علوم رایانش و فناوری اطلاعات

نشریه علمی انجمن کامپیوتر ایران

صاحب امتیاز: انجمن کامپیوتر ایران

مدیر مسئول: دکتر جعفر حبیبی

سردبیر: دکتر احمد خونساری

شورای علمی

حمیدرضا ربیعی، استاد دانشگاه صنعتی شریف
حمید سربازی‌آزاد، استاد دانشگاه صنعتی شریف
کریم فائز، استاد دانشگاه صنعتی امیرکبیر
اکبر غفارپور رهبر، استاد دانشگاه صنعتی سهند
احسان‌الله کبیر، استاد دانشگاه تربیت مدرس
کیوان ناوی، استاد دانشگاه شهید بهشتی
ناصر یزدانی، استاد دانشگاه تهران
محمد حسین یغمایی مقدم، استاد دانشگاه فردوسی مشهد
مرتضی آنالویی، دانشیار دانشگاه علم و صنعت ایران
محسن ابراهیمی مقدم، دانشیار دانشگاه شهید بهشتی
حسین اسدی، دانشیار دانشگاه صنعتی شریف
احمد اکبری ازیرانی، دانشیار دانشگاه علم و صنعت ایران
رضا برنگی، دانشیار دانشگاه علم و صنعت ایران
حسین پدرام، دانشیار دانشگاه صنعتی امیرکبیر
نصراله مقدم چرکری، دانشیار دانشگاه تربیت مدرس
قاسم جابری‌پور، دانشیار دانشگاه شهید بهشتی
جعفر حبیبی، دانشیار دانشگاه صنعتی شریف
امیر حسین جهانگیر، دانشیار دانشگاه صنعتی شریف
شاهین حسابی، دانشیار دانشگاه صنعتی شریف
سید حمید حاجی سید جوادی، دانشیار دانشگاه شاهد
مسعود رهگذر، دانشیار دانشگاه تهران
مهدی صدیقی، دانشیار دانشگاه صنعتی امیرکبیر
هشام فیلی، دانشیار دانشگاه تهران
عبدالرسول قاسمی، دانشیار دانشگاه خواجه نصرالدین طوسی
مقصود عباسپور، دانشیار دانشگاه شهید بهشتی
محمد عبداللهی ازگمی، دانشیار دانشگاه علم و صنعت ایران
مهدی کارگهی، دانشیار دانشگاه تهران
مازیار گودرزی، دانشیار دانشگاه صنعتی شریف
ناصر مزینی، دانشیار دانشگاه علم و صنعت ایران

همکاران دفتر نشریه

لیلا نورانی
مهدی دولتی

نشانی

تهران، خیابان آزادی، ضلع غربی دانشگاه صنعتی شریف، کوچه شهید ولی‌ا... صادقی، پلاک ۲۶، طبقه ۴، واحد ۱۶، دفتر انجمن کامپیوتر ایران، نشـریه علوم رایانش و فناوری اطلاعات

تلفن: ۶۶۰۸۷۲۲۴-۶۶۰۳۲۰۰۰

دورنگار: ۶۶۰۲۱۱۴۹

پست‌الکترونیکی: csitjour@gmail.com

نشانی سایت: http://csi.org.ir/fa/publication/archive/name/csit

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

لیتوگرافی، چاپ و صحافی:

علوم رایانش و فناوری اطلاعات نشریه علمی انجمن کامپیوتر ایران

مجلد ۱۳	شماره ۲	۱۳۹۴

فهرست مقالات

- **یافتن بهترین مکان جغرافیایی برای مرکز داده سبز در ایران با توجه به شرایط اقلیمی، سیاسی و اجتماعی ۱**
مجید حاجی بابا و سعید گرگین

- **ارائه روش بهبود یافته زمان‌بندی کارها در محیط ابر با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات ۱۴**
فاطمه عبادی‌فرد و احمد اکبری

- **افزایش طول‌عمر حافظه‌ی نهان سطح آخر غیرفرار با کمک بلوک‌های ذخیره ۲۳**
محمدرضا جوکار، محمد ارجمند و حمید سربازی‌آزاد

- **نگاشت و زمان‌بندی همزمان وظایف و ارتباطات انرژی‌آگاه بی‌درنگ در ساختارهای چندهسته‌ای ۳۲**
امین‌اله مه‌آبادی و فاطمه عسگری بیدهندی

- **نظرکاوی بین‌زبانی با استفاده از ویژگی‌های معنایی ۴۷**
شیما اسمعیلی تفت و آزاده شاکری

- **ارائه یک شبکه روی تراشه با کارآیی بالا و توان مصرفی کم برای شبکه‌های عصبی ۶۰**
نسرین اکبری، بیتا دبیری و مهدی مدرسی

علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۱-۱۳
مقاله عادی

یافتن بهترین مکان جغرافیایی برای مرکز داده سبز در ایران با توجه به شرایط اقلیمی، سیاسی و اجتماعی

مجید حاجی بابا سعید گرگین

پژوهشکده برق و کامپیوتر، سازمان پژوهش‌های علمی و صنعتی ایران، تهران، ایران

چکیده

شناسایی و تعیین مختصات جغرافیایی مکان مراکز داده، برای سازمان‌ها دارای اهمیت بسزایی است. ارزیابی،شناسایی و تعیین مختصات به فاکتورها، معیارها و پارامترهای مختلفی وابسته است که این تنوع پیچیدگی، تصمیم‌گیری برای مدیران سطح بالا را دوچندان می‌سازد. این مقاله برای اولین بار، به بررسی جامع معیارهای تصمیم‌گیری و تبیین گزینه‌های در دسترس برای میزبانی مراکز داده بومی سبز در ایران می‌پردازد. سیستم ارائه شده در این مقاله، با بررسی و ارزیابی لیست جامعی از عوامل بالقوه و با توجه به شرایط اقلیمی، سیاسی و اجتماعی ایران و دیگر فاکتورهای موثر، استان‌ها را به منظور تعیین مختصات جغرافیایی مراکز داده رتبه‌بندی می‌کند. در این سیستم بیش از ۴۰ معیار در قالب ۴ دسته اصلی شامل آب و هوا، حوادث طبیعی، امکانات و حوادث غیرطبیعی مورد ارزیابی قرار می‌گیرد. فرآیند استنتاج این سیستم، با ادغام و وزن‌دهی به پارامترها با استفاده از فرآیند تحلیل سلسله مراتبی و براساس نظرسنجی بین متخصصان و استفاده از رای‌گیری اکثریت، با فرآیندی تطبیقی و انعطاف‌پذیر تکمیل می‌شود و با صحت بسیار بالا به ترتیب نواحی جغرافیایی مناسب را پیشنهاد می‌دهد.

کلمات کلیدی: مرکز داده، مرکز داده سبز، جایابی مرکز داده، معیارهای انتخاب مکان مرکز داده، رتبه‌بندی استان‌ها.

۱- مقدمه

محل استقرار مرکز داده اغلب جزء اولین معیارهای مورد بحث در هنگام برپایی یک مرکز داده جدید است و برای اکثر سازمان‌ها حیاتی است؛ نه تنها به این دلیل که میزبان تجهیزات و سرویس‌هایشان خواهد بود، بلکه این مکان تا مدت زیادی تغییر نخواهد کرد و از آنجایی‌که انتخاب محل استقرار برای هر مرکز داده تنها یک بار صورت می‌پذیرد باید توجه ویژه‌ای به آن شود.

