزای آن عبارت‌اند از:

- ا. φ نرخ ورود مشتریان است.
- ب. مجموعه‌ی T وظیفه‌های تعریف‌شده در فرآیند کسب‌وکار است.
- ت. مجموعه‌ی $F \subseteq T \times T$ شامل اتصالات جهت‌دار بین وظیفه‌ها است که بیانگر جریان کاری وظایف تعریف‌شده در فرآیند کسب‌وکار است.
- ث. مجموعه‌ی R منابع انسانی موجود در سازمان است که وظیفه‌های تعریف‌شده در فرآیند سازمانی به‌وسیله آن‌ها انجام می‌شود.
- ج. مجموعه‌ی $U \subseteq T \times R$ منابع مسئول هر وظیفه را مشخص می‌کند. به‌عنوان مثال $\langle t, r \rangle \in U$ نشان می‌دهد در فرآیند p منبع r مسئول انجام دادن وظیفه‌ی t است.
- ح. $\mu: U \rightarrow \mathbb{R}^+$ یک نگاشت از مجموعه‌ی U به اعداد حقیقی مثبت است. مقدار $\mu(\langle t, r \rangle)$ بیانگر «متوسط زمان مستقل» انجام وظیفه‌ی t توسط منبع r است که به‌اختصار با $\mu^r(t)$ نشان داده می‌شود.
- خ. $\delta: U \times T \rightarrow \mathbb{R}^+$ یک نگاشت از مجموعه‌ی $U \times T$ به اعداد حقیقی مثبت است. مقدار $\delta(\langle t, r, t' \rangle)$ بیانگر «متوسط زمان وابسته» است که به انجام وظیفه‌ی t به‌صورت هم‌زمان با وظیفه‌ی t' توسط منبع r اشاره دارد که به‌اختصار با $\delta^r(t, t')$ نشان داده می‌شود.

در ادامه‌ی این تحقیق، روش محاسبه‌ی δ و μ شرح داده می‌شود و الگوریتم پیشنهادی بر پایه‌ی مقادیر δ و μ بیان می‌گردد.

وقتی درخواست مشتری به یک فرآیند می‌رسد، یک نمونه فرآیند ایجاد می‌شود. هم‌زمان با ایجاد نمونه فرآیند، یک واحد کاری در اولین وظیفه‌ی تعریف‌شده در فرآیند، ایجاد می‌گردد. این واحد کاری در چرخه‌ی حیات خود به رویدادهای زیادی برخورد خواهد کرد. مهم‌ترین این رویدادها، وقتی است که یک منبع به واحد کاری تخصیص داده می‌شود. بعد از این تخصیص، واحد کاری مفهوم «فعالیت» را به خود می‌گیرد. شکل (۴-۱) چرخه‌ی حیات یک واحد کاری را نشان می‌دهد، همان‌طور که در شکل مشخص است یک واحد کاری دارای وضعیت‌های مختلفی است که با رخ دادن هر رویداد (یال) وضعیت آن عوض می‌شود. در شکل (۴-۱) وضعیت‌هایی که واحد کاری دارای مفهوم فعالیت است مشخص شده است.



شکل (۴-۱) چرخه‌ی حیات یک واحد کاری

بعد از رویداد تخصیص، یکی از رویدادهای شروع، تکمیل، انصراف، شکست و تعلیق روی واحد کاری (فعالیت) رخ می‌دهد. در BPMS هنگام رخ دادن هر یک از این رویدادها، اطلاعات مربوط به آن در سابقه‌ی رویداد ذخیره می‌شود.

۴-۱-۲ رویداد ذخیره شده

پنج‌تایی $e = \langle Type, Case, Task, Time, ResourceId \rangle$ یک رویداد ذخیره‌شده در سابقه‌ی رویداد است. این رویداد هنگام ایجاد، انتساب، شروع، پایان هر فعالیت در پایگاه داده ثبت می‌شود. اجزای یک رویداد ذخیره شده عبارت‌اند از:

- **Type:** نوع رویداد را مشخص می‌کند که می‌تواند شامل یکی از مقادیر Allocation, Start و Complete, Fail, Cancel, Suspend و Resume باشد.
 - **Case:** شناسه‌ی درخواست (مورد) است که به ازای هر درخواست ورودی یک عدد یکتا است و مشخص می‌کند رویداد ثبت‌شده مربوط به کدام درخواست است.
 - **Task:** وظیفه‌ی متناظر با فعالیتی است که رویداد e هنگام انجام آن فعالیت ثبت شده است.
 - **Time:** زمان ثبت رویداد است.
 - **ResourceId:** شناسه‌ی منبعی است که فعالیت مربوط به رویداد e را انجام داده است.
- در ادامه، یک رویداد با نماد e نشان داده خواهد شد و برای دسترسی به نوع و درخواستی (موردی) که رویداد روی آن اتفاق افتاده به ترتیب از توابع $e.Type$ و $e.Case$ استفاده خواهد شد.

۴-۱-۳ استخراج اطلاعات فعالیت‌ها از سابقه‌ی رویداد

به ازای هر درخواستی که به فرآیند کسب‌وکار وارد می‌شود، یک نمونه فرآیند ایجاد می‌شود. هر فعالیت در نمونه فرآیند با یک وظیفه در مدل تحلیلی فرآیند متناظر می‌شود. اگر مجموعه‌ی فعالیت‌های موجود در نمونه فرآیند با A نشان داده شود آنگاه مجموعه‌ی A با مجموعه‌ی T در مدل تحلیلی فرآیند متناظر می‌شود. عملیاتی که روی فعالیت $a \in A$ در BPMS انجام می‌شود باعث ثبت مجموعه‌ای از رویدادها در سابقه‌ی رویداد می‌گردد. هر رویداد ثبت‌شده‌ی e شامل اطلاعات درخواست (مورد)، منبع

و وظیفه است که به ترتیب به کمک سه تابع $e.Case$ ، $e.ResourceId$ و $e.Task$ قابل دسترسی است. این سه مقدار برای کلیه رویدادهای یک فعالیت یکسان هستند. در ادامه برای دسترسی راحت تر به این اطلاعات برای هر فعالیت a از توابع $a.Case$ ، $a.ResourceId$ و $a.Task$ استفاده خواهد شد. در هر فعالیت کامل شده حداقل یک رویداد با نوع Start و یک رویداد با نوع Complete وجود دارد. که به ترتیب مقادیر $a.Start$ و $a.Complete$ را مشخص می کنند. رابطه‌ی (۴-۱) نحوه‌ی مقداردهی $a.Start$ و $a.Complete$ را نشان می دهد.

$$\begin{aligned} \forall a \in A, \exists e \in Eventlog, \left(\begin{array}{l} e.Type = Start, \\ e \text{ occurs on performing } a \end{array} \right) &\Rightarrow a.Start = e.Time \\ \forall a \in A, \exists e \in Eventlog, \left(\begin{array}{l} e.Type = Complete, \\ e \text{ occurs on performing } a \end{array} \right) &\Rightarrow a.Complete = e.Time \end{aligned} \quad (۴-۱)$$

۴-۲ تخمین زمان اجرای وظایف

با تحلیل آماری رویدادهای ثبت شده در سابقه‌ی رویداد، امکان محاسبه‌ی میانگین مدت زمان اجرای هر وظیفه وجود دارد و این میانگین می تواند تخمینی مناسب از مدت زمان اجرای وظیفه‌ی جاری باشد. رابطه‌ی (۴-۲) روش محاسبه‌ی تخمینی مدت اجرای وظیفه‌ی t را توسط منبع r نشان می دهد.

$$\begin{aligned} \mu^r(t) &= \frac{\sum_{a \in AC^{t,r}} (a.Complete - a.Start)}{|AC^{t,r}|}, \\ AC^{t,r} &= \{a | a \in eventlog, a.Task = t, a.resourceId = r\} \end{aligned} \quad (۴-۲)$$

۴-۲-۱ تخمین زمان اجرای وظایف مستقل

در رابطه‌ی (۴-۲) اگر هم زمان با انجام فعالیت $a \in EventLog$ که توسط منبع r انجام شده است؛ هیچ فعالیت دیگری توسط همین منبع انجام نشده باشد آنگاه رابطه‌ی (۴-۲) مقدار «تخمین زمان مستقل» برای وظیفه‌ی t را محاسبه می کند که توسط منبع r انجام شده است.

۴-۲-۲ تخمین زمان اجرای وظایف وابسته

بررسی پایگاه داده BPI Challenge 2012 نشان داد، گاهی اوقات در اجرای فرآیندهای سازمانی ممکن است دو فعالیت a و b توسط یک منبع مشخص r به طور هم زمان انجام شود. اگر $A_t^r = \{a | a \in EventLog, a.Task = t, a.ResourceId = r\}$ مجموعه‌ی تمام فعالیت های متناظر با وظیفه‌ی t که توسط منبع r انجام شده است، باشد آنگاه برای دو وظیفه‌ی مفروض t_1 و t_2 ،

مجموعه‌ی $\langle a, b \rangle \in A_{t_1}^r \times A_{t_2}^r$ جفت فعالیت‌هایی است که توسط منبع r انجام شده است. حال اگر یکی از دو شرط ذیل برقرار باشد آنگاه فعالیت‌های a و b توسط منبع r هم‌زمان انجام شده‌اند و وظیفه‌های t_1 و t_2 وابسته هستند.

الف) اگر زمان شروع و زمان تکمیل انجام یک فعالیت کاملاً بین زمان شروع و زمان تکمیل فعالیت دیگر باشد. به عبارت دیگر

$$a.Start \leq b.Start \leq \quad \text{یا} \quad b.Start \leq a.Start \leq a.Complete \leq b.Complete \\ b.Complete \leq a.Complete$$

ب) اگر بخشی از ابتدا یا انتهای یک فعالیت در زمان انجام فعالیت دیگری باشد. به عبارت دیگر

$$a.Start \leq b.Start \leq \quad \text{یا} \quad a.Start \leq b.Start \leq a.Complete \leq b.Complete \\ a.Complete \leq b.Complete$$

طبق شرط‌های فوق، مجموعه‌ی $A_{t_1}^r \phi A_{t_2}^r$ به عنوان یک زیرمجموعه از $A_{t_1}^r \times A_{t_2}^r$ تعریف می‌شود، اگر شامل همه‌ی فعالیت‌های هم‌زمان دو وظیفه‌ی وابسته‌ی t_1 و t_2 باشد و مطابق با رابطه‌ی (۳-۴) تعریف می‌گردد.

$$A_{t_1}^r \phi A_{t_2}^r = \left\{ (a, b) \mid a \in A_{t_1}^r, b \in A_{t_2}^r, \begin{pmatrix} a.Start \leq b.Start \leq b.Complete \leq a.Complete \\ b.Start \leq a.Start \leq a.Complete \leq b.Complete \\ a.Start \leq b.Start \leq a.Complete \leq b.Complete \\ b.Start \leq a.Start \leq b.Complete \leq a.Complete \end{pmatrix} \right\} \quad (3-4)$$

برای هر وظیفه‌ی مستقل t عبارت $A_t^r \phi A_{t_k}^r = \emptyset, \forall t_k \in T$ برقرار است. برای یک وظیفه‌ی غیرمستقل t' می‌بایست ابتدا مجموعه‌ی فعالیت‌های مستقل $(inA_{t'}^r)$ و مجموعه‌ی فعالیت‌های غیرمستقل $(deA_{t'}^r)$ به دست آید، سپس زمان اجرای وابسته و مستقل آن قابل محاسبه است. در رابطه‌های (۴-۴) و (۵-۴) روش محاسبه‌ی آن‌ها آمده است.

$$deA_{t'}^r = \{a \mid a \in A_{t'}^r, \exists \langle t, r \rangle \in U, \exists b \in A_t^r, \langle a, b \rangle \in A_{t'}^r \phi A_t^r\} \quad (4-4)$$

$$inA_{t'}^r = A_{t'}^r - deA_{t'}^r \quad (5-4)$$

برای تخمین زمان مستقل یک وظیفه‌ی غیرمستقل t' کافی است در رابطه‌ی (۲-۴) به جای

عبارت $a \in EventLog$ از عبارت $a \in inA_{t'}^r$ استفاده شود. در نتیجه تخمین زمان مستقل یک

وظیفه‌ی غیرمستقل t' از رابطه‌ی (۶-۴) حاصل می‌شود.

$$\mu^r(t') = \frac{\sum_{a \in AC^{t',r}} (a.Complete - a.Start)}{|AC^{t',r}|}, \quad (6-4)$$

$$AC^{t',r} = \{a | a \in \mathbf{inA}_{t'}, a.Task = t', a.resourceId = r\}$$

برای یک وظیفه‌ی غیرمستقل (وابسته) t' می‌توان «زمان اجرای وابسته» تعریف کرد و برای

تخمین آن لازم است توابع T_+ , T_U و $T_\cap$ برای دو فعالیت a و b تعریف شود.

$$T_+(a, b) = (a.Complete - a.Start) + (b.Complete - b.Start) \quad (7-4)$$

$$T_\cap(a, b) = \min(a.Complete - b.Complete) - \max(a.Start, b.Start) \quad (8-4)$$

$$T_U(a, b) = T_+(a, b) - T_\cap(a, b) \quad (9-4)$$

$$\delta^r(t', t) = \frac{\sum_{\langle a, b \rangle \in A_{t'}^r, \phi A_t^r} T_U(a, b)}{|A_{t'}^r, \phi A_t^r|} \quad (10-4)$$

در رابطه‌ی (۱۰-۴) تابع $\delta^r(t', t)$ تخمین زمان اجرای وظیفه‌ی وابسته‌ی t' هم‌زمان با وظیفه‌ی

t است، برای محاسبه‌ی $\delta^r(t', t)$ ، میانگین زمان اجراهای هم‌زمان جفت فعالیت‌های $\langle a, b \rangle$ متناظر با

(t', t) محاسبه شده است. برای محاسبه‌ی زمان اجرای جفت فعالیت‌های $\langle a, b \rangle$ ابتدا زمان مجموع

آن‌ها با تابع $T_+(a, b)$ محاسبه می‌شود، سپس بخشی از دو فعالیت که به‌صورت هم‌زمان در حال اجرا

بودند ($T_\cap(a, b)$) از آن کسر می‌شود تا مقدار اجرای هم‌زمان دو فعالیت در رابطه‌ی (۹-۴) محاسبه

شود. در رابطه‌ی (۱۰-۴) میانگین مقادیر $T_U(a, b)$ برای تمام جفت فعالیت‌های هم‌زمان که توسط

منبع r انجام شده است، محاسبه می‌گردد. مقدار $\delta^r(t', t)$ به‌عنوان تخمین زمان وابسته‌ی وظیفه‌های

t' و t که توسط منبع r انجام شده است، در نظر گرفته می‌شود.

۴-۲-۳ فاصله‌ی وابستگی وظایف

وقتی وظیفه‌ی وابسته‌ی t' به‌طور هم‌زمان با وظیفه‌ی t توسط منبع r انجام می‌شود، اگر مقدار

$\delta^r(t', t)$ کوچک‌تر از مقدار $\mu^r(t') + \mu^r(t)$ باشد آنگاه بهتر است منبع r وظیفه‌ی t' را هم‌زمان با

وظیفه‌ی t انجام دهد و منبع r مدت $\mu^r(t') + \mu^r(t) - \delta^r(t', t)$ در زمان انجام وظایف t و t'

صرفه‌جویی کرده است. از طرف دیگر اگر $\delta^r(t', t)$ بزرگ‌تر از مقدار $\mu^r(t') + \mu^r(t)$ باشد آنگاه بهتر

است وظیفه‌ی وابسته‌ی t' به‌صورت مستقل اجرا شود تا باعث بهبود در چرخه‌ی زمانی شود. نکته‌ی

قابل‌توجه دیگر این است که ممکن است یک وظیفه برای یک منبع مستقل و برای منبعی دیگر

غیرمستقل باشد و مقادیر $\delta^r(t', t)$ و $\mu^r(t') + \mu^r(t)$ با توجه به منبع r متفاوت باشند. بنابراین نسبت $\frac{\delta^r(t', t)}{\mu^r(t') + \mu^r(t)}$ به عنوان یک معیار مناسب، تعیین می کند که وظیفه‌ی وابسته‌ی t' به صورت جداگانه انجام شود یا به صورت همزمان با وظیفه‌ی دیگر t انجام شود.

از آنجایی که نسبت $\frac{\delta^r(t', t)}{\mu^r(t') + \mu^r(t)}$ یک عدد حقیقی در بازه‌ی $(0, +\infty)$ است و مقدار یک به عنوان مرز تشخیص، بازه‌ی $(1, +\infty)$ را به دو بازه‌ی $(0, 1)$ و $(1, +\infty)$ با طول غیر برابر تقسیم می کند، لازم است نسبت $\frac{\delta^r(t', t)}{\mu^r(t') + \mu^r(t)}$ با استفاده از رابطه‌ی (۴-۱۱) روی بازه‌ی $(0, 1)$ نگاشت شود تا مرز تشخیص به مقدار $0/5$ منتقل گردد و بازه‌های $(0, 1)$ و $(1, +\infty)$ به ترتیب روی بازه‌های $(0/5, 1)$ و $(1/5, 0)$ نگاشت شود.

$$DF_r(t', t) = \frac{1}{1 + \frac{\delta^r(t', t)}{\mu^r(t') + \mu^r(t)}} \quad (4-11)$$

در رابطه‌ی (۴-۱۱) تابع $DF_r(t', t)$ را «فاصله‌ی وابستگی» می نامیم اگر فاصله‌ی وابستگی کم تر از $0/5$ باشد آنگاه مقدار $\frac{\delta^r(t', t)}{\mu^r(t') + \mu^r(t)}$ بیش تر از یک می شود و بهتر است وظیفه‌ی وابسته‌ی t' به صورت جداگانه انجام شود. اگر فاصله‌ی وابستگی بیش تر از $0/5$ باشد بهتر است وظیفه‌ی وابسته‌ی t' همزمان با وظیفه‌ی t انجام شود و برای فاصله‌ی وابستگی $0/5$ فرقی نمی کند وظیفه‌ی وابسته‌ی t' به صورت جداگانه انجام شود یا با وظیفه‌ی دیگر t به صورت همزمان انجام شود.