مکان یک مرکز داده باید با توجه به عوامل مختلفی انتخاب شود و هر سازمان هنگام برپایی مرکز داده، خود را با مجموعه‌ای از معیارها روبرو می‌بینند که باید مدنظر قرار دهد. در مورد آنچه مهم‌ترین عوامل تصمیم‌گیری در جایابی یک مرکز داده هستند، مطالعاتی صورت گرفته است [۲۵، ۲۶]. به طور کلی عوامل مهم در انتخاب مکان مرکز داده عبارتند از: وضعیت آب و هوا، حوادث طبیعی مانند سیل و زمین‌لرزه، حوادث غیر طبیعی مانند جنگ و تروریسم، دسترسی به زیرساخت‌های ارتباطی، تامین پایدار برق و الکتریسیته، راحتی حمل و نقل و دسترس بودن نیروی انسانی واجد شرایط برای کار در مرکز داده.

البته عوامل موثر دیگری همچون فاصله مناسب از منابع لرزشی و تولیدکننده نویز مانند فرودگاه، اتوبان، راه‌آهن، بزرگراه، استادیوم، پالایشگاه، خط لوله و غیره و نزدیکی به ایستگاه‌های امداد و نجات و حمل و نقل وجود دارند که در درجه دوم از اهمیت قرار می‌گیرند.

اهمیت انتخاب مکان مرکز داده با توجه به معیارهای مختلف و تأکید بر اهمیت حفاظت از داده‌ها و تجهیزات، باعث شده تا عامل نزدیکی به دفاتر مرکزی آن‌ها کم‌اهمیت‌تر شود و فرصتی برای دیگر شهرها به منظور میزبانی مراکز داده پیش آید.

در این مقاله بیش از ۴۰ معیار در قالب ۴ دسته اصلی و ۱۳ زیر دسته مورد ارزیابی قرار گرفته است. دسته‌های اصلی شامل آب و هوا، حوادث طبیعی، امکانات و حوادث غیرطبیعی می‌باشند. آب و هوا شامل دما و رطوبت، حوادث طبیعی

شامل زمین‌لرزه، خشکسالی، سیل، طوفان و آتشفشان، امکانات شامل بستر ارتباطی، تأمین انرژی، منابع انسانی و توسعه‌یافتگی و حوادث غیرطبیعی شامل جنگ و تروریسم می‌باشد که زیر دسته‌های این معیارها را تشکیل می‌دهند.

در بخش‌های بعدی این مقاله هر کدام از این دسته‌ها و زیر دسته‌ها به منظور یافتن مطلوب‌ترین استان برای یک مرکز داده بررسی خواهند شد. سپس هر یک از استان‌ها به‌ترتیب مناسب بودن برای میزبانی مرکز داده، با توجه به آن عامل خاص رتبه‌بندی می‌شوند. به هر یک از عوامل یک ضریب تأثیر اضافه خواهد شد و گویای اهمیت آن عامل از نقطه نظر فناوری اطلاعات و سبز بودن مرکز داده می‌باشد. در نهایت با به دست آوردن میانگین وزنی، بهترین استان‌ها برای میزبانی مرکز داده سبز معرفی خواهند شد.

۲- آب و هوا

بهره‌گیری از سرمایش رایگان عامل بسیار مهمی است که برای یافتن مکان یک مرکز داده، به ویژه در مورد یک مرکز داده سبز[1] باید مد نظر قرار داد. از آنجایی که خنک‌کننده‌ها بخش اعظمی از مصرف انرژی در مراکز داده را دارند، افزایش ساعات استفاده‌ی مرکز داده از صرفه‌جوگرها[2] به منظور صرفه‌جویی اقتصادی، بیشترین تأثیر را در بهره‌گیری انرژی در مرکز داده خواهد داشت. صرفه‌جوگرها سرمایش را از طریق هوای بیرون به منظور کاهش یا حذف نیاز به دستگاه‌های خنک‌کننده فراهم می‌آورند. تعداد ساعاتی که می‌توان از صرفه‌جوگرها استفاده کرد براساس وضعیت آب و هوایی محل مرکز داده متفاوت است. بنابراین یافتن مکانی که در آن پایین بودن دمای محیط نیاز به خنک‌کننده‌ها را کم کند، فرصت مناسبی برای بهره‌گیری از سرمایش رایگان فراهم خواهد کرد.

در این بخش وضعیت آب و هوایی شهرهای مختلف کشور با هم مقایسه شده و با شرایطی که یک مرکز داده نیاز دارد مطابقت داده شده و شهر (استان)ها رتبه‌بندی می‌شوند.

طبق گزارش ASHRAE در سال ۲۰۱۱ محدوده دمای هوا برای سرورها در یک مرکز داده باید بین ۱۸ تا ۲۷ درجه سانتی گراد و رطوبت نسبی هوا ۶۰٪ باشد [۱۹، ۲۰].

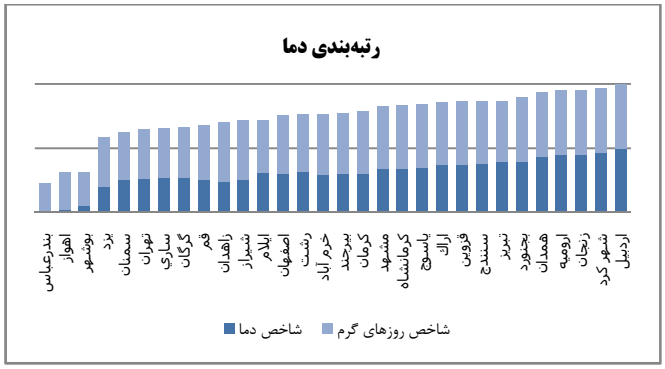
حال وضعیت آب و هوای استان‌های کشور (مراکز استان‌ها) برای تطابق با این عوامل بررسی خواهد شد. باید توجه شود که در اینجا از میانگین به جای آمار ساعت به ساعت وضعیت آب و هوای شهرها استفاده شده است.

۲-۱- دما

برای استفاده از سرمایش رایگان از طریق هوای بیرون به طور کلی دمای مناسب محیط باید در محدوده ۱۰- تا ۱۵ (با آستانه حداقل ۳۰-) درجه سانتی‌گراد باشد. با مقایسه این مقادیر با میانگین حداقل و میانگین حداکثر دمای شهرها و محاسبه میزان انحراف آنها، رتبه‌بندی شهرها را به‌دست می‌آوریم. آمار دیگری که در این زمینه می‌تواند به ما کمک کند، میانگین تعداد روزهایی از سال است که یک شهر دمای حداقل ۲۱ درجه یا بیش از آن داشته است که توسط ایستگاه‌های سینوپتیک کشور [۱۸] به‌دست آمده است و گویای تعداد روزهایی است که قطعاً نمی‌توان از سرمایش رایگان از طریق هوای بیرون در هیچ ساعتی از روز استفاده کرد.

پس از بررسی آمار از آنجایی‌که همه شهرها حداقل دمایی بالاتر از ۱۰- (و البته ۳۰-) درجه سانتی‌گراد دارند این آستانه نادیده گرفته شده و تنها حد کمتر از ۱۵ درجه سانتی‌گراد به کار گرفته خواهد شد. برای اینکه نوسانات هوا تا حدی در نظر گرفته شود از میانگین حداقل دما و میانگین حداکثر دما به جای میانگین دما استفاده می‌کنیم. ضمناً از آنجایی‌که با میانگین حداقل و حداکثر دما کار می‌کنیم بنابراین هر چه فاصله بیشتری از ۱۵ درجه داشته باشیم، رتبه بالاتری خواهیم داشت.

با بررسی میانگین حداقل و میانگین حداکثر دمای مراکز استان‌ها در پنج سال ۸۶ تا۹۰ [۷] و میانگین تعداد روزهایی که یک شهر در دراز مدت دمای حداقل ۲۱ درجه یا بیش از آن داشته است [۱۸]، نمودار رتبه‌بندی شهرها بر حسب معیار دما براساس شکل ۱ می‌باشد.



شکل ۱- نمودار رتبه‌بندی شهرها براساس شاخص دما

این رتبه‌بندی طبق رابطه (۱) به‌دست آمده است:

$$\text{score} = A * 0.5 + B * 0.5 \qquad (۱)$$

که در آن A مجموع فاصله میانگین حداقل و میانگین حداکثر از مقدار مطلوب ۱۵ درجه سانتی‌گراد است و B تعداد روزهایی از سال است که ممکن است بتوان در ساعاتی از روز از سرمایش رایگان از طریق هوای بیرون استفاده کرد. A و B نیز طبق رابطه (۲) و (۳) به‌دست می‌آیند:

$$A = \frac{(a-15)+(b-15)+25}{35} \qquad (۲)$$

$$B = \frac{365-c}{365} \qquad (۳)$$

که در آن a میانگین حداقل دما، b میانگین حداکثر دما و ۱۵ حد بالای دما می‌باشد که به عنوان معیار فاصله استفاده شده است. عدد ۲۵ به‌عنوان بیشترین فاصله منفی به‌منظور مثبت کردن مقادیر استفاده شده و عدد ۳۵ به‌عنوان بیشترین فاصله مثبت برای نرمال کردن مقادیر بین ۰ و ۱ استفاده شده است. مقدار c میانگین تعداد روزهایی از سال است که نمی‌توان در هیچ ساعتی از روز از سرمایش رایگان استفاده کرد.

بنابراین اردبیل بهترین شهر (استان) برای بهره بردن از سرمایش رایگان است و اگر تنها سبز بودن یک مرکز داده اولویت باشد، بدون توجه به دیگر معیارها، می‌توان گفت رتبه‌بندی شکل ۱رتبه‌بندی نهایی خواهد بود، که البته فرض صحیحی نیست.

۲-۲- رطوبت نسبی

رطوبت نسبی واژه‌ای است که به منظور بیان وضعیت رطوبت هوا استفاده می‌شود و در حقیقت بیان می‌کند که در یک دمای مشخص، هوا به چه میزان به حالت اشباع نزدیک است. در حجم هوایی با دما و فشار معین، رطوبت نسبی بیان کننده نسبت بخار آب موجود به مقدار آب مورد نیاز جهت اشباع آن حجم هوا می‌باشد.

چون فشار واقعی بخار آب موجود در هوا در طول شبانه روز تغییر چندانی نمی‌یابد، عامل اصلی تغییر رطوبت نسبی روزانه، تغییر دما می‌باشد. با سرد شدن هوا در شب و گرم شدن آن در طول روز رطوبت نسبی به ترتیب افزایش و کاهش می‌یابد. حداکثر رطوبت نسبی، معمولاً در صبحگاه به هنگام وقوع کمینه دما و حداقل رطوبت نسبی در ظهر به‌هنگام وقوع بیشینه دما اتفاق می‌افتد.

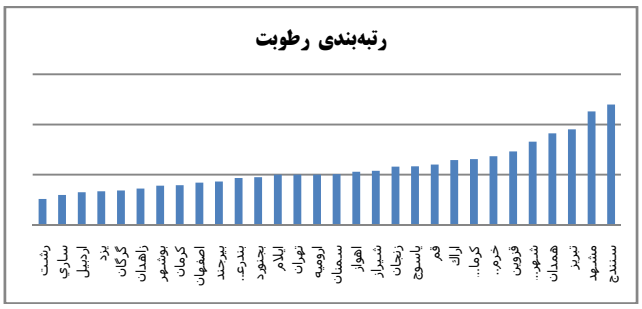
در صورتی که رطوبت هوای بیرون برای سرورها مناسب باشد دیگر نیازی به مرطوب کردن هوای بیرون نیست. ازآنجایی‌که طبق گزارش ASHRAE در سال ۲۰۱۱ رطوبت نسبی بین ۲۰٪ تا ۸۰٪ برای مراکز داده مناسب دیده شده است [۱۹]، این مقادیر را به عنوان آستانه قرار داده و شهرها را براساس آن رتبه‌بندی می‌کنیم.

برای ارزیابی تناسب شهرها میانگین حداکثر و حداقل رطوبت شهرها را حساب کرده و انحراف مجموع آن‌ها از ۵۰٪ را ملاک قرار داده و براساس کمترین انحراف به بیشترین، رتبه‌بندی می‌کنیم. در نهایت نمودار شکل ۲ را خواهیم داشت که با تحلیل اطلاعات به دست آمده از سالنامه آماری مراکز استان‌ها در دهه ۸۱ تا ۹۰ شکل گرفته است [۷].

برای این رتبه‌بندی از رابطه (۴) استفاده شده است:

$$\text{score} = \log \frac{1}{\left|\frac{a+b}{2} - 50\right|} \qquad (۴)$$

که در آن a میانگین کمینه و b میانگین بیشینه رطوبت نسبی مراکز استان در دهه اخیر می‌باشد و ۵۰ میانگین رطوبت نسبی مناسب برای سرورها می‌باشد.



شکل ۲- رتبه‌بندی شهرها براساس شاخص رطوبت

با استفاده از این رابطه شهرهایی با رطوبت بالای ۸۰٪ و زیر ۲۰٪ در انتهای رتبه‌بندی قرار دارند.

۳- حوادث طبیعی

عملکرد یک مرکز داده می‌تواند به راحتی توسط حوادث طبیعی به خطر بیافتد. حوادث طبیعی از جمله زمین‌لرزه، آتشفشان، طوفان، سیل و گردباد همیشه برای یک مرکز داده تهدید به حساب می‌آید. معمولا فرض می‌شود که احتمال وقوع این حوادث بسیار کم است و در صورت وقوع شدت زیادی نخواهند داشت؛ اما به طور کلی مکان یک مرکز داده باید جایی باشد که به طور بالقوه دور از این خطرات باشد.

در اینجا حوادث غیرمترقبه را به طور کلی به پنج دسته زمین‌لرزه، خشکسالی، طوفان، سیل و آتشفشان تقسیم کرده و در بخش‌های بعدی هر یک را جداگانه با توجه به شرایط اقلیمی ایران بررسی می‌نماییم.

۳-۱- زمین‌لرزه

ایران کشوری لرزه‌خیز است و در ایران زلزله به عنوان اولین فاجعه طبیعی برای انسان محسوب می‌شود. ایران بر روی یکی از دو کمربند بزرگ لرزه‌خیزی جهان قرار دارد و در ردیف ۱۰ کشور اول زلزله‌خیز جهان قرار دارد که هر ساله زلزله‌های بزرگ و کوچکی در آن به وقوع می‌پیوندد.