۴-۳ تعادل بارکاری و همزمان سازی وظایف وابسته

ایجاد تعادل در بارکاری منابع و کاهش چرخه‌ی زمانی درخواست‌ها از مهم ترین اهداف هر سیستم مدیریت فرآیندهای کسب و کار است. اگر یک سیستم به دنبال برقراری تعادل بارکاری باشد آنگاه بهینه سازی چرخه‌ی زمانی به خوبی میسر نمی شود. از طرف دیگر، ایجاد چرخه‌ی زمانی کمینه باعث غیرمتعادل شدن بارکاری می شود [۲۸]. وقتی برقراری تعادل بارکاری در سیستم هدف اصلی باشد، به چند دلیل ممکن است بهینه سازی چرخه‌ی زمانی با مشکل مواجه شود. این دلایل از این قرارند:

الف) کاهش چرخه‌ی زمانی به صورت تئوری در معادلات تعادل بارکاری پیش بینی نشده است.

ب) ایجاد تعادل بارکاری بر اساس مقادیر تخمینی در زمان تخصیص منابع انجام می‌شود و کارها بر اساس این مقادیر در لیست کاری منابع قرار می‌گیرند. ولی در زمان اجرا، ممکن است یک لیست کاری زودتر از زمان تخمینی، تمام شود و منبع یا منابعی برای مدتی غیرفعال باشند. به عبارت دیگر از توان حداکثری منابع استفاده نمی‌شود.

Byung-Hyun Ha و همکاران [۴] با تخمین بارکاری منابع بر اساس مدل شبکه‌ی صف و ارائه‌ی الگوریتم تعادل بارکاری جدید (متعادل‌سازی مجدد در زمان اجرا) توانستند چرخه‌ی زمانی درخواست‌ها را به میزان قابل‌توجهی کاهش دهند. در این فصل، هدف مشابهی دنبال می‌شود و اثر نامطلوب (دو دلیل ذکر شده) ایجاد تعادل بارکاری با تخصیص هم‌زمان وظایف وابسته جبران می‌شود و چرخه‌ی زمانی درخواست‌ها کاهش می‌یابد. در این تحقیق، الگوریتم ارائه‌شده‌ی Byung-Hyun Ha و همکاران [۴] توسعه داده می‌شود. به‌طوری‌که جابجایی هوشمندانه‌ای از وظایف بین لیست‌های کاری منابع بعد از تخصیص اولیه‌ی آن‌ها انجام می‌شود که علاوه بر ایجاد تعادل در بارکاری با تخصیص هم‌زمان وظایف وابسته، باعث کاهش چرخه‌ی زمانی می‌گردد. الگوریتم ارائه‌شده ($CPDT^1$) شامل مراحل ذیل است:

۱. انتخاب تصادفی یک منبع غیرفعال $r \in R$ از مجموعه‌ی منابع تعریف‌شده در فرآیند.
۲. محاسبه‌ی مجموعه‌ی T_r به‌عنوان وظایفی که منبع انتخاب‌شده مجاز به انجام آن‌ها است.

$$(T_r = \{t | \langle t, r \rangle \in U\})$$
۳. اگر حداقل دو فعالیت متناظر با دو وظیفه‌ی وابسته‌ی t, t' در مجموعه‌ی T_r وجود داشته باشد آنگاه دو فعالیت وابسته در صورتی که $DF_r(t', t)$ بیش‌تر 0.5 باشد، انتخاب می‌شوند و رفتن به مرحله‌ی ۵.
۴. انتخاب یک فعالیت متناظر با یک وظیفه‌ی دلخواه t از مجموعه‌ی T_r
۵. قرار دادن فعالیت‌های انتخاب‌شده در لیست کاری منبع r
۶. رفتن به مرحله‌ی ۱ در صورتی که حداقل یک منبع غیرفعال وجود داشته باشد.

الگوریتم ارائه‌شده مشابه الگوریتمی است که Byung-Hyun Ha و همکاران [۴] ارائه دادند با این تفاوت که مرحله‌ی ۳ به آن اضافه شده است. در این مرحله، الگوریتم سعی می‌کند یک جفت فعالیت را انتخاب کند به‌طوری‌که اجرای هم‌زمان آن‌ها توسط منبع r باعث می‌شود زمان اجرای هر دو فعالیت کم‌تر شود و در مجموع باعث کاهش چرخه‌ی زمانی می‌گردد. از آنجایی که الگوریتم ارائه‌شده یک روش

¹ Concurrent Performing of Dependent Tasks

تکمیلی بعد از تخصیص اولیه وظایف به منابع است، امکان ترکیب آن با هر روش دیگری (در مرحله‌ی اولیه وظایف را به منابع تخصیص می‌دهد) وجود دارد.

۱-۳-۴ بهبود الگوریتم هم‌زمان‌سازی وظایف وابسته

در الگوریتم CPDT برای جابجایی وظایف وابسته از منابع غیرفعال استفاده شده است. برای افزایش احتمال جابجایی وظایف وابسته می‌توان مجموعه منابع موجود را بر اساس متوسط بارکاری منابع به دو دسته تقسیم کرد و وظایف وابسته را از منابع با بارکاری کم‌تر از میانگین به منابع با بارکاری بیش‌تر از میانگین منتقل کرد. برای این منظور می‌توان الگوریتم بهبودیافته (CPDT2) را به صورت ذیل ارائه داد.

Algorithm: Workload rebalancing by reallocating dependent tasks (CPDT2)

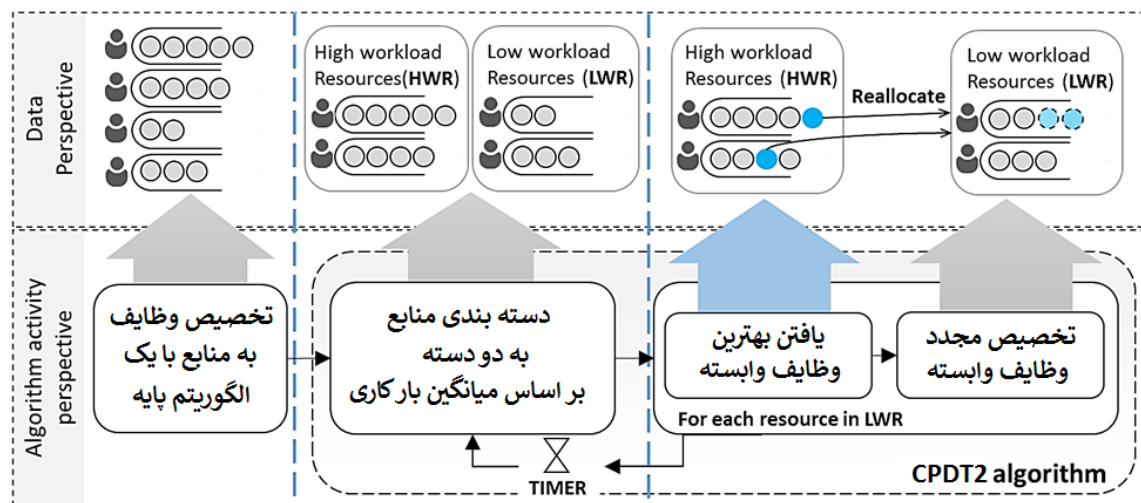
Input: business process model $p = \langle \varphi, T, F, R, U, \mu, \delta, Q, wl \rangle$, *Timer* as integer

```

1: while(true){
2:    $\overline{wl} \leftarrow \sum_{r \in R} wl(r) / |R|$ 
3:    $LWR \leftarrow \{r | r \in R, wl(r) < \overline{wl}\}$  //low workload resources
4:    $HWR \leftarrow \{r | r \in R, wl(r) \geq \overline{wl}\}$  //high workload resources
5:    $allTaskInHWR \leftarrow \{\langle a, b \rangle | a, b \in \bigcup_{r \in HWR} Q^r\}$ 
6:    $r \leftarrow \arg \min_{res \in LWR} (wl(res))$ 
7:    $\langle a, b \rangle \leftarrow \arg \max_{\langle a, b \rangle \in (allTaskInHWR \cap A_t^r \varphi A_{t'}^r)} (\mu^r(t') + \mu^r(t) - \delta^r(t', t))$ 
8:    $m \leftarrow \max_{\langle a, b \rangle \in (allTaskInHWR \cap A_t^r \varphi A_{t'}^r)} (\mu^r(t') + \mu^r(t) - \delta^r(t', t))$ 
9:   if ( $m > 0$ )
10:     $Q^r \leftarrow Q^r \cup \{a, b\}$  //reallocate to resource  $r$ 
11:   else{
12:     $r_2 \leftarrow \arg \min_{res \in (LWR - \{r\}), \langle b, res \rangle \in U} (wl(res))$ 
13:     $Q^r \leftarrow Q^r \cup \{a\}$  //reallocate to resource  $r$ 
14:     $Q^{r_2} \leftarrow Q^{r_2} \cup \{b\}$  //reallocate to resource  $r_2$ 
15:   }//if
16:    $sleep(Timer)$ 
17: }//while

```

برای توصیف بهتر الگوریتم CPDT2 در شکل (۴-۲) مراحل اجرای الگوریتم از دو دیدگاه فعالیت^۱ و داده^۲ نشان داده شده است. همان‌طور که در الگوریتم و شکل مربوطه مشخص است بعد از تخصیص اولیه واحدهای کاری با یک الگوریتم تخصیص پایه، مجموعه‌ی لیست‌های کاری منابع بر اساس مقدار متوسط بار کاری به دو مجموعه HWR و LWR به ترتیب مجموعه‌ی منابع با بار کاری بالا و مجموعه‌ی منابع با بار کاری پایین تقسیم می‌شوند. سپس بهترین جفت وظیفه‌ی وابسته انتخاب می‌شود و بر اساس مقدار $\mu^r(t') + \mu^r(t) - \delta^r(t', t)$ جفت وظیفه‌ی انتخاب شده را به یک یا دو منبع مناسب از مجموعه‌ی LWR تخصیص داده می‌شود.



شکل (۴-۲) مراحل اجرای الگوریتم بهبود یافته هم‌زمان‌سازی وظایف وابسته

در ادامه این فصل در بخش ۴-۴ الگوریتم‌های تکمیلی CPDT و CPDT2 با سایر الگوریتم موجود در سابقه‌ی پژوهش مورد مقایسه قرار خواهد گرفت.

^۱ Activity perspective

^۲ Data perspective

۴-۴ آزمایش‌ها و نتایج

در این بخش برای ارزیابی روش‌های پیشنهادی پنج آزمایش (چهار آزمایش برای مقایسه‌ی الگوریتم‌های تکمیلی و یک آزمایش برای الگوریتم یادگیر با هدف کاهش بی‌نظمی لیست کاری) متفاوت روی سه فرآیند «مالی»، «تعیین اعتبار» و «تولید» انجام می‌شود و نتایج به‌صورت جداگانه گزارش خواهد شد. جدول (۱-۴) فهرست چهار آزمایش و الگوریتم‌هایی را نشان می‌دهد که قرار است در این آزمایش‌ها مورد مقایسه قرار بگیرند. کلیه‌ی آزمایش‌های گزارش‌شده در این تحقیق به کمک شبیه‌ساز «نما» انجام شده است. این شبیه‌ساز بخشی از یک BPMS با عنوان «نما» است که در آزمایشگاه داده‌کاوی دانشکده‌ی کامپیوتر و فن‌آوری اطلاعات دانشگاه صنعتی شاهرود طراحی شده است. در هر آزمایش، هر یک از الگوریتم‌های جدول (۱-۴)، ۱۰ مرتبه در شبیه‌ساز «نما» ارزیابی می‌شوند. در هر مرتبه ۲۰۰ درخواست با نرخ‌های ورود متفاوت (به ازای هر نرخ ورود ۴ بار) وارد فرآیندهای تعریف‌شده می‌شوند. در بخش‌های ۱-۴ الی ۳-۴ چهار آزمایش روی سه فرآیند متفاوت با پایگاه داده‌های متفاوت، گزارش شده است. در آزمایش اول هفت الگوریتم مورد ارزیابی قرار می‌گیرند، در آزمایش دوم و سوم، سه الگوریتم برتر از آزمایش اول مورد ارزیابی مجدد قرار می‌گیرند و در آزمایش چهارم الگوریتم‌های پیشنهادی با الگوریتم DSWB مقایسه می‌شوند. در آزمایش پنجم نیز الگوریتم پیشنهادی در بخش **Error! Reference source not found.** تحت عنوان Min Entropy+MDPs با الگوریتم Min Workload+MDPs مورد مقایسه قرار می‌گیرند.

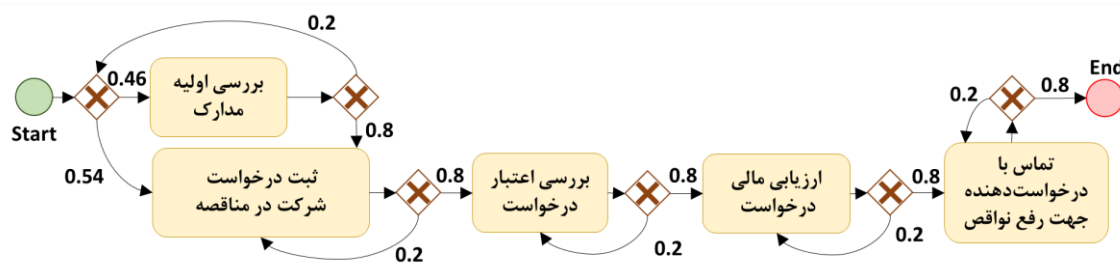
جدول (۱-۴) الگوریتم‌های مقایسه شده

عنوان	الگوریتم تخصیص		پایگاه داده	توضیحات
	اصلی	تکمیلی		
آزمایش اول (فرآیند مالی)	Random	×	BPI Challenge 2012 [27]	تخصیص تصادفی وظایف به منبع
	Ordered	×		تخصیص ترتیبی وظایف به منبع
	WB_FIFO	×		تخصیص بر اساس شبکه‌ی صف با هدف تعادل بارکاری [۴]
	QSR ^۱	×		تخصیص با یادگیری تقویتی بر پایه‌ی ارتباط اجتماعی منابع [۲۷]
	DSWB	×		تخصیص تصادفی پویا با هدف تعادل بارکاری [۶]
	WB_FIFO	DR		تخصیص بر اساس شبکه‌ی صف و روش تکمیلی تعادل مجدد در زمان اجرا [۴]
	WB_FIFO	CPDT		الگوریتم پیشنهادشده در بخش ۳-۴
آزمایش دوم (فرآیند تعیین اعتبار)	DSWB	×	داده‌های شبیه‌سازی [۴]	تخصیص تصادفی پویا با هدف تعادل بارکاری [۶]
	WB_FIFO	DR		تخصیص بر اساس شبکه‌ی صف با روش تکمیلی تعادل مجدد در زمان اجرا [۴]
	WB_FIFO	CPDT		الگوریتم پیشنهادشده در بخش ۳-۴
آزمایش سوم (فرآیند تولید)	DSWB	×	داده‌های شبیه‌سازی [۶]	تخصیص تصادفی پویا با هدف تعادل بارکاری [۶]
	WB_FIFO	DR		تخصیص بر اساس شبکه‌ی صف با روش تکمیلی تعادل مجدد در زمان اجرا [۴]
	WB_FIFO	CPDT		الگوریتم پیشنهادشده در بخش ۳-۴
آزمایش چهارم (فرآیند مالی)	DSWB		BPI Challenge 2012 [27]	تخصیص تصادفی پویا با هدف تعادل بارکاری [۶]
	WB_FIFO			
	QSR			تخصیص با یادگیری تقویتی بر پایه‌ی ارتباط اجتماعی منابع [۲۷]
	WB_FIFO	DR		تخصیص بر اساس شبکه‌ی صف با روش تکمیلی تعادل مجدد در زمان اجرا [۴]
	WB_FIFO	CPDT		الگوریتم پیشنهادشده در بخش ۳-۴
	WB_FIFO	CPDT2		الگوریتم پیشنهادشده در بخش ۱-۳-۴

^۱ Q-learning for task allocation based on Social Relation

۴-۱-۴ آزمایش اول: فرآیند مالی

در این آزمایش از فرآیند واقعی «مالی» شکل (۳-۴) استفاده کرده‌ایم. این فرآیند شامل پایگاه داده‌ی BPI Challenge 2012 است. این پایگاه داده توسط یک موسسه‌ی مالی در کشور هلند ایجاد شده است که شامل ۵ وظیفه، ۵۵ منبع و ۱۳۰۸۷ درخواست است که در مجموع شامل ۲۶۲۲۰ رویداد ذخیره شده است.