مکان استقرار یک مرکز داده باید از نقطه نظر زلزله بررسی شده و در محل مناسبی قرار گیرد. نه تنها احتمال وقوع زمین‌لرزه باید مدنظر قرار گیرد، بلکه جنس خاک زمین نیز مطرح است. در خاک‌های ماسه‌ای با افزایش غشاء آب حفره‌ای، مقاومت خاک کاهش می‌یابد که به این پدیده روانگرایی می‌گویند و بیشتر در ماسه شل و اشباع رخ می‌دهد. در این حالت خاک به صورت یک مایع رفتار می‌کند. اگر خاک مکان استقرار مرکز داده مناسب نباشد، در صورت وقوع زمین‌لرزه حتی اگر ساختمان مرکز داده مقاوم باشد، خاک زیرین آن همانند یک مایع عمل کرده و موجب حرکت ساختمان و در نهایت خرابی‌هایی برای توزیع برق، آب، سوخت و غیره می‌شود و باعث از کار افتادن مرکز داده خواهد شد.

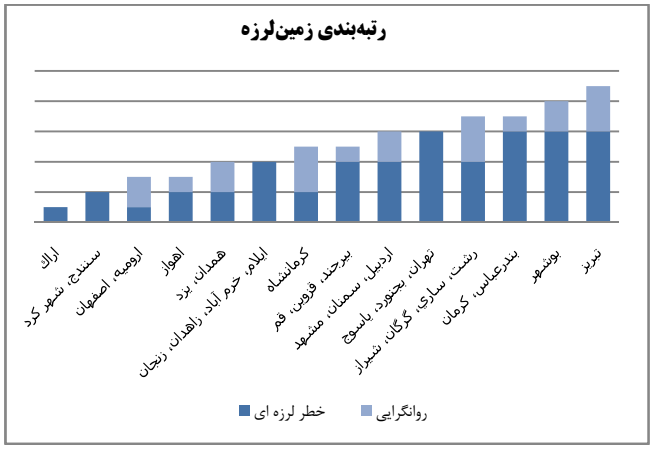
برای ارزیابی خطر زلزله از ترکیب قابلیت روانگرایی طبق نقشه قابلیت روانگرایی [۱۵] و خطر لرزه‌ای طبق نقشه خطر لرزه‌ای [۱۵] و نقشه توزیع گسل‌ها [۱۶] استفاده شده است. نقشه خطر لرزه‌ای نشان‌دهنده خطر زلزله در مناطق مختلف ایران با ۵۰٪ احتمال وقوع در ۵۰ سال آینده می‌باشد.

برای رتبه‌بندی شهرها از امتیازدهی جدول ۱ استفاده شده است که براساس شدت خطر لرزه‌ای و قابلیت روانگرایی امتیازدهی شده است.

جدول ۱- امتیاز شاخص‌های مربوط به زمین‌لرزه

خصیصه	شدت	ضریب خطر زلزله
خطر لرزه‌ای	خیلی زیاد	۶
	زیاد	۴
	متوسط	۲
	کم	۱
قابلیت روانگرایی	زیاد	۳
	متوسط	۲
	کم	۱
	بدون روانگرایی	۰

با استفاده از آمار و امتیازهای فوق رتبه‌بندی شکل ۳ را برای استان‌های کشور خواهیم داشت که نشان دهنده زلزله‌خیز بودن شهرهاست.



شکل۳- رتبه‌بندی شهرها براساس خطر زمین‌لرزه

رتبه‌بندی فوق طبق رابطه (۵) به‌دست آمده است:

$$score = a + b \quad (۵)$$

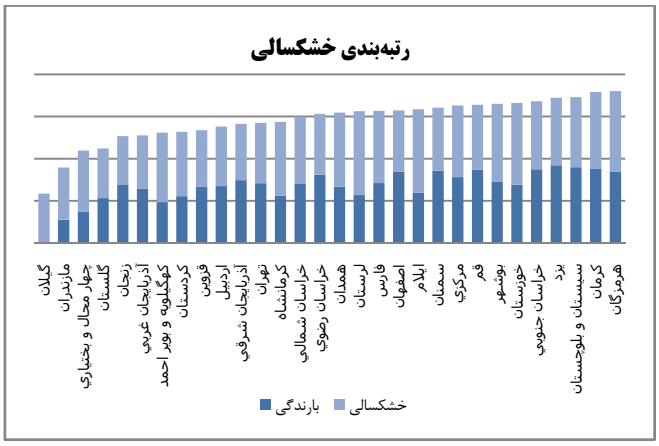
که در آن a ضریب خطر لرزه‌ای و b ضریب قابلیت روانگرایی طبق جدول ۱ است.

۳-۲- خشکسالی

از آنجایی‌که بسیاری از مراکز داده از جمله مراکز داده سبز، به سوی سرمایش مبتنی بر آب[۳] حرکت می‌کنند، تأمین فراوان آب برای مراکز داده حیاتی است. کمبود آب و جیره‌بندی آب در برخی شهرها می‌تواند تأثیرات چشمگیری بر مراکز داده‌ای داشته باشد که نیاز به آب کافی برای سرمایش دارند.

معمولاً برای اندازه‌گیری خشکسالی از شاخص شدت خشکسالی پالمر[۴] استفاده می‌شود که در مقیاس زمانی ماهانه به کار می‌رود و براساس دما، بارش، رطوبت خاک، تبخیر و تعرق و محاسبه فرمول‌های فراوان و نسبتاً پیچیده استوار است. در اینجا به دلیل در دسترس نبودن برخی آمار، از روش شرح داده شده در ادامه، استفاده می‌شود.

با تحلیل نقشه پهنه‌بندی خشکسالی که با استفاده از شاخص بارش استاندارد[۵] برای دو سال اخیر (۹۰-۹۱) توسط پژوهشکده اقلیم‌شناسی ایران تهیه شده است [۱۸]، نقشه میانگین بارندگی سالیانه ایران که توسط موسسه تحقیقات و مطالعات علوم اجتماعی اداره کل هواشناسی در سال ۱۳۹۰ تهیه شده [۱۳] و میانگین ارتفاع بارش سالانه استان‌های کشور در سال ۱۳۹۰ و در دهه اخیر که توسط اداره کل آمارو فن‌آوری اطلاعات و اطلاع‌رسانی سازمان هواشناسی کشور تهیه و در سالنامه آماری کشور چاپ شده است [۷]، رتبه‌بندی شکل ۴ را برای خشکسالی استان‌ها خواهیم داشت.



شکل ۴- رتبه‌بندی استان‌ها براساس شاخص خشکسالی

شاخص بارش استاندارد براساس تفاوت بارش از میانگین برای یک مقیاس زمانی مشخص و سپس تقسیم آن بر انحراف معیار به دست می‌آید و تنها فاکتور مؤثر در محاسبه این شاخص عنصر بارندگی می‌باشد. براساس این شاخص می‌توان آستانه‌ی خشکسالی را برای هر دوره‌ی زمانی تعیین کرد.

برای این رتبه‌بندی از رابطه (۶) استفاده شده است:

$$score = \frac{A_i}{\max\{A_i\}} + \frac{B_i}{\max\{B_i\}} \quad (۶)$$

که در آن i اندیس استان، A_i شاخص بارندگی استان و B_i شاخص خشکسالی استان است که به ترتیب توسط رابطه (۷) و (۸) حساب می‌شوند. تقسیم شاخص‌ها به مقدار بیشینه‌شان به منظور نگاشت به بازه ۰ و ۱ می‌باشد تا عمل جمع دو شاخص منطقی باشد.

$$A = a * 0.9 + b * 0.1 \quad (۷)$$

$$B = c * 1 + d * 2 + e * 3 + f * 4 + g * 5 + h * 6 + k * 7 + m * 8 + n * 9 \quad (۸)$$

a میانگین بارندگی بلند مدت استان و b میزان بارندگی در سال ۱۳۹۰ برای استان می‌باشد. همچنین c درصد مساحت با ترسالی حاد، d درصد مساحت با ترسالی شدید، e درصد مساحت با ترسالی متوسط، f درصد مساحت با ترسالی ضعیف، g درصد مساحت نرمال، h درصد مساحت با خشکسالی ضعیف، k درصد مساحت با خشکسالی متوسط، m درصد مساحت با خشکسالی شدید و n درصد مساحت با خشکسالی حاد می‌باشد.

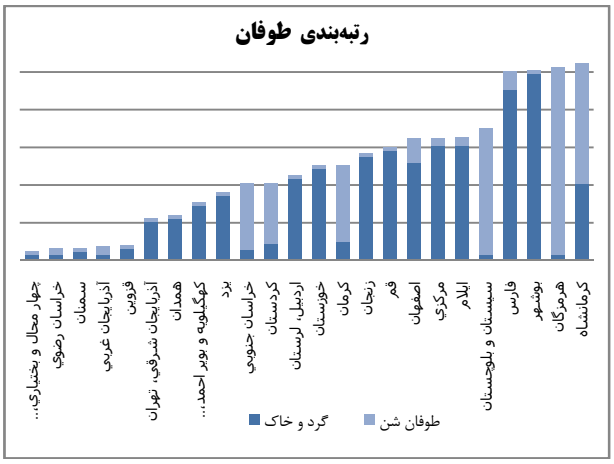
۳-۳- طوفان

این بخش با توجه به موضوع پژوهش تنها شامل طوفان شن و طوفان گرد و خاک می‌باشد که هر دو می‌توانند در طول زمان آسیب‌های جدی به مراکز داده وارد آورند.

گرد و غبار از جمله پدیده‌های طبیعی است که بیشتر در ماه‌های گرم سال به وقوع می‌پیوندد و مناطق وسیعی را در بر می‌گیرد. شناسایی و پهنه‌بندی درجه خطرپذیری گرد و غبار و فراوانی وقوع آن در مناطق مختلف کشور با استفاده از داده‌های ماهواره‌ای مودیس (غلظت آئروسل و شاخص‌های ضخامت نوری آئروسل) انجام می‌گردد.

برخلاف طوفان شن که عمدتاً در مناطق شرقی کشور است، بیشترین وقوع گرد و خاک مربوط به استان‌های غربی می‌باشد که در سال‌های اخیر رشد فزاینده‌ای داشته است.

با استفاده از نقشه پهنه‌بندی تعداد روزهای همراه با گرد و خاک و نقشه پهنه‌بندی تعداد روزهای همراه با طوفان شن در سال ۲۰۱۲ [۱۲] رتبه‌بندی شهرها به صورت شکل ۵ خواهد بود:



شکل ۵- نمودار رتبه‌بندی استان‌ها براساس وقوع طوفان

برای این رتبه‌بندی از رابطه (۹) استفاده شده است:

$$\text{score} = \frac{A_i}{\max\{A_i\}} + \frac{B_i}{\max\{B_i\}} \quad (۹)$$

که در آن i اندیس استان، A_i شاخص طوفان شن و B_i شاخص طوفان گرد و خاک استان (طی سال ۲۰۱۲) است که به ترتیب توسط رابطه‌های (۱۰) و (۱۱) حساب می‌شوند.

$$A = a*1 + b*31 + c*60 + d*88 + e*117 + f*146 \quad (۱۰)$$

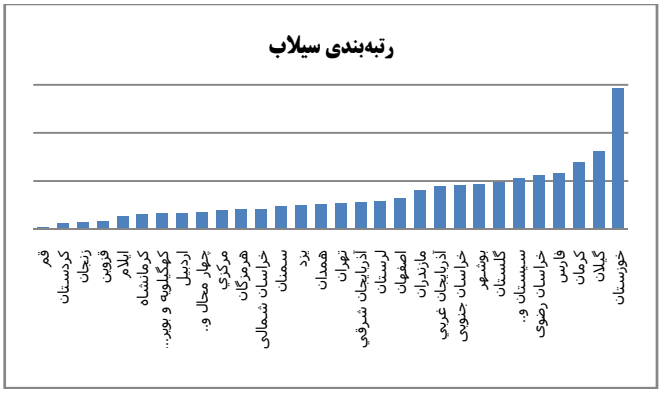
$$B = g*1 + h*20 + i*38 + j*57 + k*75 + m*93 \quad (۱۱)$$

که در آن a، b، c، d، e و f درصد مساحت استان به‌ترتیب شامل ۱ تا ۳۰، ۳۱ تا ۵۹، ۶۰ تا ۸۷، ۸۸ تا ۱۱۶، ۱۱۷ تا ۱۴۵ و ۱۴۶ تا ۱۷۰ روز همراه با طوفان شن و g، h ، i، j، k و m درصد مساحت استان به ترتیب شامل ۱ تا ۱۹، ۲۰ تا ۳۷، ۳۸ تا ۵۶، ۵۷ تا ۷۴، ۷۵ تا ۹۲ و ۹۳ تا ۱۱۰ روز همراه با طوفان گرد و خاک می‌باشد. ضرایب استفاده شده برابر با حداقل تعداد روزهای طوفانی برای یک رده است.

۳-۴- سیل

سیل نیز به همراه زلزله و خشکسالی از مخرب‌ترین بلای طبیعی در ایران است. شدت سیل‌خیزی در نقاط مختلف کشور یا به عبارت دیگر در حوضه‌های آبریز مختلف، با توجه به شرایط اقلیمی و عوامل دیگر مانند پوشش گیاهی از نقطه‌ای به نقطه دیگر متفاوت می‌باشد. این نوع خطر در مورد مرکز داده تنها زمانی می‌تواند مدنظر قرار گیرد که مرکز داده در اطراف یک رودخانه قرار داشته باشد.

با استفاده از نقشه پهنه‌بندی وقوع سیل در ایران در سال ۲۰۱۲ میلادی [۱۲] و ارزیابی سیلاب‌های ایران برای دوره ۲۵ ساله‌ی ۱۳۵۰ تا ۱۳۷۵ [۳]، رتبه‌بندی نهایی شکل ۶ را برای سیل‌خیز بودن استان‌ها خواهیم داشت:



شکل ۶- نمودار مربوط به رتبه‌بندی استان‌ها براساس سیل‌خیز بودن

در مورد رتبه‌بندی فوق به صورت رابطه (۱۲) عمل شده است.

$$\text{score} = \frac{a*25+b}{26} * c \quad (۱۲)$$

که در آن a تعداد سیلاب‌های دوره‌ی ۲۵ ساله، b تعداد سیلاب‌های سال ۲۰۱۲ و c شدت سیلاب است که توسط رابطه (۱۳) محاسبه شده است:

$$c = \begin{Bmatrix} \text{بسیارشدید} = 1.9, \text{شدید} = 1.6, \\ \text{تاحدی‌شدید} = 1.3, \text{معمولی} = 1 \end{Bmatrix} \quad (۱۳)$$

تعداد سیل‌های به وقوع پیوسته برای خراسان شمالی، جنوبی و رضوی بر حسب تعداد سیل‌های به وقوع پیوسته در سال ۲۰۱۲ برای دوره ۲۵ ساله با مقادیری به ترتیب، ۱۷٪، ۴۵٪ و ۳۸٪ تقریب زده شده است.