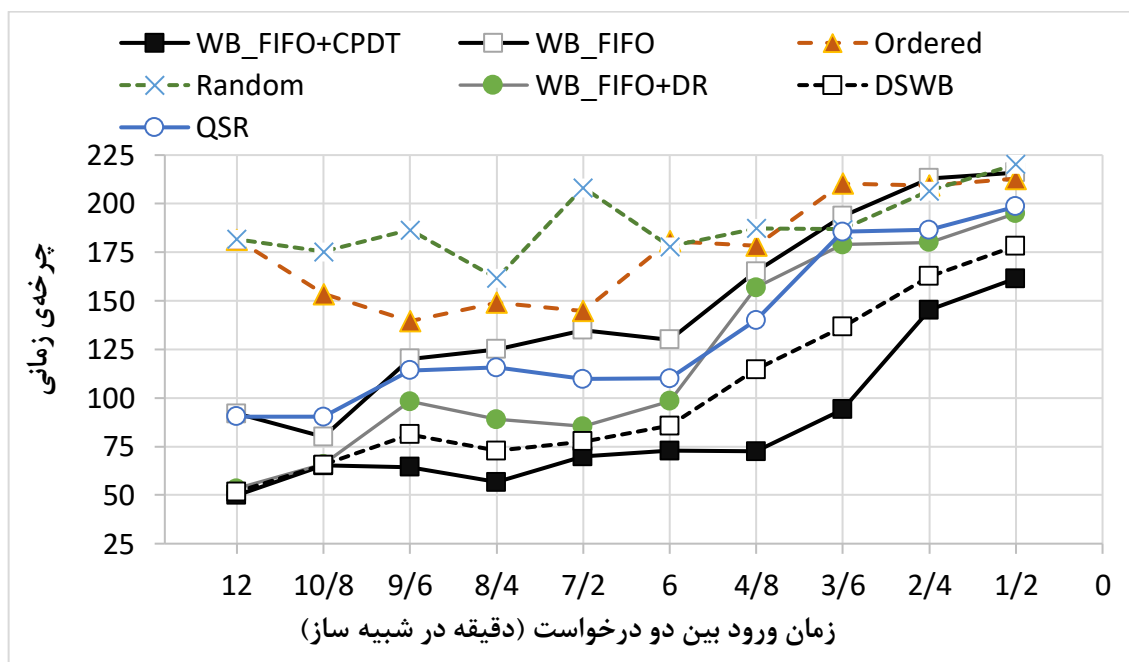
شکل (۳-۴) فرآیند موسسه‌ی مالی هلند [۲۷]

در این آزمایش مقادیر فاصله‌ی وابستگی برای پنج وظیفه‌ی تعریف شده در این پایگاه داده به تفکیک هر منبع محاسبه شد به‌عنوان مثال در جدول (۲-۴) مقادیر فاصله‌ی وابستگی برای یکی از منابع (منبع با شناسه‌ی ۱۱۱۲۱) در پایگاه داده‌ی BPI Challenge 2012 را نشان می‌دهد. در این جدول برخی از وظایف در پایگاه داده به‌صورت هم‌زمان توسط منبع ۱۱۱۲۱ انجام نشده‌اند لذا فاصله‌ی وابستگی برای آن‌ها قابل محاسبه نیست و با علامت $\times$ مشخص شده است. در مواردی که وظایف توسط منبع مربوطه به‌صورت هم‌زمان انجام شده باشد و مقادیر بزرگ‌تر از ۰/۵ باشد، به این معنی است که اجرای هم‌زمان آن‌ها زمان کم‌تری را نسبت به اجرای جداگانه آن‌ها خواهد داشت لذا در تخصیص وظایف به‌صورت هم‌زمان به منبع ۱۱۱۲۱، مقادیر جدول (۲-۴) نقش تعیین‌کننده‌ای را ایفا می‌کنند.

در این تحقیق، فعالیت‌هایی که به‌صورت هم‌زمان به‌وسیله هر یک از منابع در پایگاه داده‌ی BPI Challenge 2012 انجام شده بودند، مورد بررسی قرار گرفت. ۴۸۲۳ جفت فعالیت (۱۵/۴ درصد) از مجموع ۶۲۵۶۱ فعالیت به‌صورت هم‌زمان توسط یک منبع یکسان انجام شده بودند. از این تعداد فعالیت هم‌زمان، ۱۰/۳ درصد از کل فعالیت‌ها به‌صورت میانگین، دارای فاصله‌ی وابستگی ۰/۶۲۲ بودند و ۵/۱ درصد از کل فعالیت‌ها به‌صورت میانگین، دارای فاصله‌ی وابستگی ۰/۴۷۱ بودند.

جدول (۲-۴) مقادیر فاصله‌ی وابستگی محاسبه شده برای منبع با شناسه‌ی ۱۱۲۱

$DF_r(t', t)$	تماس با درخواست دهنده جهت رفع نواقص	ارزیابی مالی درخواست	درخواست دهنده بررسی اعتبار	ثبت درخواست شرکت در مناقصه	بررسی اولیه مدارک
بررسی اولیه مدارک	۰/۷۹۶	×	۰/۴۱۶	۰/۷۰۲	۰/۳۵۵
ثبت درخواست شرکت در مناقصه	×	×	۰/۶۴۷	×	۰/۷۰۲
بررسی اعتبار درخواست دهنده	۰/۶۹۶	×	۰/۱۹۳	۰/۶۴۷	۰/۴۱۶
ارزیابی مالی درخواست	×	×	×	×	×
تماس با درخواست دهنده جهت رفع نواقص	۰/۶۰۲	×	۰/۶۹۶	×	۰/۷۹۶



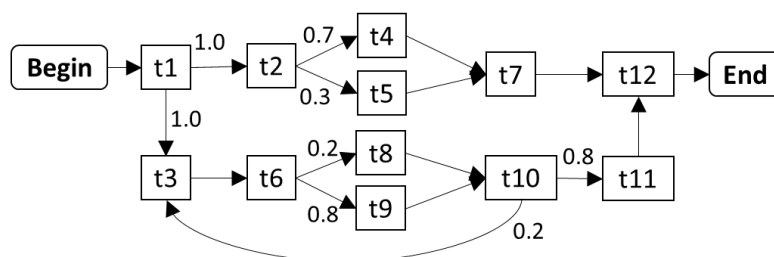
شکل (۴-۴) نتایج آزمایش اول روی فرآیند واقعی موسسه‌ی مالی هلند

در این آزمایش الگوریتم‌های Ordered، Random، WB_FIFO [۴]، WB_FIFO+DR [۴]، QSR [۲۷]، DSWB [۶] و الگوریتم پیشنهادی WB_FIFO+CPTD مورد مقایسه قرار گرفتند. همان‌طور که در شکل (۴-۴) مشخص است انتخاب فعالیت‌های وابسته باعث شد روش پیشنهادی WB_FIFO+CPDT نتایج بهتری را نسبت به سایر الگوریتم‌ها به خصوص در شرایط بار سنگین (مقادیر

کمتر زمان ورود بین دو درخواست داشته باشد. در این مقایسه، بعد از الگوریتم پیشنهادی به ترتیب الگوریتم‌های DSWD و WB_FIFO+DR نتایج بهتری داشتند.

۲-۴-۴ آزمایش دوم: فرآیند تعیین اعتبار

از آنجایی که الگوریتم ارائه شده در بخش ۳-۴، روش پیشنهادی تحقیق Byung-Hyun Ha و همکاران [۴] را توسعه می‌دهد، لذا در آزمایش دوم، الگوریتم ارائه شده روی فرآیند معرفی شده توسط Byung-Hyun Ha و همکاران [۴] مورد ارزیابی قرار می‌گیرد و روش پیشنهادی WB_FIFO+CPDT با روش‌های WB_FIFO+DR [۴] و DSWD [۶] که در آزمایش اول نتایج خوبی داشتند، مجدداً مقایسه می‌شود.



شکل (۵-۴) فرآیند تعیین اعتبار [۴]

فرآیند «تعیین اعتبار» به عنوان معیار ارزیابی روش WB_FIFO+DR توسط Byung-Hyun Ha و همکاران [۴] استفاده شده است. آن‌ها برای ارزیابی روش خود (WB_FIFO+DR) از داده‌های شبیه‌سازی استفاده کرده‌اند. در جدول‌های (۳-۴) و (۴-۴) بخشی از این داده‌های شبیه‌سازی آمده است. در جدول (۳-۴) میانگین زمان اجرای هر وظیفه توسط منابع مسئول آمده است و جدول (۴-۴) احتمال تخصیص وظیفه به منابع جهت ایجاد تعادل بارکاری با نرخ ورود $\varphi = 1/12$ را نشان می‌دهد، این مقادیر بر اساس روابط موجود در شبکه‌ی صف با هدف ایجاد تعادل بارکاری محاسبه شده است [۴].

جدول (۳-۴) میانگین زمان سرویس‌دهی هر منبع مسئول برای هر وظیفه‌ی مشخص

$\mu^r(t)$	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
r1	۷				۵	۸		۲	۴	۵		۶
r2	۶	۳	۵	۲			۸	۱			۳	
r3		۲	۷	۲			۸	۱			۲	
r4			۶	۲		۶	۷	۱		۳	۲	
r5			۷	۳	۵		۸	۱	۲		۲	۸

جدول (۴-۴) احتمال تخصیص وظیفه به منابع برای ایجاد تعادل بارکاری با نرخ ورود ۱/۱۲

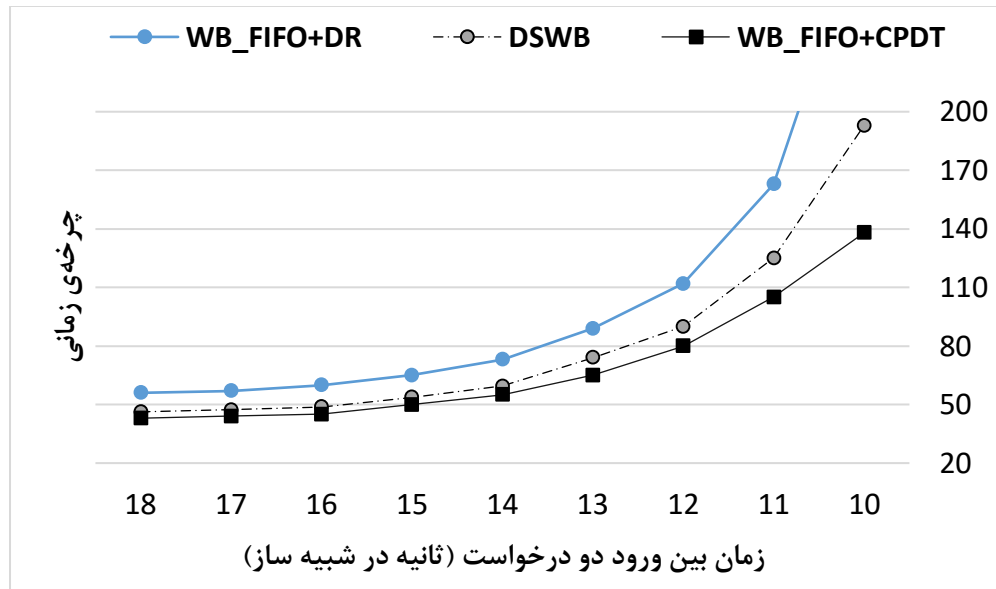
	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
r1	۰/۲۸				۱	۰/۱۹		۰	۰	۰		۱
r2	۰/۷۲	۰	۰/۸۹	۰			۰	۰			۰	
r3		۱	۰/۱۱	۱			۰/۶۵	۱			۰	
r4			۰	۰		۰/۸۱	۰	۰		۱	۰	
r5			۰	۰	۱		۰/۳۵	۰	۱		۱	۰

در این آزمایش از داده‌های شبیه‌سازی Byung-Hyun Ha و همکاران [۴] استفاده می‌شود ولی متأسفانه محاسبه‌ی مقادیر تخمین زمان وابسته‌ی وظایف برای منابع، روی داده‌های معرفی شده توسط Byung-Hyun Ha و همکاران وجود نداشت لذا مقادیر تخمین زمان وابسته‌ی وظایف، مانند پایگاه داده‌ی BPI Challenge 2012 فرض می‌شود. همان‌طور که پیش‌تر ذکر شد، در BPI Challenge 2012، ۱۵/۴ درصد از فعالیت‌ها به‌صورت هم‌زمان انجام شده‌اند. ۱۰/۳ درصد از کل فعالیت‌ها، دارای فاصله‌ی وابستگی ۰/۶۲۲ بودند و ۵/۱ درصد از کل فعالیت‌ها، دارای فاصله‌ی وابستگی ۰/۴۷۱ بودند بنابراین در این آزمایش، مقادیر فاصله‌ی وابستگی به‌صورت تصادفی با توزیعی که در رابطه‌ی (۴-۱۲) آمده است، فرض می‌شود.

$$DF_r(t', t) = \begin{cases} \times & 84.6\% \\ 0.471 & 5.1\% \\ 0.622 & 10.3\% \end{cases} \quad (۴-۱۲)$$

شکل (۴-۶) نتایج مقایسه‌ی روش پیشنهادی WB_FIFO+CPDT را با روش‌های WB_FIFO+DR و DSWB نشان می‌دهد. در این آزمایش مانند آزمایش اول انتخاب جفت فعالیت‌های وابسته و اجرای هم‌زمان آن‌ها باعث شد چرخه‌ی زمانی نسبت به الگوریتم‌های WB_FIFO+DR و DSWD کم‌تر شود. در مقادیر کم‌تر زمان بین دو ورود، طول صف لیست کاری منابع بیش‌تر می‌شود، در نتیجه احتمال انتخاب جفت فعالیت‌های وابسته افزایش می‌یابد و باعث کاهش بیش‌تر چرخه‌ی زمانی می‌گردد.

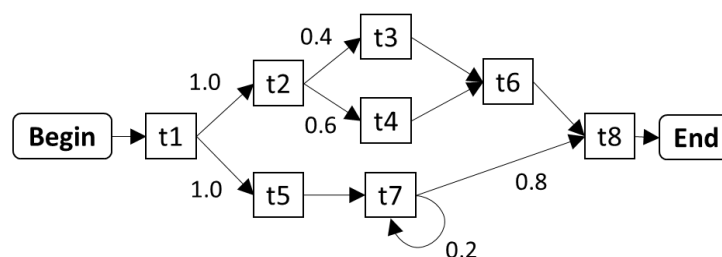
نتیجه‌ی آزمایش دوم کمی با نتایج آزمایش اول نسبت به تغییرات زمان بین دو ورود متفاوت است، این تفاوت در روند افزایش منظم چرخه‌ی زمانی نسبت به کاهش زمان بین دو ورود است. افزایش چرخه‌ی زمانی در شکل (۴-۴) نسبت به شکل (۴-۶) الگوی نامنظم‌تری دارد. دلیل آن به ساختار فرآیند «مالی» در آزمایش اول برمی‌گردد که دارای حلقه‌های بیش‌تری نسبت به فرآیند «تعیین اعتبار» است.



شکل (۴-۶) نتایج آزمایش دوم روی فرآیند تعیین اعتبار

۴-۴-۳ آزمایش سوم: فرآیند تولید

در این آزمایش از فرآیند «تولید» شکل (۴-۷) جهت ارزیابی روش پیشنهادی استفاده می‌شود. این فرآیند شامل ۸ وظیفه و ۴ منبع است و توسط Xie و همکاران [۶] جهت ارزیابی الگوریتم DSWB استفاده شده است. داده‌های مورد استفاده در این آزمایش همان داده‌های Xie و همکاران است که بخشی از این داده‌ها در جدول‌های (۴-۵) و (۴-۶) آمده است.



شکل (۴-۷) فرآیند تولید [۶]

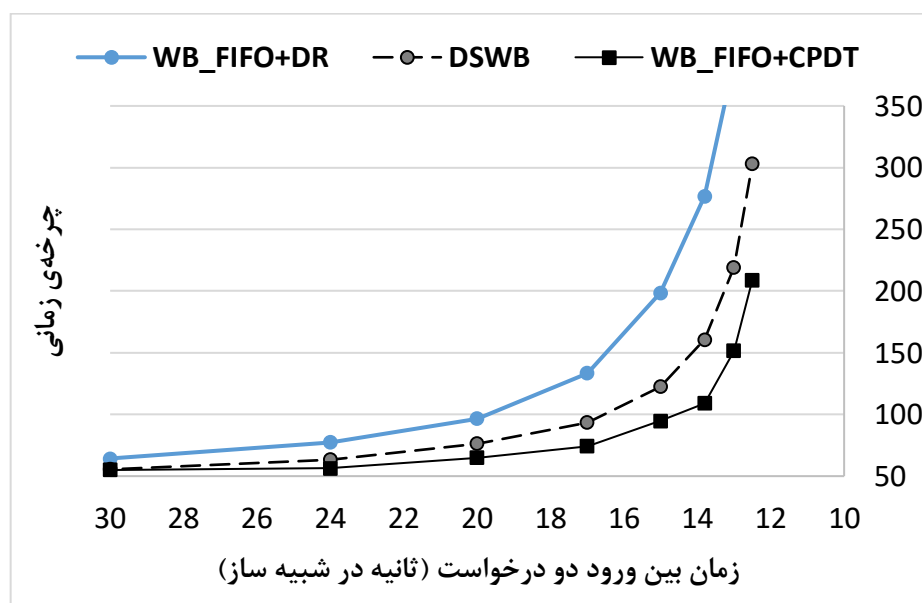
جدول (۴-۵) میانگین زمان سرویس‌دهی هر منبع مسئول برای هر وظیفه‌ی مشخص در فرآیند تولید [۴]

$\mu^r(t)$	t1	t2	t3	t4	t5	t6	t7	t8
r1	۱۴		۸			۱۰	۱۲	
r2				۱۲	۸		۴	۲
r3	۸	۶	۵		۱۲		۸	۴
r4	۸	۱۴		۶		۱۸		۳

جدول (۴-۶) احتمال تخصیص وظیفه به منابع برای ایجاد تعادل بارکاری با نرخ ورود ۱/۱۳

	t1	t2	t3	t4	t5	t6	t7	t8
r1	۰		۰/۴۴۲			۱	۰	
r2				۰	۰/۶۱۷۷		۱	۰/۹۸۰۸
r3	۰	۱	۰/۵۵۸		۰/۳۸۲۳		۰	۰
r4	۱	۰		۱		۰		۰/۰۱۹۲

در این آزمایش مانند آزمایش دوم روش پیشنهادی با دو الگوریتم WB_FIFO+DR و DSWB مقایسه می‌شود و مقادیر فاصله‌ی وابستگی، مطابق با رابطه‌ی (۴-۱۲) فرض می‌شود. شکل (۴-۸) نتایج مقایسه‌ی روش پیشنهادی WB_FIFO+CPDT را با روش‌های WB_FIFO+DR و DSWB نشان می‌دهد. در این آزمایش مانند دو آزمایش قبل، انتخاب جفت فعالیت‌های وابسته و اجرای هم‌زمان آن‌ها باعث شد میانگین چرخه‌ی زمانی نسبت به دو روش دیگر کمتر شود و مانند آزمایش‌های قبل، چرخه‌ی زمانی در الگوریتم پیشنهادی برای مقادیر کمتر زمان بین دو ورود، کاهش بیشتری داشت.