۳-۵- آتشفشان

آتشفشان زمانی خطرناک است که فعال یا نیمه فعال و نزدیک به شهر باشد. یک آتشفشان زمانی آتشفشان فعال نامیده می‌شود که به تناوب فوران کند و یا علایم فوران را نشان دهد. آتشفشان نیمه فعال، آتشفشانی است که در گذشته فعال بوده ولی در زمان حاضر به اصطلاح خواب می‌باشد. آتشفشان‌های نیمه فعال می‌توانند دوباره فوران کنند. این آتشفشان‌ها اغلب زلزله‌های کوچکی دارند.

هرچند هیچ سازمانی مرکز داده خود را در کنار آتشفشان برپا نمی‌سازد ولی می‌توان دوری از آتشفشان و تأثیرات محیطی آن را به عنوان یک عامل در نظر گرفت. تأثیراتی که یک آتشفشان بر هوای محیط اطراف خود می‌گذارد، به‌خصوص خاکستری که کیلومترها در محیط پخش می‌کند، می‌تواند به تجهیزات مرکز داده و سیستم تهویه مطبوع آن آسیب وارد کند.

آتشفشان‌های دماوند در مازندران، سبلان در اردبیل، تفتان و بزمان در سیستان و بلوچستان، سهند در آذربایجان شرقی، قلعه حسن علی در کرمان، به همراه آتشفشان بدون نام دیگری در سیستان و بلوچستان و آتشفشان آرارات در مرز ایران با کشور آذربایجان و ارمنستان (که مجاورت آذربایجان غربی محسوب می‌شود) از آتشفشان‌های مهم ایران هستند [۳۰].

آتشفشان‌های بزمان، تفتان و دماوند جزء آتشفشان‌های فعال محسوب می‌شوند. برای هر آتشفشان باید محدوده زیر پوشش آن نیز مدنظر قرار گیرد؛ مانند تهران که در مجاورت کوه دماوند می‌باشند هرچند که این کوه در استان مازندران واقع می‌باشد. به طور معمول تا شعاع ۳۰ کیلومتری آتشفشان محدوده خطر به شمار می‌رود.

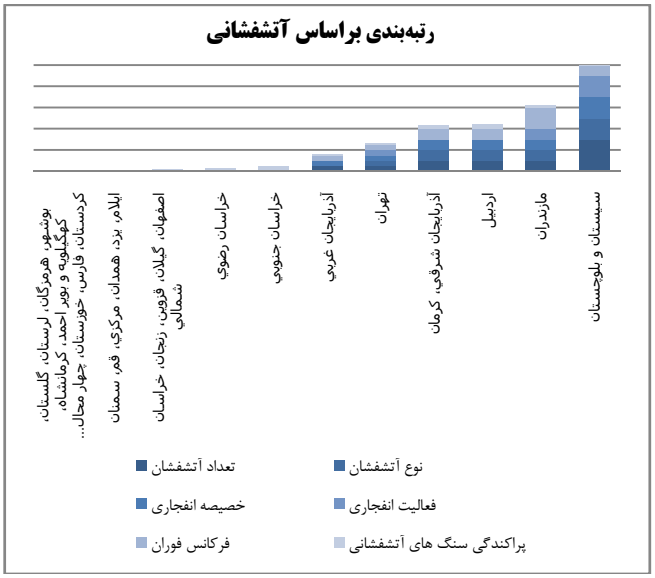
سیستم ملی هشداردهنده زود هنگام آتشفشان[۶] (NVEWS) به منظور ارزیابی خطر آتشفشان در ایالات‌متحده با تجزیه و تحلیل و ارزیابی عوامل متعددی که در وقوع آتشفشان مطرح است ساخته شده است. سیستم NVEWS فهرستی از معیارهای خطر ارائه می‌دهد که هر معیار در صورت وجود در یک شهر یک نمره محسوب می‌شود [۲۱]. این معیارها شامل معیارهای متفاوتی، از جمله معیارهای انسانی و تجاری می‌باشد که نمی‌توان از همه آن‌ها برای هدف این مقاله استفاده کرد. هر کدام از معیارها نیز در به دو صورت نمره محافظه کارانه[۷] و نمره حداکثر[۸] نمره‌دهی می‌شوند [۲۱]. در این بخش سعی شده است معیارهای اصلی با تمرکز بر ساخت مرکز داده تا حد ممکن سنجیده شده و یا از معیارهای مشابه استفاده شود و خطر آتشفشان برای استان‌های کشور با الهام از روش NVEWS ارزیابی شود. ضمناً برای ارزیابی از نمره حداکثر استفاده می‌شود. نتیجه این رتبه‌بندی در شکل ۷ آمده است.

در نمودار فوق رتبه‌بندی با استفاده از رابطه (۱۴) به دست آمده است:

$$\text{score} = n + \text{type} + VEI + EA + ER + S \quad (۱۴)$$

که در آن score درجه خطر آتشفشان در استان، n تعداد آتشفشان‌های استان، type نوع آتشفشان مانند استراتوولکان[۹] یا مارس[۱۰] (آتشفشان قوی=۱،

آتشفشان ضعیف=۰)، VEI شاخص عمومی خصیصه انفجاری فوران (بین ۳ تا ۴ نمره ۱، بین ۵ تا ۶ نمره ۲، عدم وجود اطلاعات نمره type)، EA فعالیت انفجاری (فعالیت در ۵۰۰ سال گذشته نمره ۱، عدم فعالیت در ۵۰۰ سال گذشته نمره ۰) که با فعال بودن آتشفشان جایگزین شده است، ER فرکانس فوران که با آخرین فوران جایگزین شده است (بین ۱ تا ۹۹ سال نمره ۴، بین ۱۰۰ تا ۱۰۰۰ سال نمره ۳، بین ۱۰۰۰ تا ۱۰۰۰۰ سال نمره ۲، دوره هلوسن نمره ۱) و S پراکندگی سنگ‌های آتشفشانی در یک استان است که به عنوان یک معیار برای آن استان به کار برده‌ایم و گویای فوران آتشفشان‌های آن استان در گذشته می‌باشد و براساس چگالی پراکندگی نشان داده شده در نقشه پهنه‌بندی سنگ‌های آذرین [۱۷]مقداری بین ۰ تا ۰.۵ به آن اختصاص داده شده است. تهران نیز به دلیل مجاورت با کوه دماوند و آذربایجان غربی به دلیل مجاورت با کوه آرارات نصف امتیازات معیارهای مربوط به این آتشفشان‌ها را شامل شده‌اند.



شکل ۷- نمودار رتبه‌بندی بر حسب خطر آتشفشان

۴- امکانات

یک شهر یا استان با در اختیار داشتن امکانات مناسب می‌تواند راه هموارتری برای برپایی یک مرکز داده در خود داشته باشد. در این بخش به معیارهای بستر ارتباطی، تأمین انرژی، نیروی انسانی تحصیل‌کرده و دیگر امکانات شهر تحت شاخص توسعه‌یافتگی پرداخته می‌شود.

۴-۱- بستر ارتباطی

یکی از مهم‌ترین معیارها برای برپایی یک مرکز داده در یک استان، توانایی استان در ارائه خدمات ICT و آمادگی الکترونیکی آن استان می‌باشد. روش‌های بسیاری برای ارزیابی فناوری اطلاعات یک شهر یا کشور وجود دارد که هر کدام نشانگر[۱۱]ها و دسته‌بندی‌های خود را دارند که شاخص[۱۲] نامیده می‌شود و برای آن‌ها اهمیت‌های گوناگونی مطرح کرده‌اند.