شکل (۴-۸) نتایج آزمایش سوم روی فرآیند تولید

جهت محاسبه‌ی میزان کاهش چرخه‌ی زمانی الگوریتم پیشنهادی (WB_FIFO+CPDT) نسبت به چرخه‌ی زمانی دو الگوریتم WB_FIFO+DR و DSWB میانگین چرخه‌ی زمانی را برای همه‌ی نرخ‌های ورود محاسبه می‌کنیم و میزان کاهش چرخه‌ی زمانی الگوریتم پیشنهادی را نسبت به دو الگوریتم دیگر به دست می‌آوریم. در جدول (۷-۴) میزان کاهش الگوریتم پیشنهادی برای سه آزمایش انجام‌شده گزارش شده است.

جدول (۷-۴) میزان کاهش الگوریتم پیشنهادی CPDT

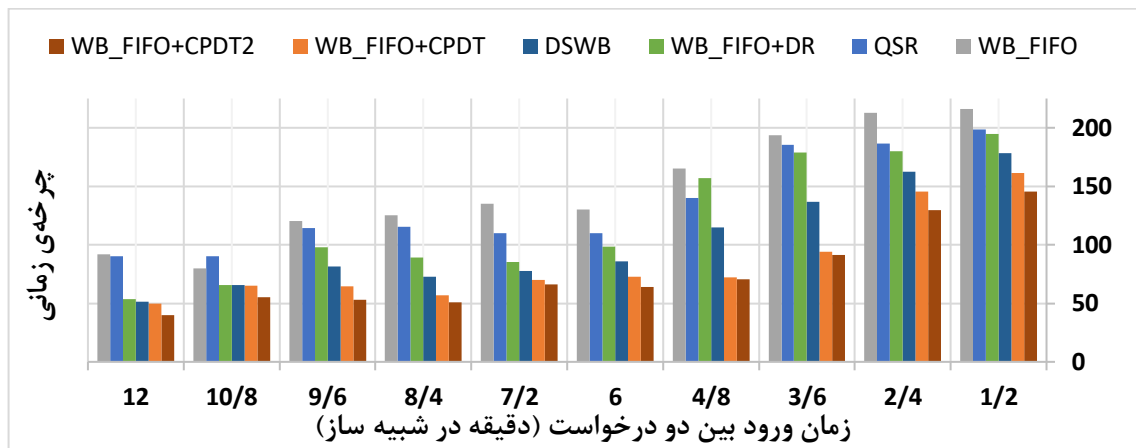
میزان کاهش الگوریتم پیشنهادی نسبت به (درصد)		میانگین چرخه زمانی			آزمایش
DSWB	WB_FIFO+DR	DSWB	WB_FIFO+DR	الگوریتم پیشنهادی	
۱۶/۹	۲۹/۰	۱۰۲/۶۶	۱۲۰/۰۸	۸۵/۲۳	اول
۱۰/۵	۲۷/۸	۶۸/۰۶	۸۴/۳۷	۶۰/۸۷	دوم
۲۲/۶	۳۶/۵	۱۰۶/۳۵	۱۲۹/۶۶	۸۲/۲۶	سوم
۱۶/۶	۳۱/۱	میانگین			

۴-۴-۴ آزمایش چهارم: مقایسه‌ی الگوریتم بهبود یافته

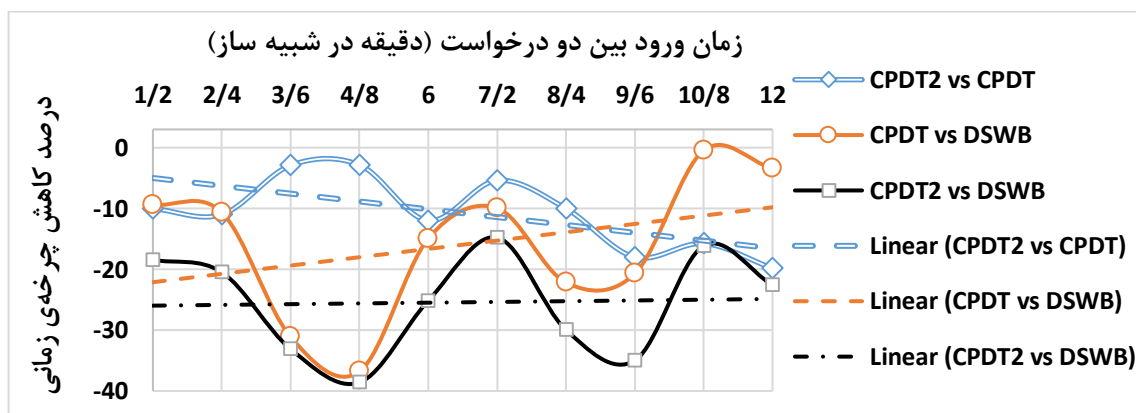
در آزمایش چهارم، مانند آزمایش اول از پایگاه داده BPI Challenge 2012 استفاده می‌کنیم و مراحل آزمایش را مانند آزمایش اول تکرار می‌کنیم با این تفاوت که این بار در لیست الگوریتم‌های مورد مقایسه الگوریتم بهبودیافته‌ی CPDT2 را در نظر می‌گیریم. نتایج نشان داده شده در شکل (۹-۴) نشان می‌دهد الگوریتم CPDT2 چرخه‌ی زمانی درخواست‌ها را نسبت به الگوریتم تکمیلی CPDT کاهش بیش‌تری داده است و این نتیجه نشان می‌دهد در الگوریتم CPDT2 احتمال جابجایی وظایف وابسته افزایش یافته است و این افزایش اثر مطلوبی در کاهش چرخه‌ی زمانی گذاشت.

در شکل (۱۰-۴) میزان کاهش چرخه‌ی زمانی سه الگوریتم DSWB، WB_FIFO+CPDT و WB_FIFO+CPDT2 (WB_FIFO) الگوریتم پایه تخصیص به همراه الگوریتم تکمیلی CPDT2) دو به دو مقایسه‌ی شده‌اند همان‌طور که در روند خطی درصد کاهش چرخه‌ی زمانی الگوریتم WB_FIFO+CPDT نسبت DSWB واضح است. با افزایش نرخ درخواست ورودی احتمال جابجایی وظایف وابسته کاهش می‌یابد در نتیجه درصد کاهش چرخه‌ی زمانی کم‌تر می‌شود. درحالی‌که روند خطی درصد کاهش چرخه‌ی زمانی در الگوریتم WB_FIFO+CPDT2 نسبت به DSWB برای مقادیر

مختلف نرخ درخواست ورودی تغییر چندانی ندارد و یکنواخت است و به طور متوسط $25/4$ درصد کاهش را نسبت به DSWB نشان می‌دهد. به بیان دیگر الگوریتم WB_FIFO+CPDT نتیجه‌ی بهتری را برای مقادیر بیش‌تر نرخ ورود (مقادیر کم‌تر زمان بین دو ورود) درخواست نشان می‌دهد در حالی که الگوریتم WB_FIFO+CPDT2 این بهبود (کاهش چرخه‌ی زمانی) را برای بازه‌ی بزرگ‌تری از مقادیر نشان می‌دهد

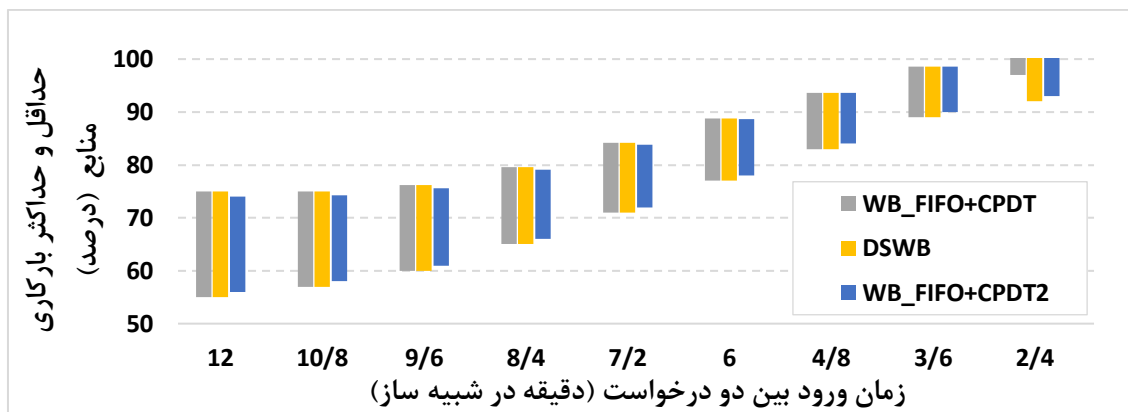


شکل (۹-۴) مقایسه‌ی الگوریتم تکمیلی بهبود یافته‌ی CPDT2 با سایر الگوریتم‌ها



شکل (۱۰-۴) درصد کاهش چرخه‌ی زمانی الگوریتم CPDT2 نسبت به CPDT و الگوریتم CPDT2 نسبت به DSWB و الگوریتم DSWB نسبت به CPDT

در شکل (۱۱-۴) مقدار حداقل و حداکثر بارکاری منابع بعد از پایدار شدن بارکاری سیستم برای سه الگوریتم DSWB, WB_FIFO+CPDT و WB_FIFO+CPDT2 را نشان می‌دهد. همان‌طور که در شکل مشخص است جابجایی هوشمندانه وظایف وابسته در الگوریتم WB_FIFO+CPDT2 نه تنها باعث کاهش چرخه‌ی زمانی شده است بلکه بارکاری منابع را متعادل‌تر کرده است در حالی که در الگوریتم WB_FIFO+CPDT فقط چرخه‌ی زمانی نسبت الگوریتم DSWB بهبود یافته است.



شکل (۴-۱۱) درصد حداقل و حداکثر بارکاری منابع برای دو الگوریتم پیشنهادی (CPDT و CPDT2) در مقایسه با الگوریتم DSWB

۴-۵ جمع بندی و نتیجه گیری

بررسی و تحلیل آماری پایگاه داده‌ی واقعی BPI Challenge 2012 نشان داد، نوعی وابستگی در اجرای هم‌زمان فعالیت‌ها وجود دارد که باعث می‌شود اجرای هم‌زمان آن‌ها زمان کم‌تری نسبت به اجرای جداگانه آن‌ها داشته باشد. در ارائه‌ی روش پیشنهادی، از وابستگی موجود بین وظایف که به‌صورت پنهان در سابقه‌ی رویدادها وجود داشت، استفاده گردید. روش‌های پیشنهادی در تخصیص وظایف به منابع، از این وابستگی استفاده می‌کند و علاوه بر ایجاد تعادل در بارکاری منابع، چرخه‌ی زمانی درخواست‌ها را کاهش می‌دهد. جهت ارزیابی روش‌های پیشنهادی، چهار آزمایش روی فرآیندهای متفاوت انجام شد و روش‌های پیشنهادی CPDT و CPDT2 با الگوریتم‌های موجود [۴، ۶] مقایسه گردید و به ترتیب کاهش ۱۶/۶ و ۲۵/۴ درصدی در چرخه‌ی زمانی درخواست‌ها دیده شد.

الگوریتم‌های تکمیلی ارائه شده در این فصل دارای مزیت‌های قابل توجهی است. اولاً، اثر نامطلوبی در تعادل بارکاری منابع ایجاد نمی‌کند. ثانیاً، پایداری بیش‌تری نسبت به افزایش نرخ ورود مشتریان دارد، زیرا با افزایش نرخ ورود احتمال اجرای هم‌زمان وظایف وابسته افزایش می‌یابد (در روش CPDT2) و چرخه‌ی زمانی درخواست‌ها کاهش بیش‌تری خواهد داشت. در پایان، مهم‌ترین مزیتی که می‌توان به آن اشاره کرد، این است که، روش‌های پیشنهادی در این تحقیق، یک روش تکمیلی هستند که بعد از تخصیص اولیه‌ی وظایف، جهت ایجاد تعادل مجدد بارکاری از اجرای هم‌زمان وظایف وابسته بهره می‌برد.

و به سادگی می توان آن را به عنوان یک الگوریتم تکمیلی با سایر الگوریتم ها تخصیص وظیفه، ترکیب کرد.

۴-۶ پیشنهادات

در ارائه ی الگوریتم کاهش بی نظمی جهت دسته بندی لیست کاری منابع از گراف وابستگی با فاصله ی وابستگی بیش تر از $0/5$ استفاده شد درحالی که نیمی از اطلاعات وابستگی وظایف در گراف وابستگی با فاصله ی وابستگی کم تر از $0/5$ است. در ادامه ی تحقیقات انجام شده در این فصل می توان از گراف وابستگی وزن دار استفاده کرد که وزن هر یال بیانگر فاصله ی وابستگی بین وظایف است و برای جابجایی یا تغییر ترتیب واحدهای کاری در هر لیست کاری منابع می توان از بزرگ ترین درخت پوشا روی گراف وابستگی استفاده کرد.

ایده ی اصلی در الگوریتم ها و روش های ارائه شده در این فصل بر پایه ی نوعی وابستگی در اجرای هم زمان وظایف استوار بود درحالی که در سابقه ی رویدادهای ثبت شده از اجرای فرآیندهای سازمانی دانش بسیاری نهفته است که کشف آن ها با الگوریتم های داده کاوی و فرآیند کاوی میسر است که می تواند باعث بهبود عملکرد BPMS در بخش های مختلف مانند طراحی مدل فرآیند، اجرای فرآیند و بهینه ساز شود.

فصل ۵- بهینه‌سازی در تنظیم هم‌روندی چند فرآیند

معرفی مسئله‌ی تنظیم هم‌روندی فرآیندها
ارائه الگوریتم پویا برای تنظیم مسئله‌ی هم‌روندی
حل مسئله‌ی تنظیم هم‌روندی با PSO
مقایسه‌ی پیچیدگی زمانی الگوریتم‌های ارائه شده

دو تا از چالش‌های اصلی در ایجاد تعادل و یکنواختی در بارکاری منابع اثر نامطلوب تغییرات آهنگ ورود و عدم تخصیص کسری وظایف به منابع است که در بخش ۱-۶ به آن‌ها اشاره شد. مدل پیشنهاد شده در فصل ۳ تا حدودی با استفاده از تکنیک یادگیر تقویتی اثر نامطلوب چالش‌ها را در برقراری تعادل و یکنواختی بارکاری کاهش داد اما هنوز امکان کاهش این اثر نامطلوب در محیط‌هایی که منابع در چند فرآیند مشغول فعالیت هستند وجود دارد. متأسفانه در سابقه‌ی تحقیق به مسئله‌ی تعادل بارکاری در محیط‌های چند فرآیندی توجه‌ی چندانی نشده است. در این فصل، دو هدف اصلی رساله، (۱) بهبود تعادل بارکاری منابع و (۲) تثبیت بارکاری هر منبع در یک سطح مشخص را در محیط‌های چند فرآیندی مورد توجه قرار می‌دهیم و «مسئله‌ی تنظیم هم‌روندی در محیط‌های چند فرآیندی» را به‌عنوان یک مسئله‌ی بهینه‌سازی معرفی می‌کنیم. در ادامه‌ی این فصل، چند روش مختلف برای حل این مسئله ارائه می‌شود. حل مسئله‌ی معرفی شده با الگوریتم جستجوی فضای حالت دارای پیچیدگی زمانی نمایی است. جهت افزایش سرعت اجرای آن، یک روش ابتکاری با بهینه‌سازی پویا برای حل آن ارائه می‌دهیم به‌طوری‌که پیچیدگی زمانی آن در مقابل الگوریتم جستجوی فضای حالت به‌صورت قابل توجهی کاهش می‌یابد. علاوه بر آن، برای حل مسئله‌ی تنظیم هم‌روندی به‌عنوان یک مسئله‌ی بهینه‌سازی از الگوریتم تکاملی PSO استفاده می‌کنیم و در انتها برای یک سناریو مشخص، سرعت اجرای سه روش ارائه شده را مورد مقایسه قرار می‌دهیم.

یک BPMS شامل مجموعه‌ای از فرآیندهای سازمانی است که برای توصیف تمام فعالیت‌های موجود در یک سازمان تعریف می‌شود. این فرآیندها با توجه به نوع و عملکرد هر سازمان تعریف می‌شوند و از مجموعه‌ای از منابع انسانی برای اجرای فعالیت‌های تعریف شده در سازمان استفاده می‌شود. عموماً، هر فرآیند دارای یک ورودی درخواست است و درخواست‌های ورودی هر فرآیند با توجه به جامعه درخواست دهنده دارای آهنگ یا نرخ ورود متفاوتی هستند البته اگر تابع توزیع آماری آن‌ها پواسن یا گاما فرض شود. بنابراین در محیط‌های چند فرآیندی حل مسئله‌ی تعادل بارکاری در شرایطی که نرخ ورود متفاوت است، بسیار مشکل می‌شود درحالی‌که این مسئله برای یک فرآیند مشخص با نرخ ورود

ثابت حل شده است [۳, ۴, ۶]. نکته‌ی قابل توجه این است؛ زمان فعال‌سازی فرآیند یا به عبارت دیگر زمان شروع برای رسیدن اولین درخواست توسط مدیران در سازمان‌ها تعیین می‌شود با تعیین زمان مناسب برای فعال‌سازی هر فرآیند در محیط‌های چند فرآیندی می‌توان تعادل بارکاری منابع و سطح بهینه برای بارکاری هر منبع را بهبود داد.

۵-۱ مفاهیم و تعاریف

۵-۱-۱ مجموعه‌ی فرآیندها با منابع مشترک

در هر سازمان مجموعه‌ای از فرآیندهای سازمانی وجود دارد. وظایف تعریف شده در این فرآیندهای به وسیله‌ی منابع انسانی مسئول در سازمان انجام می‌شود. بنابراین مجموعه‌ی $P = \{p | \exists q \in P, p \neq q, p.R \cap q.R \neq \emptyset\}$ به عنوان کلیه‌ی فرآیندهای موجود در یک سازمان با منابع مشترک تعریف می‌شود و مجموعه‌ی $RS = \bigcup_{p \in P} p.R$ به عنوان مجموعه‌ی تمام منابع انسانی موجود در سازمان در نظر گرفته می‌شود.

۵-۱-۲ الگوی بارکاری منابع در طول زمان

اگر به تابع تعریف شده در رابطه‌ی (۳-۱) پارامتر فرآیند (p) اضافه شود، رابطه‌ی (۵-۱) به دست می‌آید. اگر مقادیر تعریف شده در رابطه‌ی (۵-۱) با استفاده از تکنیک‌های شبیه‌سازی محاسبه شود و اگر اجرای i ام شبیه‌ساز با $ld_{r,p}^i(t)$ نشان داده شود و N تعداد دفعات اجرای شبیه‌ساز باشد آنگاه الگوی بارکاری منابع در طول زمان مطابق با رابطه‌ی (۵-۲) حاصل می‌شود.

$$ld_{r,p}(t) = \sum_{w \in W_r} \frac{\xi_{w,r} \cdot \varphi_w(t)}{\mu_{w,r}}, p \in P, w \in p.W, r \in p.R \quad (۵-۱)$$

$$\overline{ld_{r,p}}(t) = \begin{cases} \frac{\sum_{i=1}^N ld_{r,p}^i(t_0)}{N} & , t = t_0 \\ \frac{\sum_{i=1}^N ld_{r,p}^i(t_1)}{N} & t = t_1 \\ \vdots & \vdots \\ \frac{\sum_{i=1}^N ld_{r,p}^i(t_n)}{N} & t = t_n \end{cases} \quad (۵-۲)$$

۵-۲ بهینه‌سازی در تنظیم هم‌روندی فرآیندها

تنظیم هم‌روندی فرآیندها هنگام اجرای آن‌ها طوری باید انجام شود که اگر یک منبع در یک فرآیند بارکاری کم‌تری دارد، در همان لحظه در فرآیندی دیگر دارای بارکاری بیش‌تری نسبت به سایر منابع داشته باشد. تنظیم هم‌روندی بهینه برای دو فرآیند با یک الگوریتم خطی ساده امکان‌پذیر است، ولی تنظیم هم‌روندی برای چند فرآیند یک مسئله پیچیده‌تر است. در این فصل، این مسئله برای چند فرآیند به‌عنوان یک مسئله بهینه‌سازی معرفی می‌شود و برای حل آن، یک الگوریتم پویا ارائه می‌شود. برای تنظیم هم‌روندی فرآیندها باید به فرض‌ها و محدودیت‌های ذیل اشاره کرد.

الف) جابجایی بارکاری یک منبع در فرآیند در راستای محور زمان امکان‌پذیر نیست، در صورت لزوم، می‌توان بارکاری تمام منابع شرکت‌کننده در فرآیند را در راستای زمان جابجا کرد. به عبارت دیگر کل فرآیند در راستای زمان قابل جابجایی است ولی نمی‌توان بارکاری یک منبع مشخص در فرآیند را در راستای زمان جابجا کرد.

ب) برای جابجایی بارکاری تمام منابع در یک فرآیند در راستای محور زمان، فقط امکان ایجاد تأخیر در ابتدا فرآیند (قبل از شروع فرآیند) وجود دارد.

اگر از یک عنصر تأخیر دهنده در ابتدای فرآیند p استفاده شود، اثر آن در الگوی بارکاری منبع r به صورت $\overline{td_{r,p}}(t - \Delta t_p)$ می‌شود که Δt_p میزان تأخیر ایجادشده توسط عنصر تأخیر دهنده در شروع فرآیند p است. طبق محدودیت (الف) برای تمام منابع شرکت‌کننده در فرآیند p مقدار Δt_p باید یکسان باشد. برای ایجاد هم‌روندی بین چند فرآیند کافی است، درخواست‌ها در یک لحظه از زمان به فرآیندها برسند و این باعث می‌شود بارکاری منابع مشترک باهم جمع شود. هدف از حل مسئله‌ی تنظیم هم‌روندی فرآیندها، قرار دادن عناصر تأخیر دهنده در ابتدا فرآیندها است به صورتی که مقادیر Δt_p ها به گونه‌ای تنظیم گردد که انحراف از تعادل بارکاری در منابع (رابطه‌ی (۵-۷)) و تغییرات نسبت به سطح بهینه برای هر منبع (رابطه‌ی (۵-۵))، حداقل شود. به عبارت دیگر، مقادیر Δt_p در رابطه‌ی (۵-۳) طوری تعیین شود تا رابطه‌ی (۵-۸) کمینه شود.

$$total_ld_r(t) = \sum_{p \in P} \overline{ld_{r,p}}(t - \Delta t_p) \quad (3-5)$$

$$Rs = \bigcup_{p \in P} p.R \quad (4-5)$$

$$ULD = \frac{\sum_{r \in Rs} \sum_{i=0}^N (total_ld_r(t_i) - \mu_r)^2}{N|Rs|} \quad (5-5)$$

$$EL(t) = \sum_{r \in Rs} total_ld_r(t) / |Rs| \quad (6-5)$$

$$VLD = \frac{\sum_{r \in Rs} \sum_{i=0}^N (total_ld_r(t_i) - EL(t_i))^2}{N|Rs|} \quad (7-5)$$

$$LBFAT = \alpha ULD + (1 - \alpha) VLD \quad (8-5)$$

در رابطه‌های (3-5) تا (8-5) به ازای هر فرآیند یک مجهول Δt_p وجود دارد که میزان تأخیر ایجادشده در ابتدای فرآیند p را نشان می‌دهد. مقادیر Δt_p ها نقش تنظیم‌کننده را در هم‌روندی فرآیندها دارند. رابطه‌ی (3-5) مقدار مجموع بارکاری منبع r را در اجرای هم‌روند فرآیندهای مجموعه‌ی P محاسبه می‌کند. رابطه‌ی (5-5) میزان یکنواختی بارکاری منابع را در طول زمان محاسبه می‌کند، در صورتی مقدار ULD^1 حداقل می‌شود که بارکاری $total_ld_r(t)$ برای تمام منابع مانند r در طول زمان تغییرات کم‌تری نسبت به مقدار μ_r داشته باشد. در رابطه‌ی (7-5)، VLD^2 جمع مربعات تفاوت بارکاری منابع نسبت به میانگین در طول زمان است که حداقل شدن آن به معنای تعادل بیش‌تر در بارکاری منابع است. در رابطه‌ی (8-5)، $LBFAT^3$ تابع هدف مسئله‌ی تنظیم هم‌روندی است و معیار اندازه‌گیری جهت تعادل بارکاری کلیه‌ی منابع و یکنواختی در بارکاری هر منبع است. مقدار α و $(1 - \alpha)$ به ترتیب به‌عنوان ضریب تأثیر مقادیر ULD و VLD معرفی می‌شوند.

¹ Uniformity of Workload

² Load Balancing Factor Along Time

³ Variance of Workload

۵-۳ حل مسئله‌ی تنظیم هم‌روندی

در این بخش از سه روش جستجوی فضای حالت، بهینه‌سازی پویا و الگوریتم تکاملی PSO، برای حل مسئله‌ی تنظیم هم‌روندی استفاده می‌کنیم و این روش‌ها را از نظر پیچیدگی زمانی حل مسئله مورد مقایسه قرار می‌دهیم.

۵-۳-۱ جستجوی فضای حالت

اولین و ساده‌ترین روش برای حل مسئله‌ی تنظیم هم‌روندی جستجو در فضای حالت مسئله است. با بررسی تمام حالت‌های موجود مسئله و محاسبه مقدار LBFAT برای هر حالت، با توجه به رابطه‌های (۵-۳) تا (۵-۸) می‌توان به‌سادگی به این نتیجه رسید که پیچیدگی زمانی محاسبه‌ی مقدار LBFAT، در بدترین حالت $O(|Rs|(|P| + 2N + 1)) = O(|Rs|(|P| + 2N + 1))$ می‌شود. که در آن $|P|$ تعداد فرایندهای هم‌روند شده است. N تعداد رویدادهای اتفاق افتاده در طول اجرای فرآیند است که در هر رویداد بارکاری برخی منابع تغییر کرده است و $|Rs|$ تعداد منابع موجود در سازمان است. برای تنظیم مسئله‌ی هم‌روندی چند فرآیند در جستجوی فضای حالت تمام حالت‌هایی که می‌توان Δt_p ها مقداردهی کرد را شمارش می‌کنیم. با محاسبه‌ی جایگشت‌های مختلف فرآیندها و فاصله زمانی هر دو فرآیند پشت سر هم می‌توان به این نتیجه رسید که کل حالت‌های مسئله برابر با $|P|! \binom{N}{|P|}$ می‌شود. با جایگذاری مقدار ترکیب $|P|$ از N این تعداد به صورت $\frac{N!}{(N-|P|)!}$ ساده می‌شود. بنابراین با اعمال پیچیدگی زمانی محاسبه‌ی LBFAT در رابطه‌ی (۵-۸) پیچیدگی زمانی الگوریتم جستجوی فضای حالت برای حل مسئله‌ی پیشنهاد شده به صورت $\theta(\frac{N!}{(N-|P|)!} |Rs|(|P| + 2N + 1))$ می‌شود. با بسط $N!$ و $(N - |P|)!$ و ساده‌سازی عبارات بسط داده شده، می‌توان به پیچیدگی زمانی $O(N^{|P|} |Rs|(|P| + 2N + 1))$ رسید.

۵-۳-۲ بهینه‌سازی پویا

تکنیک برنامه‌نویسی پویا یا بهینه‌سازی پویا، روش شناخته شده‌ای است که در ریاضیات، علوم کامپیوتر و علوم مدیریت استفاده می‌شود. با اینکه اولین بار این تکنیک توسط ریچارد بلمن در سال ۱۹۴۰ معرفی و استفاده شد [۴۰]، هنوز مورد توجه محققان و نویسندگان قرار می‌گیرد [۴۱-۴۴].

بهینه‌سازی پویا یکی از روش‌های بهینه‌سازی به روش تقسیم حل است که به صورت بازگشتی (پایین به بالا) جواب بهینه‌ی مسئله را پیدا می‌کند. اگر $P = \{p_1, p_2\}$ باشد با دو بار پیمایش روی بازه‌ی زمانی $[t_0, t_N]$ مقدار بهینه برای LBFAT حاصل می‌شود. الگوریتم WorkloadMerge نحوه‌ی ادغام بارکاری منابع برای حالتی که $|P| \geq 2$ است و مقدار کمینه‌ی LBFAT حاصل می‌شود را نشان می‌دهد.

اگر فرض کنیم برای مجموعه‌ای از فرآیندهای مانند A مسئله‌ی تنظیم هم‌روندی به صورت بهینه حل شده است و مقدار Δt_p برای $\forall p \in A$ به صورت بهینه به دست آمده است و در بردار D قرار گرفته است، الگوریتم WorkloadMerge یک ادغام بهینه (حداقل مقدار برای LBFAT در رابطه‌ی (۵-۸)) از بارکاری منابع در فرآیند q با بارکاری منابع در فرآیندهای مجموعه‌ی A ایجاد می‌کند. اگر $A = \{p\}$ و $D = [0]$ باشد. الگوریتم WorkloadMerge یک ادغام بهینه بین بارکاری منابع در فرآیندهای p و q ایجاد می‌کند. در این حالت کافی است تابع به صورت $WorkloadMerge(\{p\}, [0], q)$ فراخوانی شود. در این حالت، در دو حلقه‌ی for یکبار برای فرآیند p تأخیر با مقادیر m ایجاد می‌شود و یکبار برای فرآیند q تأخیر با مقادیر n ایجاد می‌شود و کم‌ترین مقدار LBFAT محاسبه می‌گردد.

پیچیدگی زمانی الگوریتم پیشنهاد شده‌ی $WorkloadMerge$ به سادگی قابل محاسبه است و برابر با $O(|Rs|(|P| + 2N + 1) \times 2N)$ می‌شود که در آن $|P|$ تعداد فرآیند، $|Rs|$ تعداد منابع مشترک در فرآیندها و $N = \max_{p \in P}(last(p))$ حداکثر تعداد رویدادهای رخ در فرآیند است که به واسطه آن بارکاری حداقل یک منبع تغییر کرده است.

Algorithm WorkloadMerge ()**Input:** $\langle A: \text{list of Process}, D_A: \text{delay vector of } A, q: \text{process} \rangle$ **Output:** $\langle \text{minLBFAT}, \text{newD}, ld_r^{A \cup \{q\}} \rangle$

```

{  $P \leftarrow A.push(q)$ ;  $\text{MinWLFAT} = \infty$ ;
for ( $m = 0; m < \max_{p \in A}(\text{last}(p)); m++$ ) {
     $D \leftarrow D_A.push(m)$ ;
     $ld_r(t) = \sum_{p_i \in P} ld_{r,p_i}(t - D(i)); \forall t \in \{t_0, t_1, t_2, \dots, t_{\max(\text{last}(p), \text{last}(q))}\}$ 
     $Rs = \bigcup_{p \in P} p.R$ 
     $ULD = \frac{\sum_{r \in Rs} \sum_{i=0}^N (\text{total}_{ld_r}(t_i) - \sigma_r)^2}{N|Rs|}$ 
     $EL(t) = \sum_{r \in Rs} \text{total}_{ld_r}(t) / |Rs|$ 
     $VLD = \frac{\sum_{r \in Rs} \sum_{i=0}^N (\text{total}_{ld_r}(t_i) - EL(t_i))^2}{N|Rs|}$ 
     $\text{LBFAT} \leftarrow (ULD + VLD) / 2$ 
    if ( $\text{LBFAT} < \text{minLBFAT}$ ) {
         $\text{minLBFAT} \leftarrow \text{LBFAT}$ ;
         $\text{newD} \leftarrow D$ ;
         $ld_r^{A \cup \{q\}}(t) = ld_r(t)$ ;
    } //if
} //for
for ( $n = 1; n < \text{last}(q); n++$ ) {
    foreach ( $d_i \in D_A$ ) do  $d_i \leftarrow d_i + n$ ;
     $D \leftarrow D_A.push(0)$ ;
     $ld_r(t) = \sum_{p_i \in P} ld_{r,p_i}(t - D(i)); \forall t \in \{t_0, t_1, t_2, \dots, t_{\max(\text{last}(p), \text{last}(q))}\}$ 
     $Rs = \bigcup_{p \in P} p.R$ 
     $ULD = \frac{\sum_{r \in Rs} \sum_{i=0}^N (\text{total}_{ld_r}(t_i) - \sigma_r)^2}{N|Rs|}$ 
     $EL(t) = \sum_{r \in Rs} \text{total}_{ld_r}(t) / |Rs|$ 
     $VLD = \frac{\sum_{r \in Rs} \sum_{i=0}^N (\text{total}_{ld_r}(t_i) - EL(t_i))^2}{N|Rs|}$ 
     $\text{LBFAT} \leftarrow (ULD + VLD) / 2$ 
    if ( $\text{LBFAT} < \text{minLBFAT}$ ) {
         $\text{minLBFAT} \leftarrow \text{LBFAT}$ ;
         $\text{newD} \leftarrow D$ ;
         $ld_r^{A \cup \{q\}}(t) = ld_r(t)$ ;
    } //if
} //for
return  $\langle \text{minLBFAT}, \text{newD}, ld_r^{A \cup \{q\}}(t) \rangle$ 
} // WorkloadMerge

```

برای حل مسئله‌ی هم‌روندی برای فرآیندهای بیش‌تر از دو فرآیند، اگر P مجموعه‌ی فرآیندهای ورودی مسئله باشد؛ کافی است ابتدا مسئله برای تمام زیر مجموعه‌های دو عضوی از P با استفاده از الگوریتم $WorkloadMerge$ حل شود. سپس تمام زیر مجموعه‌های سه عضوی P را به‌صورت بهینه ادغام می‌کنیم. به‌عنوان مثال برای حل یک زیرمجموعه سه عضوی مانند $\{p_1, p_2, p_3\}$ سه بار تابع ادغام $WorkloadMerge(\{p_1, p_2\}, [\Delta t_{p_1}, \Delta t_{p_2}], p_3)$ برای ادغام p_3 با $\{p_1, p_2\}$ و $WorkloadMerge(\{p_2, p_3\}, [\Delta t_{p_2}, \Delta t_{p_3}], p_1)$ برای ادغام p_1 با $\{p_2, p_3\}$ و به همین صورت برای ادغام p_2 با $\{p_1, p_3\}$ تابع به‌صورت $WorkloadMerge(\{p_1, p_3\}, [\Delta t_{p_1}, \Delta t_{p_3}], p_2)$ فراخوانی می‌شود و خروجی کمینه این سه فرآخوانی به‌عنوان ادغام بهینه برای مجموعه $\{p_1, p_2, p_3\}$ در نظر گرفته می‌شود. به همین صورت زیر مجموعه‌های چهار عضوی حل می‌شود و از مقادیر حاصل شده در مراحل قبلی استفاده می‌شود و این مراحل تا رسیدن به جواب نهایی که مجموعه P است ادامه می‌یابد. حال اگر S_P^k مجموعه‌ی تمام زیر مجموعه‌های k تایی از مجموعه فرآیندهای P باشد و اگر تابع $WorkloadMerge$ به‌اختصار به‌صورت WM نشان داده شود. برای دسترسی به خروجی تابع $WorkloadMerge$ که به‌صورت سه‌تایی $\langle w, D, ld \rangle$ است از نمادهای WM_w ، WM_D و WM_{ld} استفاده شده است؛ رابطه‌ی بازگشتی (۹-۵) برای حل مسئله‌ی تنظیم هم‌روندی به‌صورت پویا ارائه می‌شود.

$$\forall A \in S_P^2: \langle w_A, D_A, ld_r^A \rangle \leftarrow WM(\langle \{p\}, [0], q \rangle), p, q \in A, p \neq q$$

$$\forall B \in S_P^k: \begin{cases} w_B \leftarrow \min_{q \in B} (WM_w(\langle B - \{q\}, D_{B-\{q\}}, ld_r^{B-\{q\}} \rangle)) \\ p \leftarrow \text{Arg min}_{q \in B} (WM_w(\langle B - \{q\}, D_{B-\{q\}}, ld_r^{B-\{q\}} \rangle)) \\ D_B \leftarrow WM_D(\langle B - \{p\}, D_{B-\{p\}}, ld_r^{B-\{p\}} \rangle) \\ ld_r^B \leftarrow WM_{ld}(\langle B - \{p\}, D_{B-\{p\}}, ld_r^{B-\{p\}} \rangle) \end{cases} \quad (9-5)$$

در رابطه‌ی (۹-۵) ابتدا مسئله‌ی تنظیم هم‌روندی برای تمام زیرمجموعه‌های دو عضوی P حل می‌شود و سه‌تایی‌های $\langle w_A, D_A, ld_r^A \rangle$ برای تمام مجموعه‌های دو عضوی به دست می‌آید سپس

مسئله‌ی تنظیم هم‌روندی برای زیر مجموعه‌های سه عضوی حل می‌شود. در حل مسائل با یک عضو بیش‌تر از مقادیر حاصل شده از زیر مجموعه‌های کوچک‌تر استفاده می‌شود و مسائل با یک عضو کم‌تر به کمک تابع *WorkloadMerge* با عضو جدید ادغام شده و مقدار کمینه‌ی آن برای مرحله‌ی بعد استفاده می‌شود.