سه روش اصلی برای این ارزیابی‌ها عبارت‌اند از روش موسسه ASPA[۱۳]، ارزیابی آمادگی الکترونیکی توسط EIU[۱۴] و بررسی شاخص‌های ارزیابی توسعه ICT توسط UNCTAD[۱۵] که توضیح عملکرد این روش‌ها از حوصله این مقاله خارج است. برخی از شاخص‌های مورد استفاده این روش‌ها عبارت‌اند از ضریب نفوذ اینترنت، ضریب نفوذ تلفن همراه، ضریب نفوذ تلفن ثابت، ضریب نفوذ رایانه، ضریب نفوذ پهنای باند، میزان رشد کسب و کار الکترونیک، درآمد سرانه، قدرت اقتصادی، پایداری وضعیت سیاسی، سواد عمومی و سواد الکترونیکی، تعداد متخصصان ICT، سرویس‌های ICT، ترافیک سرویس‌های ارتباطی [۴].

تمامی شاخص‌های فوق برای برپایی یک مرکز داده نقش خواهد داشت و در این مقاله بسیاری از آن‌ها در قسمت‌های گوناگون بررسی شده‌اند و برای بخش بستر ارتباطی این مقاله شاخص‌های همچون ترافیک سرویس‌های ارتباطی، ضریب نفوذ تلفن (نرخ تعداد تلفن‌های استان به جمعیت استان) و ضریب نفوذ اینترنت بررسی می‌شوند که از شاخص‌های مهم در ارزیابی فناوری اطلاعات می‌باشند.

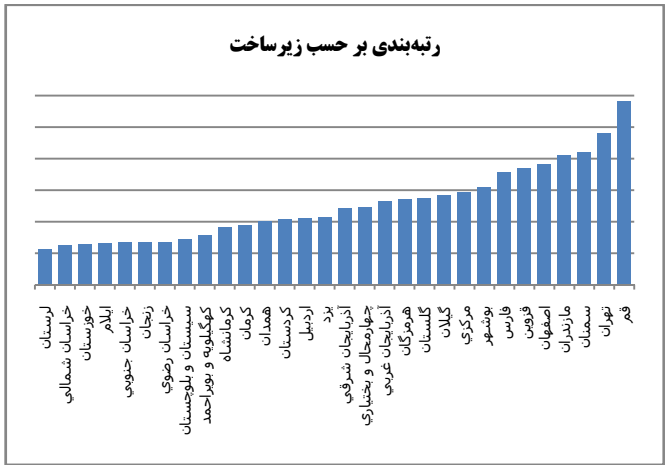
روش UNCTAD براساس استانداردهای ITU می‌باشد و روشی برگزیده از میان روش‌های موجود می‌باشد. از این روش به عنوان الگوی اصلی در انتخاب قالب برای ارزیابی وضعیت فن‌آوری اطلاعات در کشور استفاده شده است [۴]. شاخص‌های این روش به چهار گروه کلی اتصال، دسترسی، سیاست، و ترافیک تقسیم شده‌اند. در هر گروه مجموعه‌ای از شاخص‌ها قرار دارد. در گروه اتصال شاخص‌های نرخ نفوذ تلفن ثابت و تلفن همراه در نظر گرفته شده است. در گروه دسترسی شاخص‌های نرخ نفوذ اینترنت، سطح سواد و سطح درآمد هر استان مد نظر هستند. در گروه سیاست تأمین‌کنندگان محتوای اینترنتی و متقاضیان مرکز داده اینترنتی هر استان در نظر گرفته شده است. در گروه ترافیک نیز ترافیک شبکه دیتا کشور هر استان مدنظر قرار گرفته است. با توجه به اینکه در این مقاله بسیاری از این شاخص‌ها جداگانه بررسی می‌شوند و نیز آمار برخی دیگر از شاخص‌ها در دسترس نیست، شاخص‌های منتخب طبق جدول ۲ انتخاب شده‌اند.

در گروه دسترسی، سطح سواد در بخش‌های بعدی این مقاله اندازه‌گیری شده است و از آن در این بخش صرف‌نظر خواهد شد. از گروه سیاست، سطح درآمد در گروه دسترسی و میزبان‌های وب در گروه اتصال به دلیل اهمیت کم در موضوع این بخش چشم‌پوشی می‌شود. این دسته‌ها هر کدام ضرایب تأثیر متفاوتی دارند که با کمک از نمایه ITU سال ۲۰۱۲ انتخاب شده است [۲۷]. داده‌های نحوه استفاده از اینترنت نرمال شده هستند و مقادیر به‌کار رفته، در واقع ضریب نفوذ هر حالت (با درصدی خطا) می‌باشد.

جدول ۲- ضرایب تأثیر و وزن استفاده شده برای معیار زیرساخت ارتباطی

گروه	شاخص		ایده آل	وزن مؤثر	وزن کل
دسترسی	ضریب نفوذ تلفن ثابت		۶۰	٪۳۳.۳	٪۴۰
	ضریب نفوذ تلفن همراه		۱۸۰	٪۳۳.۳	
	ضریب نفوذ رایانه شخصی		۱۰۰	٪۳۳.۳	
استفاده	ضریب نفوذ اینترنت		۱۰۰	٪۵۰	٪۴۰
	نحوه استفاده از اینترنت	Dialup	۱۰۰	٪۱۶.۶	
		ADSL	۴۰	٪۱۶.۶	
		Wireless	۲۰	٪۱۶.۶	
ترافیک	میانگین ترافیک داده‌ای		۲۵.۴	٪۱۰۰	٪۲۰

در نهایت با استفاده از ضریب نفوذ رایانه شخصی جمعیت شهری [۶]، ضریب نفوذ تلفن ثابت و ضریب نفوذ تلفن همراه [۲]، ضریب نفوذ اینترنت و نحوه دسترسی به اینترنت در محل سکونت [۶] و نرخ میانگین ترافیک ورودی و خروجی داده استان‌ها براساس داده های به دست آمده از شرکت ارتباطات زیرساخت [۴] رتبه‌بندی استان‌ها به صورت شکل ۸ خواهد بود.



شکل ۸- نمودار رتبه‌بندی بر حسب زیرساخت ارتباطی

رتبه‌بندی فوق به صورت رابطه (۱۵) محاسبه شده است:

$$score = ((L * .40) + (M * .40) + (N * .20)) * 10 \quad (۱۵)$$

و خواهیم داشت:

$$L = \frac{a}{60} * 0.33 + \frac{b}{180} * 0.33 + \frac{c}{100} * 0.33 \quad (۱۶)$$

$$M = \frac{d}{100} * 0.5 + \frac{e}{100} * 0.16 + \frac{f}{40} * 0.16 + \frac{g}{20} * 0.16 \quad (۱۷)$$

$$N = \frac{h}{25.4} * 1.0 \quad (۱۸)$$

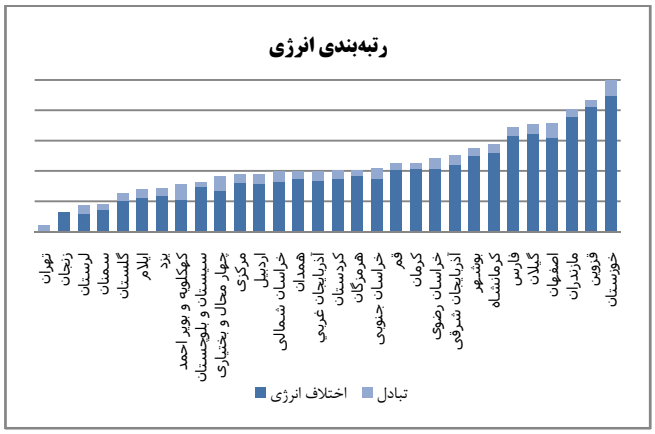
که در آن a ضریب نفوذ تلفن ثابت، b ضریب نفوذ تلفن همراه، c ضریب نفوذ رایانه شخصی، d ضریب نفوذ اینترنت، e درصد استفاده از اینترنت به صورت Dialup، f درصد استفاده از اینترنت به صورت ADSL، g درصد استفاده از اینترنت به صورت Wireless، h میانگین ترافیک داده‌ای می‌باشد و ۲۵.۴ میانگین نرخ تبادلی داده در کشور می‌باشد.