پیچیدگی زمان الگوریتم پویای ارائه شده در این بخش با توجه به رابطه‌ی (۵-۹) قابل محاسبه است. تعداد دفعاتی که تابع *WorkloadMerge* فراخوانی می‌شود برابر است با $\binom{|P|}{2} + 3\binom{|P|}{3} + \dots + |P|\binom{|P|}{|P|}$ است. درنتیجه در بدترین حالت تعداد دفعات فراخوانی $O(|P|2^{|P|})$ می‌شود و با اعمال پیچیدگی زمانی تابع *WorkloadMerge* پیچیدگی زمانی الگوریتم پویای ارائه شده در بدترین حالت $O(|P|2^{|P|} \times |Rs|(|P| + 2N + 1) \times 2N)$ می‌شود بنابراین در حل مسئله‌ی تنظیم هم‌روندی، الگوریتم پویای ارائه شده نسبت به الگوریتم جستجوی فضای حالت دارای پیچیدگی زمانی کم‌تری است. در هر دو الگوریتم پیچیدگی زمانی نسبت به $|P|$ نمایی است با این تفاوت که در نسخه‌ی پویا پایه از N به ۲ کاهش یافته است و در الگوریتم جستجوی فضای حالت پیچیدگی زمانی نسبت به N یک چند جمله‌ای از درجه‌ی ۱ $|P| + 1$ است درحالی‌که در نسخه‌ی پویا یک چند جمله‌ای از درجه ۲ است. از آنجایی که مقدار N بزرگ‌ترین مؤلفه در مسئله‌ی تنظیم هم‌روندی است و این مؤلفه نسبت به $|P|$ بسیار بزرگ است. تغییرات حاصل شده در پیچیدگی زمانی در نسخه‌ی پویا تأثیر چشم‌گیری در سرعت اجرای الگوریتم دارد.

۵-۳-۳ الگوریتم تکاملی PSO

الگوریتم‌های ارائه شده در بخش‌های قبل مسئله‌ی تنظیم هم‌روندی فرآیندهای کسب‌وکار را به‌صورت بهینه حل می‌کنند و علی‌رغم کاهش چشم‌گیر در پیچیدگی زمانی در نسخه پویا، هنوز هر دو الگوریتم نسبت به مؤلفه‌ی $|P|$ نمایی هستند. الگوریتم‌های زیادی وجود دارد که به‌صورت تقریبی مسائل بهینه‌سازی را حل می‌کنند و پیچیدگی زمانی آن‌ها نسبت به حل بهینه کم‌تر است. مانند الگوریتم

ژنتیک، PSO، ICA و کلونی مورچه‌ها، تپه نوردی و غیره. در این بخش برای حل مسئله‌ی تنظیم هم‌روندی از الگوریتم PSO استفاده می‌شود.

PSO یک الگوریتم تکاملی برای حل مسائل بهینه‌سازی است [۴۵]. که در بسیار از تحقیقات از آن استفاده شده است [۴۶-۴۸]. این الگوریتم با استفاده از یک جمعیت اولیه از ذرات در یک فضای d بعدی با مقادیر اولیه تصادفی برای موقعیت و سرعت این ذرات، با یک الگوریتم تکراری، مسائل بهینه‌سازی را به‌صورت تقریبی حل می‌کند. در این الگوریتم یک تابع هزینه برای هر ذره تعریف می‌شود، ورودی این تابع مقدار موقعیت ذره $X(c)$ در فضای d بعدی است. برای مقداردهی سرعت V^{new} و موقعیت جدید X^{new} هر ذره c در هر تکرار از دو مقدار بهترین (کم هزینه‌ترین) جواب عمومی ذرات و بهترین جواب محلی ذره استفاده می‌شود. در رابطه‌های (۵-۱۰) و (۵-۱۱) روش به‌روزرسانی سرعت و موقعیت هر ذره آمده است.

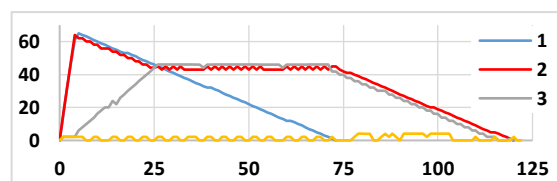
$$V^{new}(c) = c_1 r_1 (Global - X^{cur}(c)) + c_1 r_1 (Local(c) - X^{cur}(c)) + w V^{cur}(c) \quad (۵-۱۰)$$

$$X^{new}(c) = X^{cur}(c) + V^{new}(c) \quad (۵-۱۱)$$

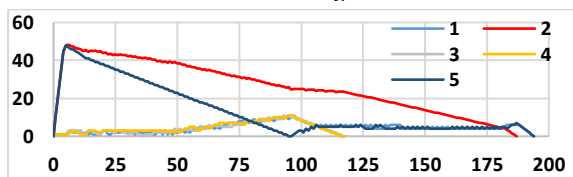
مقادیر r_1 و r_2 دو عدد تصادفی با توزیع یکنواخت بین صفر و یک است، مقادیر c_1 و c_2 به ترتیب ضرایب تأثیر جواب عمومی و محلی ذرات است، مقدار w ضریب اینرسی ذره است. در آزمایش انجام شده در این بخش مقادیر c_1 ، c_2 و w به ترتیب $۰/۷$ ، $۰/۳$ و $۰/۶$ در نظر گرفته شده است. برای حداقل کردن مقدار رابطه‌ی (۵-۸) از الگوریتم تکاملی PSO استفاده می‌شود. جهت بهینه‌سازی مسئله‌ی تنظیم هم‌روندی چند فرآیند، کافی است، مقدار رابطه‌ی (۵-۸) به‌عنوان تابع هدف، کمینه شود، در این بهینه‌سازی، بردار Δt به‌عنوان بردار موقعیت ذرات در نظر گرفته می‌شود، که بُعد این بردار برابر با تعداد فرآیندهایی است که قرار است هم‌روند شوند.

۴-۵ آزمایش و نتایج

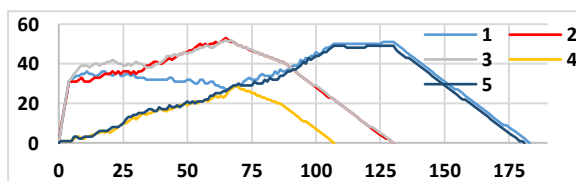
در این بخش، برای مقایسه‌ی سرعت اجرای سه الگوریتم ارائه شده در بخش ۳-۵ یک آزمایش با پنج فرآیند کسب‌وکار با پنج منبع مشترک انجام می‌شود. در شکل (۱-۵) نمودار YAWL [۱۴] پنج فرآیندها کسب‌وکار با نمودارهای بارکاری منابع مسئول در این فرآیندها آمده است. در شکل (۱-۵) الگوهای بارکاری منابع برای هر یک از فرآیندهای به‌وسیله شبیه‌سازی به دست آمده است. شبیه‌سازی توزیع آماری درخواست‌های ورودی پواسن فرض شده است و نرخ ورود روی مقدار ۰/۱ تنظیم شده است و مجموع کل درخواست‌های ورودی به ۲۰۰ درخواست محدود شده است. برای هر فرآیند اجرای شبیه‌ساز ۵ بار تکرار شد و الگوی بارکاری طبق رابطه‌ی (۲-۵) به دست آمد.



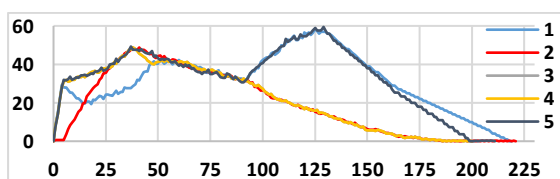
$R = \{1,2,3,4\}, t_N = 221$



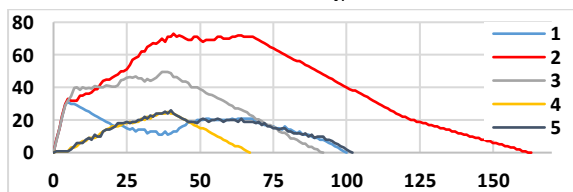
$R = \{1,2,3,4,5\}, t_N = 194$



$R = \{1,2,3,4,5\}, t_N = 183$

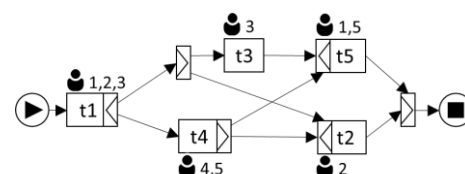
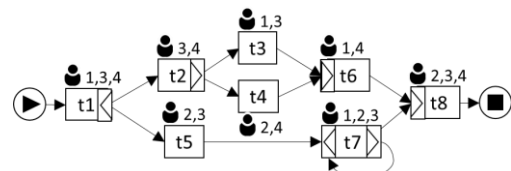
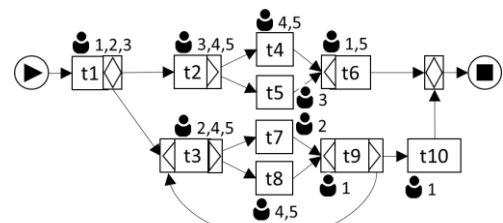
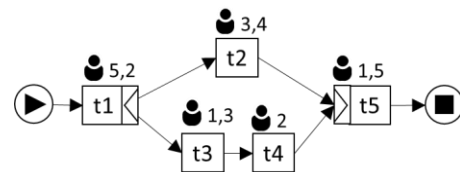
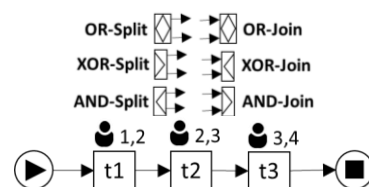


$R = \{1,2,3,4\}, t_N = 221$

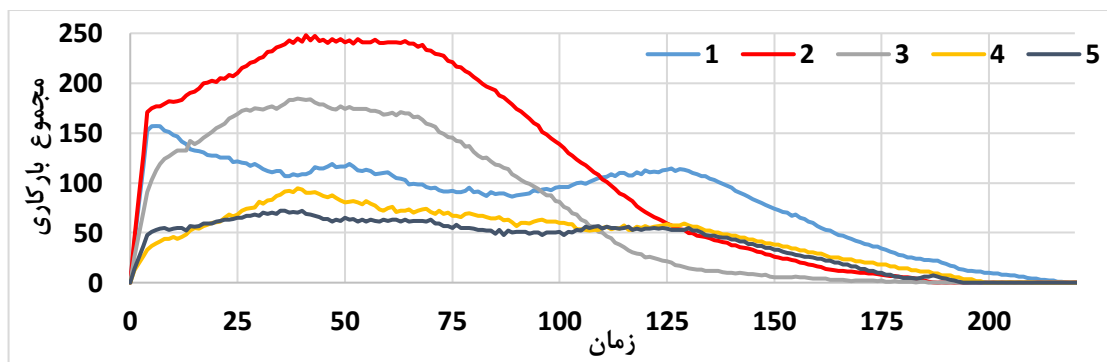


$R = \{1,2,3,4,5\}, t_N = 163$

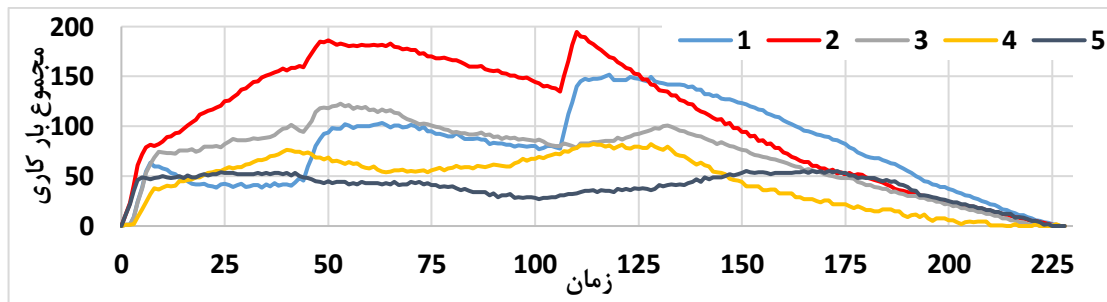
شکل (۱-۵) پنج فرآیند کسب‌وکار و نمودار بارکاری آن‌ها



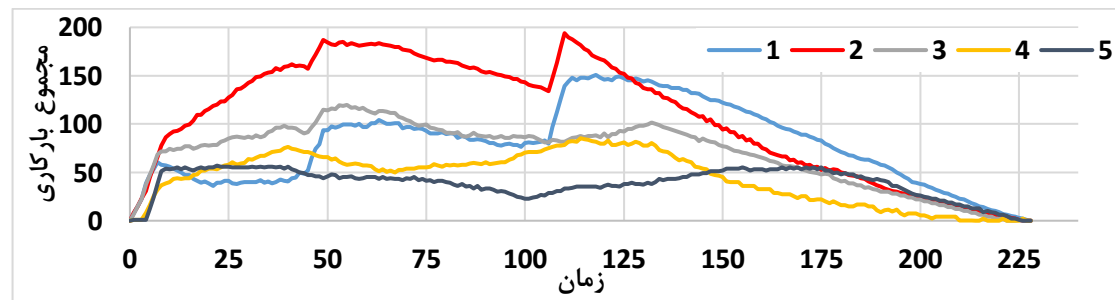
در شکل (۱-۵) علاوه بر الگوی بارکاری منابع مدل فرآیند کسب و کار با زبان مدل سازی YAWL رسم شده است. در هر مدل روی هر وظیفه شماره‌ای منابع مسئول مشخص شده است. اگر مقادیر Δt_p ها را به عنوان بردار تأخیر D (تنظیم کننده هم‌روندی) دارای مقدار $D = [0,0,0,0,0]$ فرض شود الگوی بارکاری جمع شده منابع به صورت شکل (۲-۵) الف می‌شود و مقدار LBFAT آن برابر با مقدار ۸۸۳۵۷ می‌گردد درحالی‌که با حل مسئله‌ی تنظیم هم‌روندی با یکی از روش‌های بخش ۳-۵ به صورت بهینه، الگوی شکل (۲-۵) ب) یا به صورت تقریبی (با الگوریتم PSO) الگوی شکل (۲-۵) ج) حاصل می‌شود که به ترتیب مقادیر ۶۱۹۸۹ و ۶۲۰۷۳ برای LBFAT به دست آمده است.



(الف) قبل از بهینه‌سازی با $D = [0,0,0,0,0]$ و $LBFAT = 88357$



(ب) جواب بهینه با $D = [106.24, 0, 44.28, 2.48, 3.42]$ و $LBFAT = 61989$

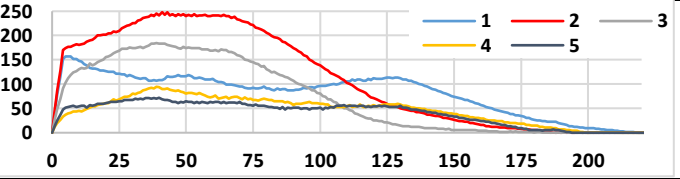
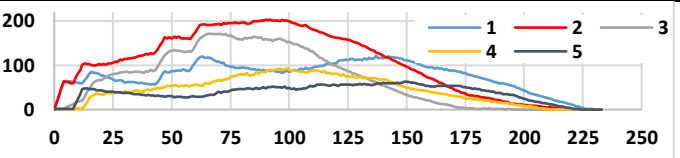
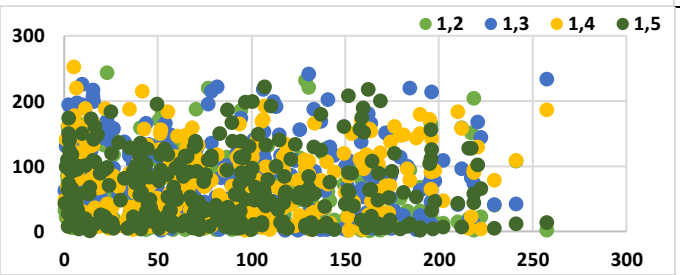
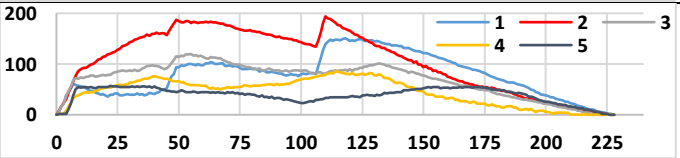
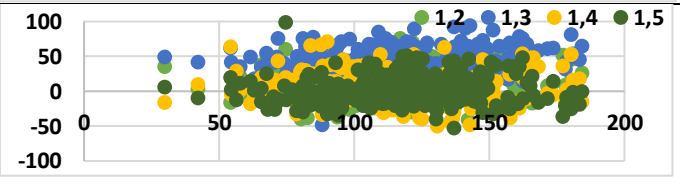
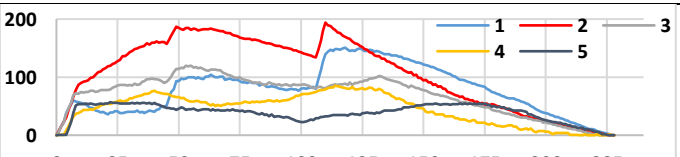
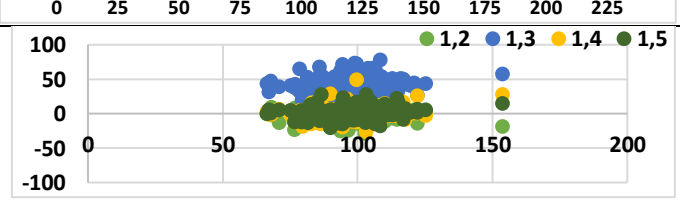
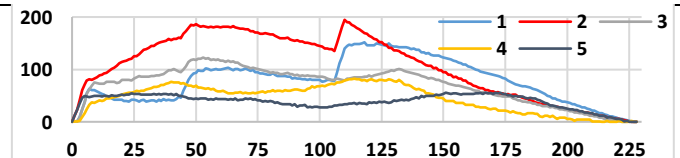
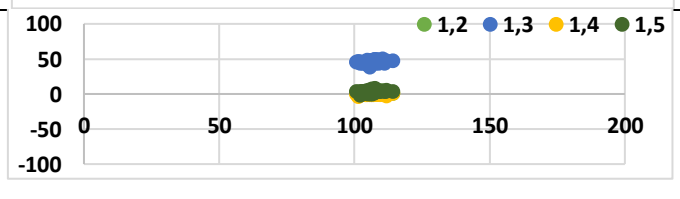


(ج) جواب تقریبی با $D = [106, 4, 45, 0, 3]$ و $LBFAT = 62073$

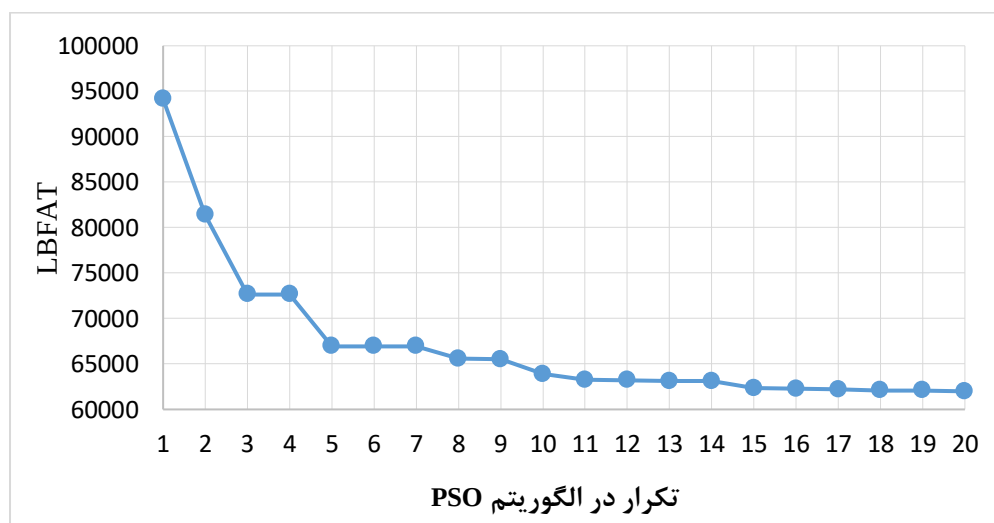
شکل (۲-۵) نتیجه‌ی تنظیم هم‌روندی فرآیندها با استفاده از الگوریتم‌های پیشنهاد شده

جدول (۵-۱) کاهش LBFAT و تعیین پارامتر Δt در بارکاری فرآیندهای هم‌روند با استفاده از الگوریتم

بهینه‌سازی PSO

قبل از بهینه‌سازی LBFAT= 88357.07 $\Delta t = [0,0,0,0,0]$	بارکاری	
بعد از تکرار اول PSO LBFAT= 80389.70 $\Delta t = [0,7.6,43,57.2,11.7]$ از آنجایی که امکان نمایش فضای ۵ بُعدی وجود نداشت هر نقطه در نمودار توزیع ذرات بیانگر مقدار بُعدهای بیش‌تر از یک با بُعد اول است. به‌عنوان مثال داده‌های 1,2 مختصات بُعد دوم را نسبت به بُعد اول نشان می‌دهد.	بارکاری	 توزیع ذرات در PSO 
بعد از تکرار پنجم PSO LBFAT= 66448.68 $\Delta t = [110,0,51.9,2.3,3.77]$	بارکاری	 توزیع ذرات در PSO 
بعد از تکرار دهم PSO LBFAT= 62252.24 $\Delta t = [106.52,3,45,0,3.2]$	بارکاری	 توزیع ذرات در PSO 
بعد از تکرار بیستم PSO LBFAT= 61989.494 $\Delta t = [106.2,0,44.2,2.4,3.4]$	بارکاری	 توزیع ذرات در PSO 

جدول (۵-۱) مراحل اجرای الگوریتم PSO را نشان می‌دهد. همان‌طور که در جدول مشخص است بارکاری هم‌روندی فرآیندها، قبل از اجرای الگوریتم بهینه‌سازی دارای مقدار ۸۸۳۵۷ برای LBFAT است. این مقدار بعد از بهینه‌سازی به مقدار ۶۱۹۸۹ کاهش یافت (شکل (۵-۳)). این کاهش نشان می‌دهد پارامتر $\Delta t = [106.24, 0, 44.28, 2.48, 3.42]$ به‌عنوان یک تنظیم‌کننده توانسته است، علاوه بر ایجاد تعادل در بارکاری منابع، در بارکاری هر منبع یکنواختی بهتری ایجاد کرده است. تنظیم مقدار Δt در هم‌روندی فرآیندها باعث می‌شود بارکاری منابع در راستای محور زمان به‌گونه‌ای جابجا شود تا هدف بهینه‌سازی مسئله که تعادل بارکاری و یکنواختی بارکاری منابع است تأمین شود.



شکل (۵-۳) مقادیر LBFAT فرآیندهای هم‌روند در تکرارهای الگوریتم بهینه‌ساز PSO

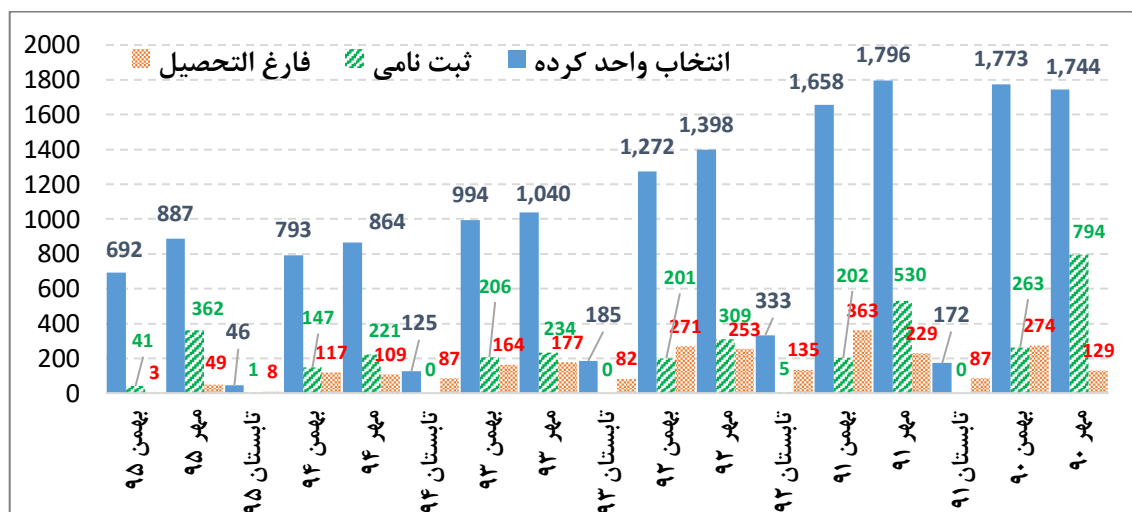
برای مقایسه‌ی سرعت اجرای الگوریتم‌های از یک دستگاه با پردازنده‌ی Core i7 با فرکانس 4GHz با 16Gbyte حافظه‌ی اصلی استفاده شد و نتایج این مقایسه در جدول (۵-۲) گزارش شد.

جدول (۵-۲) پیچیدگی زمانی و زمان اجرای الگوریتم‌های ارائه شده ($|P| = 5, |Rs| = 5, N = 211$)

الگوریتم	پیچیدگی زمانی	زمان اجرا	نتیجه
جستجوی فضای حالت	$O(N^{ P } Rs (P + 2N + 1))$	۵/۸ سال (تخمینی)	بهینه
بهینه‌سازی پویا	$O(P 2^{ P } \times Rs (P + 2N + 1) \times 2N)$	۳۷ ساعت	بهینه
PSO	$O(Rs (P + 2N + 1) \times mp)$	۱:۲۶ دقیقه	تقریبی
PSO	$O(Rs (P + 2N + 1) \times mp)$	۳:۰۲ دقیقه	بهینه

۵-۵ مطالعه موردی

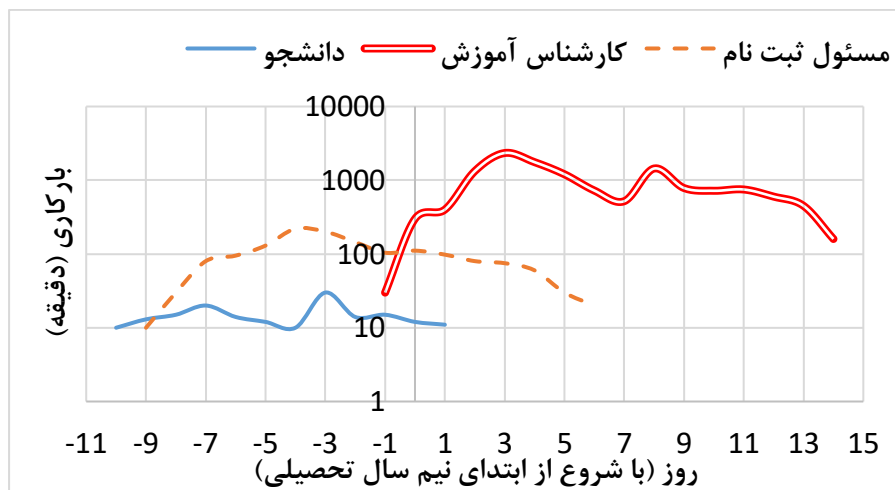
در این بخش از پایگاه داده موسسه‌ی غیرانتفاعی آموزش عالی میرداماد گرگان طی سال‌های ۱۳۹۰ الی ۱۳۹۶ استفاده شده است. شکل (۴-۵) تعداد دانشجویان ثبت نامی، فارغ‌التحصیل و دانشجویان انتخاب واحد کرده را طی این ۶ سال به تفکیک نیم‌سال تحصیلی نشان می‌دهد.



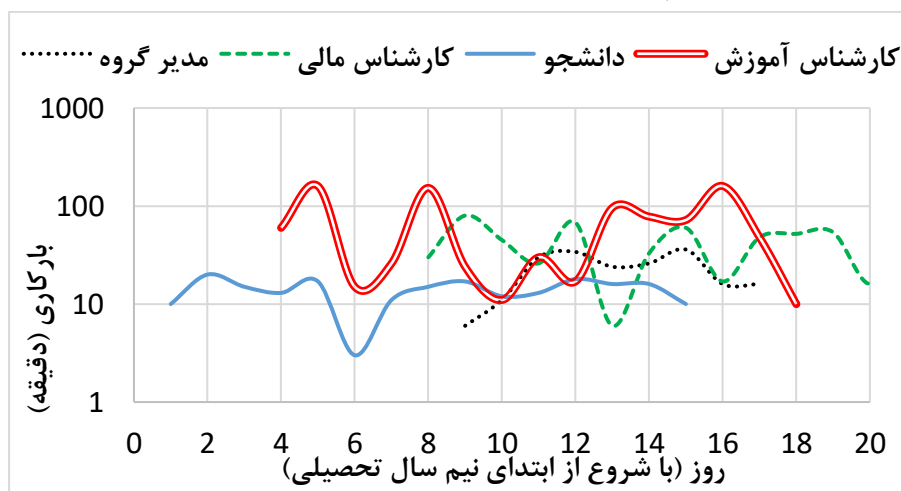
شکل (۴-۵) تعداد دانشجویان انتخاب واحد کرده، ثبت نامی و فارغ‌التحصیل در پایگاه داده موسسه آموزش عالی میرداماد

در این موسسه طی این ۱۲ نیم‌سال تحصیلی (در این مطالعه از اطلاعات نیم‌سال‌های تابستانی استفاده نشده است) پنج فرآیند اصلی وجود دارد که عبارت‌اند از:

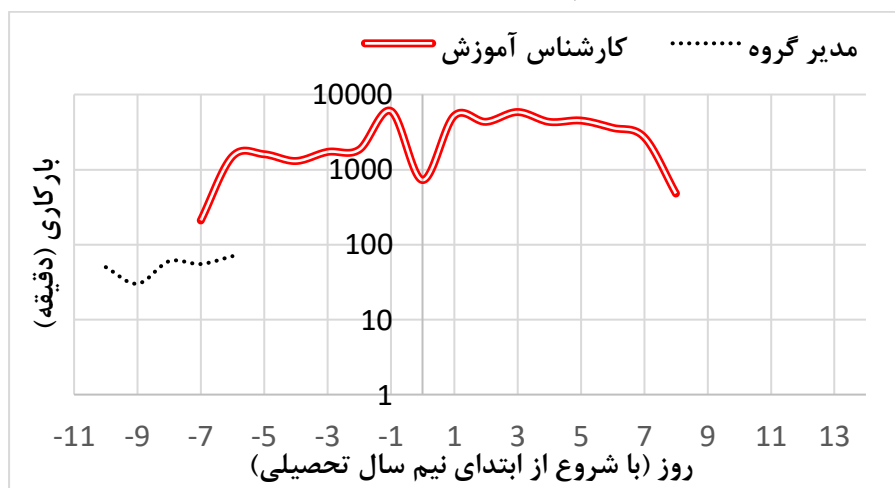
- ۱- ثبت‌نام دانشجویان جدیدالورود
 - ۲- انتخاب واحد دانشجویی
 - ۳- ارائه‌ی دروس نیم‌سال تحصیلی جاری
 - ۴- برگزاری امتحانات و اعلام نمرات
 - ۵- تسویه حساب و فارغ‌التحصیلی
- در این موسسه بارکاری منابع انسانی در ابتدای نیم‌سال تحصیلی مؤثر از سه فرآیند ثبت‌نام، ارائه‌ی دروس و انتخاب واحد است. کاربران و منابع انسانی در این موسسه در نقش‌های دانشجو، کارشناس آموزشی، کارشناس مالی، مدیر گروه آموزشی، مسئول ثبت‌نام و تائید اولیه مدارک، مسئول فارغ‌التحصیلی و اساتید قرار گرفته‌اند. این منابع بر اساس نقش تعریف شده در فرآیندهای ذکر شده مشغول فعالیت هستند. در شکل (۵-۵) میانگین بارکاری برخی از منابع طبق رابطه‌ی (۲-۵) برای سه فرآیند ذکر شده محاسبه شده است.



(الف) بارکاری در فرآیند ثبت نام دانشجویان جدیدالورود



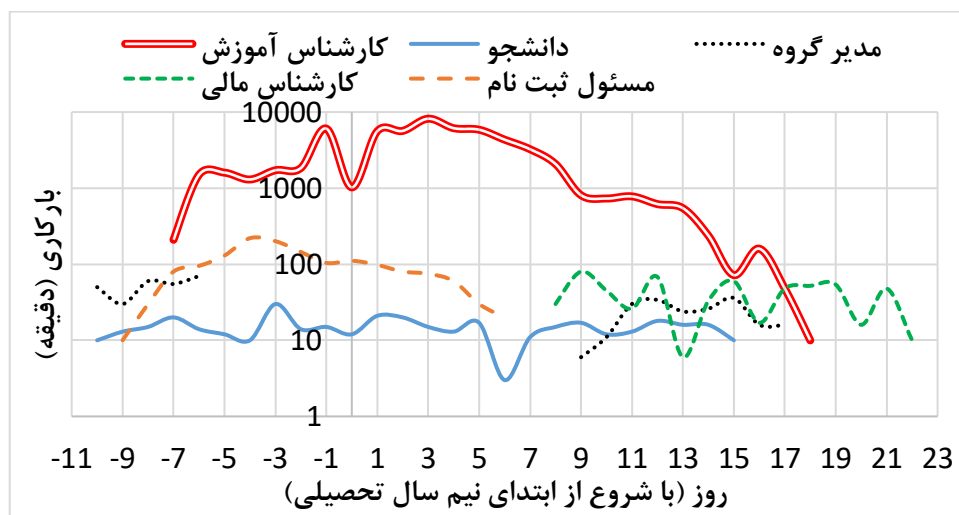
(ب) بارکاری در فرآیند انتخاب واحد



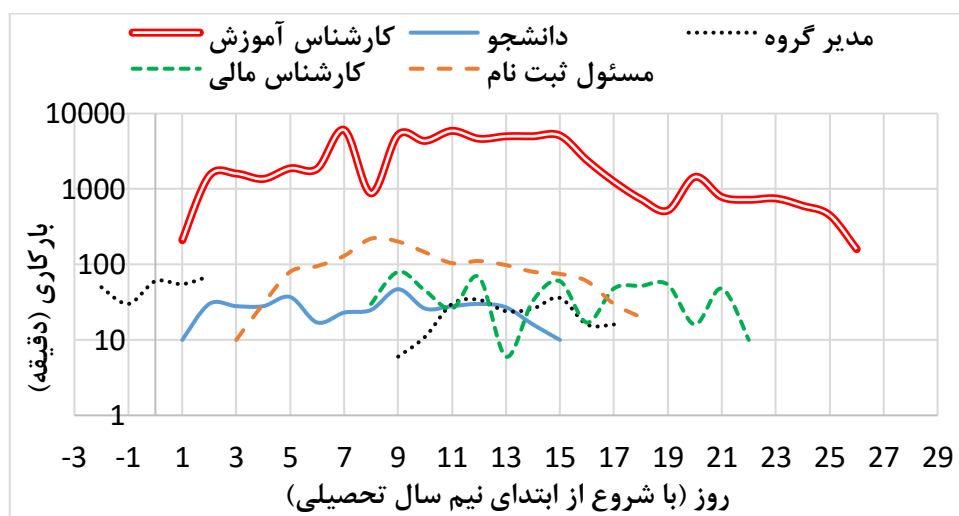
(ج) بارکاری در فرآیند ارائه دروس

شکل (۵-۵) میانگین بارکاری منابع انسانی در ابتدای نیم سال تحصیلی

هدف رساله در این بخش افزایش تعادل و یکنواختی بارکاری منابع درگیر در این فرآیندها مخصوصاً کارشناسان آموزشی موسسه است. بعد از اجرای الگوریتم پیشنهادی و تنظیم همروندی برای سه فرآیند ثبت نام، انتخاب واحد و ارائه‌ی دروس مقدار بردار تأخیر به صورت $D=[12.01,0,9.09]$ حاصل شد که نسبت به بردار تأخیر $D=[0,0,0]$ (حالت فعلی فرآیندها) مقدار LBFAT از ۱۴۲۰۴ به ۱۱۱۳۰ کاهش یافت. به بیان دیگر تنظیم همروندی باعث ۲۱/۶۴ درصد کاهش در مقدار LBFAT شد. این کاهش به معنای این است که اگر فرآیند ثبت نام و ارائه دروس به ترتیب با تأخیر ۱۲ و ۹ روز (در صورت امکان) انجام شوند بارکاری منابع تعادل و یکنواختی بیشتری خواهد داشت. شکل (۵-۶) وضعیت مجموع بارکاری منابع قبل و بعد از اجرای الگوریتم همروندی را نشان می‌دهد.



(الف) قبل از اجرای الگوریتم همروندی فرآیندها



(ب) بعد از اجرای الگوریتم همروندی فرآیندها

شکل (۵-۶) مجموع بارکاری منابع انسانی در سه فرآیند

همان‌طور که در شکل (۵-۶ ب) مشخص است الگوریتم هم‌روندی فرآیندها با ایجاد تأخیر در دو فرآیند ثبت‌نام (۱۲ روز) و ارائه‌ی دروس (۹ روز) توانست بارکاری کارشناس آموزش را یکنواخت‌تر کند و بارکاری دانشجویان که کم‌ترین بارکاری بین کاربران سیستم دارند را بین روزهای اول تا پانزدهم جمع کند. علاوه بر آن الگوریتم سعی کرده است مقادیر حداکثر بارکاری منابع را تا حد امکان بر روی هم منطبق کند که نشان از تعادل بیش‌تر بارکاری منابع است.

۵-۶ نتیجه‌گیری

تعادل بارکاری منابع در سیستم‌های مدیریت فرآیندهای کسب‌وکار (BPMS) به‌عنوان یک فاکتور مهم کارایی و پایداری سیستم مطرح است. قرار گرفتن بارکاری هر منبع نیز در یک سطح مشخص باعث افزایش کارایی و بازدهی منابع می‌شود. دو مفهوم تعادل بارکاری منابع و یکنواختی بارکاری هر منبع در یک سطح مشخص به‌عنوان اهداف اصلی این رساله تعریف شد و برای بهینه‌سازی میزان تعادل بارکاری و یکنواختی بارکاری هر منبع از ایده‌ی تنظیم هم‌روندی چند فرآیند استفاده شد و مسئله‌ی تنظیم هم‌روندی چند فرآیند را به‌عنوان یک مسئله‌ی بهینه‌سازی بیان شد. برای حل مسئله‌ی مطرح شده سه روش متفاوت برای حل آن ارائه شده است این روش‌ها عبارت‌اند از: الگوریتم جستجوی فضای حالت، الگوریتم بهینه‌سازی پویا و الگوریتم تکاملی PSO. پیچیدگی زمانی هر یک از این الگوریتم‌ها در این فصل محاسبه شد و در انتها، برای آزمایش سرعت اجرای روش‌های ارائه شده یک مثال مشخص از پنج فرآیند با الگوریتم‌های معرفی شده مورد آزمایش قرار گرفت و نتایج گزارش شده نشان داد که روش بهینه‌سازی پویا معرفی شده کاهش قابل توجهی در پیچیدگی زمانی نسبت به الگوریتم جستجوی فضای حالت داشت به‌طوری‌که نسبت سرعت آن برای آزمایش انجام شده ۳۷ ساعت به ۵/۸ سال بود. علاوه بر آن، حل مسئله مطرح شده با الگوریتم تکاملی PSO نشان داد الگوریتم تکاملی PSO به خوبی می‌تواند مسئله‌ی تنظیم هم‌روندی فرآیندها را حل کند و برای آزمایش مطرح شده تنها در ۳ دقیقه جواب بهینه را یافت. با افزایش جمعیت ذرات در PSO احتمال به دست آمدن جواب بهینه

افزایش می‌یابد اما از نظر تئوری تضمین نمی‌کند در همه‌ی شرایط جواب بهینه را به دست آورد. در انتهای این تحقیق، روی پایگاه داده‌ی واقعی میرداماد طی سال‌های ۱۳۹۰ الی ۱۳۹۶ یک مطالعه‌ی موردی انجام شد و نتایج آزمایش الگوریتم تنظیم هم‌روندی روی سه فرآیند ثبت‌نام، انتخاب واحد و ارائه‌ی دروس، ۲۱/۶۴ درصد بهبود در تعادل و یکنواختی بارکاری منابع را نشان داد.

۵-۷ پیشنهادات

در جابجایی و ایجاد تأخیر مصنوعی در ابتدای فرآیندها ممکن است محدودیت‌هایی دیگری وجود داشته باشد. به عنوان مثال در فرآیند ارائه‌ی دروس در بخش ۵-۵ نمی‌توان به دلخواه تأخیر ایجاد کرد زیرا ایجاد تأخیر در این فرآیند باعث ایجاد تأخیر در فرآیند انتخاب واحد می‌شود بنابراین ممکن است مقادیر مولفه‌های بردار Δt به هم وابسته باشند. لذا در ادامه‌ی تحقیقات انجام شده در این بخش پیشنهاد می‌شود مسئله‌ی تنظیم هم‌روندی با محدودیت‌های بیش‌تر تعریف شود و روش‌های مختلف حل آن بررسی شود.

منابع و مراجع

- [1] N. Russell, W. M. van der Aalst, A. H. Ter Hofstede, and D. Edmond, (2005), "Workflow resource patterns: Identification, representation and tool support", *Advanced Information Systems Engineering (Springer)*, vol. 3716, pp. 353-368.
- [2] H. Verbeek, J. C. Buijs, B. F. Van Dongen, and W. M. Van Der Aalst, (2011), "Xes, xesame, and prom 6", in *CAiSE Forum (Springer)*, vol. 72, pp. 60-75.
- [3] B.-H. Ha, J. Bae, and S.-H. Kang, (2004), "Workload balancing on agents for business process efficiency based on stochastic model", *IE interfaces (Springer)*, vol. 16, no. spc, pp. 76-81.
- [4] B.-H. Ha, J. Bae, Y. T. Park, and S.-H. Kang, (2006), "Development of process execution rules for workload balancing on agents", *Data & Knowledge Engineering*, vol. 56, no. 1, pp. 64-84.
- [5] S. Larbi and S. Mohamed, (2014), "Modeling the Scheduling Problem of Identical Parallel Machines with Load Balancing by Time Petri Nets", *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 7, no. 1, pp. 42-48.
- [6] Y. Xie, C.-F. Chien, and R.-Z. Tang, (2015), "A dynamic task assignment approach based on individual worklists for minimizing the cycle time of business processes", *Computers & Industrial Engineering*, vol. 99, no. 1, pp. 401-414.
- [7] E. A. Alluisi and B. B. Morgan Jr, (1976), "Engineering psychology and human performance", *Annual review of psychology*, vol. 27, no. 1, pp. 305-330.
- [8] W. M. van der Aalst, (2013), "Business process management: A comprehensive survey", *ISRN Software Engineering*, vol. 2013.
- [9] Y. Liu, J. Wang, Y. Yang, and J. Sun, (2008), "A semi-automatic approach for workflow staff assignment", *Computers in Industry*, vol. 59, no. 5, pp. 463-476.
- [10] Z. Huang, W. M. van der Aalst, X. Lu, and H. Duan, (2011), "Reinforcement learning based resource allocation in business process management", *Data & Knowledge Engineering*, vol. 70, no. 1, pp. 127-145.
- [11] A. G. Barto, (1998), "*Reinforcement learning: An introduction*", MIT press.
- [12] W. Van Der Aalst and K. M. Van Hee, (2004), "*Workflow management: models, methods, and systems*", MIT press, pp .
- [13] W. M. van der Aalst, K. M. van Hee, A. H. ter Hofstede, N. Sidorova, H. Verbeek, M. Voorhoeve, (2011), "Soundness of workflow nets: classification, decidability, and analysis", *Formal Aspects of Computing*, vol. 23, no. 3, pp. 333-363.
- [14] W. M. van der Aalst and A. H. Ter Hofstede, (2005), "YAWL: yet another workflow language", *Information systems*, vol. 30, no. 4, pp. 245-275.
- [15] D. E. Culler, J. P. Singh, and A. Gupta, (1999), "*Parallel computer architecture: a hardware/software approach*", Gulf Professional Publishing.
- [16] D. Grosu and A. T. Chronopoulos, (2004), "Algorithmic mechanism design for load balancing in distributed systems", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 77-84.
- [17] E. Rahm, (1996), "Dynamic load balancing in parallel database systems", in *European Conference on Parallel Processing (Springer)*, Berlin, Heidelberg, pp. 37-52.

- [18] L.-j. Jin, F. Casati, M. Sayal, and M.-C. Shan, (2001), "Load balancing in distributed workflow management system", in *Proceedings of the 2001 ACM symposium on Applied computing*, pp. 522-530.
- [19] W. Zhao, L. Yang, H. Liu, and R. Wu, (2015), "The Optimization of Resource Allocation Based on Process Mining", *Springer*, pp. 341-353.
- [20] M. Zur Muehlen, (2004), "Organizational management in workflow applications—issues and perspectives", *Information Technology and Management*, vol. 5, no. 3-4, pp. 271-291.
- [21] J. Xu, C. Liu, and X. Zhao, (2008), "Resource allocation vs. business process improvement: How they impact on each other" in *BPM*, pp. 228-243 ,
- [22] A. S. Nisafani, A. Wibisono, S. Kim, and H. Bae, (2014), "Bayesian Selection Rule for Human-Resource Selection in Business Process Management Systems", *Journal of Society for e-Business Studies*, vol. 17, no. 1, pp. 53-74.
- [23] A. Wibisono, A. S. Nisafani, H. Bae, and Y.-J. Park, (2015), "On-the-Fly Performance-Aware Human Resource Allocation in the Business Process Management Systems Environment Using Naïve Bayes", *Springer*, pp. 70-80.
- [24] S. Rhee, H. Bae, D. Ahn, and Y. Seo, (2003), "Efficient workflow management through the introduction of TOC concepts", *Proceedings of the 8th annual international conference on industrial engineering theory, applications and practice (IJIE2003)*.
- [25] W. M. Van Der Aalst, A. H. Ter Hofstede, and M. Weske, (2003), "Business process management: A survey", *International conference on business process management*, vol. 2678, pp. 1-12.
- [26] M. Shen, G.-H. Tzeng, and D.-R. Liu, (2003), "Multi-criteria task assignment in workflow management systems", *System Sciences*, *Proceedings of the 36th Annual Hawaii International Conference on 2003, IEEE* , pp. 1-9.
- [27] X. Liu, J. Chen, Y. Ji, and Y. Yu, (2015), "Q-learning Algorithm for Task Allocation Based on Social Relation", *Process-Aware Systems*, pp. 49-58, Berlin, Heidelberg.
- [28] T. Hegazy, (1999), "Optimization of resource allocation and leveling using genetic algorithms", *Journal of construction engineering and management*, vol. 125, no. 3, pp. 167-175.
- [29] S.-S. Leu and C.-H. Yang, (1999), "A genetic-algorithm-based resource-constrained construction scheduling system", *Construction Management & Economics*, vol. 17, no. 6, pp. 767-776.
- [30] J. Alcaraz and C. Maroto, (2001), "A robust genetic algorithm for resource allocation in project scheduling", *Annals of Operations Research*, vol. 102, no. 1-4, pp. 83-109.
- [31] C. V. I. Debby, (2011), "Genetic Algorithm Based Resource Allocation for Business Processes," *University of Macau*.
- [32] A. Kumar, W. M. van der Aalst, and E. M. Verbeek, (2002), "Dynamic work distribution in workflow management systems: How to balance quality and performance", *Journal of Management Information Systems*, vol. 18, no. 3, pp. 157-193.

- [33] Y. Narahari, N. Viswanadham, and V. K. Kumar, (1999), "Lead time modeling and acceleration of product design and development", *IEEE Transactions on robotics and automation*, vol. 15, no. 5, pp. 882-896.
- [34] J. H. Son and M. H. Kim, (2001), "Improving the performance of time-constrained workflow processing", *Journal of Systems and Software*, vol. 58, no. 3, pp. 211-219.
- [35] D.-H. Chang, J. H. Son, and M. H. Kim, (2002), "Critical path identification in the context of a workflow", *Information and software Technology*, vol. 44, no. 7, pp. 405-417.
- [36] W. M. Van Der Aalst, K. M. Van Hee, and H. A. Reijers, (2000), "Analysis of discrete-time stochastic petri nets", *Statistica Neerlandica*, vol. 54, no. 2, pp. 237-255.
- [37] K. van Hee and H. Reijers, (1999), "An analytical method for computing throughput times in stochastic workflow nets", *Simulation in industry*.
- [38] L. Zerguini and K. M. van Hee, (2002), "A new reduction method for the analysis of large workflow models", *Promise*, vol. pp. 188-201.
- [39] J. Eder, H. Pichler, W. Gruber, and M. Ninaus, (2003), "Personal schedules for workflow systems", *Business Process Management, Springer*, pp. 216-231.
- [40] S. Dreyfus, (2002), "Richard Bellman on the birth of dynamic programming", *Operations Research*, vol. 50, no. 1, pp. 48-51.
- [41] P. Diban, M. K. A. Aziz, D. C. Foo, X. Jia, Z. Li, and R. R. Tan, (2016), "Optimal biomass plantation replanting policy using dynamic programming", *Journal of Cleaner Production*, vol. 126, no. pp. 409-418.
- [42] R. Jia, S. J. Mellon, S. Hansjee, A. Monk, D. Murray, and J. A. Noble, (2016), "Automatic bone segmentation in ultrasound images using local phase features and dynamic programming", *Biomedical Imaging (ISBI), IEEE 13th International Symposium on 2016*, pp. 1005-1008.
- [43] M. Roozegar, M. Mahjoob, and M. Jahromi, (2016), "Optimal motion planning and control of a nonholonomic spherical robot using dynamic programming approach: simulation and experimental results", *Mechatronics*, vol. 39, no. 1, pp.174-184.
- [44] C. Finck and R. Li, (2016), "Operational load shaping of office buildings connected to thermal energy storage using dynamic programming", *Aalborg University*.
- [45] J. Kennedy, (2011), "Particle swarm optimization", *Springer*, pp. 760-766.
- [46] Z. Liu, P. Zhu, W. Chen, and R.-J. Yang, (2015), "Improved particle swarm optimization algorithm using design of experiment and data mining techniques", *Structural and Multidisciplinary Optimization*, vol. 52, no. 4, pp. 813-826.
- [47] C. Ou-Yang, H.-J. Cheng, and Y.-C. Juan, (2015), "An Integrated mining approach to discover business process models with parallel structures: towards fitness improvement", *International Journal of Production Research*, vol. 53, no. 13, pp. 3888-3916.
- [48] H.-J. Cheng, C. Ou-Yang, and Y.-C. Juan, (2015), "A hybrid approach to extract business process models with high fitness and precision", *Journal of Industrial and Production Engineering*, vol. 32, no. 6, pp. 351-359.

Abstract

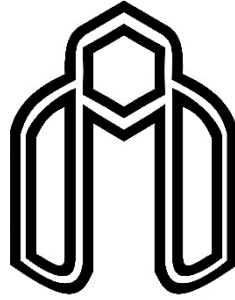
Management of human resources is one of the most important goals of business process management systems (BPMS) which strive to improve the quality of products and services. Workload balancing and workload uniformity of resources improve the performance of business process execution in BPMS. Different techniques and methods are used for modeling of the business process execution such as Markov Decision Processes (MDPs), Queue Network Model (QNM) and Workflow Network (WF-net). In this thesis, a statistical process-aware model is presented for modeling of business process execution that is called Extended Markov Decision Processes (XMDPs). This model is a developed version of MDPs, so that the business process execution can map on the XMDPs model better than the MDPs model.

The main idea in the proposed model is to support more events that they map on the proposed model. Thus, the main goal of optimization in mapping business process execution on the proposed model is workload balancing and creating uniformity in the workload of resources. This problem is solved by using reinforcement learning which helps to make better resource allocation in order to achieve the mentioned goals. The proposed method is compared with Reinforcement Learning Resource Allocation Mechanism (RLRAM) based on the MDPs model on the BPI Challenge 2102 data set. The experimental results show that the problem of workload balancing and uniformity of workload can be managed 12.7% better with a proper mapping of the business process execution on the proposed model.

Furthermore, by presenting and employing a statistical method on event logs, a kind of dependency of tasks is extracted. By considering the mentioned tasks dependency, some algorithms are presented and suggested to improve resource allocation performance. The proposed algorithms are also compared with other algorithms in the research background. The observed results show the optimal effect of the concurrent execution of tasks in the resources allocation performance. In continued research in this thesis, the “tuning concurrency of business processes” problem is introduced with the aim of balancing and uniformity of workload. Solving this problem is studied in three different methods. The tuning concurrency of business processes have experimented on real data set. The experimental results show improvement by 21.64% in the balancing and uniformity of resource workloads.

Keywords

Bussines process management system, Workload balancing, Workload uniformity, Business process modeling, Process-aware statistical model, Tasks dependency, Tuning concurrency of business processes.



Shahrood University of Technology

**Computer & Information Technology Engineering
Artificial Intelligence Group**

A statistical process-aware model for workload balancing based on reinforcement learning

Thesis submitted in partial fulfillment of the requirements
for the degree of PhD in Computer Engineering

Author

Mehdi Yaghoubi

Supervisor

Dr. Morteza Zahedi

Advisor

Dr. Alireza Ahmadyfard

August 2017