۴-۲- تأمین انرژی

هزینه‌های انرژی الکتریسیته درصد بالایی از هزینه‌های عملیاتی یک مرکز داده را تشکیل می‌دهند و هدف یک مرکز داده سبز کاهش این نوع هزینه و جایگزینی آن می‌باشد. ولی استفاده از این نوع انرژی در یک مرکز داده اجتناب‌ناپذیر است و بنابراین هنگامی‌که مکان‌های مختلف برای یک مرکز داده بررسی می‌شود باید هزینه برق آن منطقه و از آن مهم‌تر دسترسی آسان و کافی به برق در آن منطقه نیز حساب شود. کمبود ظرفیت تولیدی نیروگاه‌های یک منطقه، اثرات نامطلوب خاموشی برق را در پی دارد که گاه ضربات جبران‌ناپذیری به مرکز داده و تجارت‌های مبتنی بر آن وارد می‌کند.

تولید و توزیع برق در ایران زیر نظر سازمان مدیریت تولید و انتقال نیروی برق (توانیر) است که وظیفه توسعه تأسیسات تولید، انتقال و عمده فروشی برق را بر عهده دارد. این سازمان به عنوان شرکت مادر تخصصی توانیر شامل چندین شرکت برق منطقه‌ای می‌باشد که هر کدام مسئولیت مدیریت تولید و تأمین انرژی الکتریکی مطمئن و پایدار در منطقه تحت پوشش جغرافیایی خود را عهده‌دار هستند. با بررسی تعرفه‌های مناطق مختلف شرکت توانیر مشخص شد که تعرفه‌های برق برای مناطق جغرافیایی گوناگون یکسان بوده و تنها برحسب نوع مصرف کننده، ساعت مصرف یا شرایط خاص هزینه‌های متفاوتی دریافت می‌کند [۱۱].

با استفاده از خلاصه وضعیت تولید و توزیع برق نیروگاه‌های استان‌ها (شامل بخش خصوصی) تا پایان سال ۱۳۹۰ [۹، ۱۰] رتبه‌بندی شهرها بر حسب انرژی به صورت شکل ۹ خواهد بود.



شکل ۹- نمودار رتبه‌بندی بر حسب انرژی الکتریسیته

رتبه‌بندی فوق طبق رابطه (۱۹) صورت گرفته است:

$$score = A * 0.9 + B * 0.1 \quad (۱۹)$$

که در آن A شاخص اختلاف انرژی و B شاخص مبادله در پیک بار می‌باشد. مقدار A که اختلاف درصد تولید و مصرف برق استان در کشور است به صورت رابطه (۲۰) حساب می‌شود.

$$A = \frac{a - b + \epsilon}{C} * 1000 \quad (۲۰)$$

که در آن a مقدار تولید برق استان، b مقدار مصرف برق استان، ϵ مقدار ثابتی برای غیرمنفی کردن مقادیر به دلیل سازگاری با پارامترهای دیگر و C حداکثر اختلاف تولید و مصرف انرژی (به‌علاوه مقدار ϵ) است (رابطه ۲۱).

$$C = \max_{i=\text{استانها}}(a_i - b_i + \epsilon) \quad (۲۱)$$

شاخص B یا درصد تبادل انرژی نیز به صورت رابطه (۲۲) حساب می‌شود.

$$B = \frac{-\frac{m}{n} + \sigma}{P} * 1000 \quad (۲۲)$$

که در آن m مقدار تبادل انرژی در ساعت پیک بار منطقه، n تعداد استان‌های منطقه، σ مقدار ثابتی برای غیرمنفی کردن مقادیر و P حداکثر تبادل انرژی (قرض داده شده به همراه مقدار σ) می‌باشد (رابطه ۲۳).

$$P = \max_{i=\text{استانها}}(-\frac{m_i}{n_i} + \sigma) \quad (۲۳)$$

علامت منفی در رابطه‌های فوق به منظور مثبت کردن عملکرد قرض دادن انرژی می‌باشد.

برق منطقه‌ای تهران با اینکه ظرفیت تولیدی بسیار بالایی دارد ولی از آنجایی‌که مصرف بالایی (بیش از تولید خود) دارد در آخرین جایگاه قرار دارد.

ناگفته نماند در صورت استفاده از انرژی خورشیدی برای تأمین برق و یا استفاده از ژنراتور برق می‌توان این آمار را نادیده گرفت ولی از آنجایی‌که استفاده از برق شهری مقرون به صرفه‌تر خواهد بود، این اطلاعات در یافتن مکان یک مرکز داده مفید خواهند بود.

۴-۳- منابع انسانی

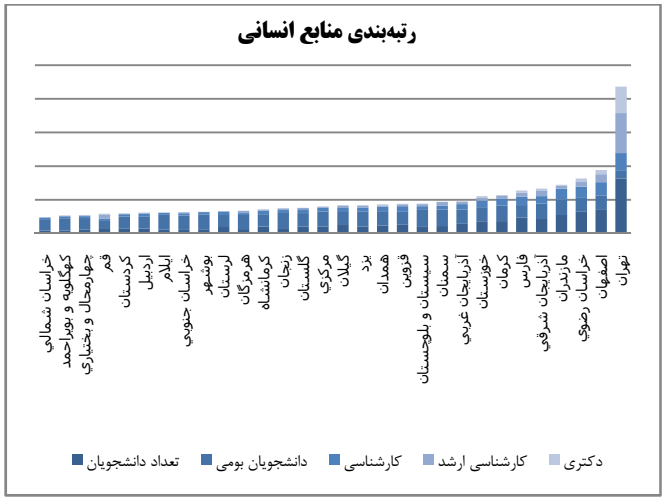
برای برپایی یک مرکز داده به نیروی انسانی تحصیل‌کرده و کاردان برای کار در مرکز داده نیاز است. بنابراین شهری که مرکز داده در آن قرار می‌گیرد باید درجه علمی مناسبی داشته باشد و این مقیاس با توجه به دانشگاه‌ها و مراکز علمی یک شهر سنجیده می‌شود.

برای رتبه‌بندی درجه علمی استان‌ها از آمار تعداد دانشجویان دانشگاه‌ها و مراکز آموزش عالی کشور در سال ۹۰-۹۱ و تعداد پذیرفته‌شدگان به تفکیک درجه تحصیلی در سال ۸۹-۹۰ و درصد دانشجویان بومی [۵، ۷، ۸] استفاده شده است. آمار نشان‌دهنده این است که رابطه مستقیمی بین تعداد پذیرفته‌شدگان کل و پذیرفته‌شدگان با درجات فوق‌لیسانس و دکتری نیز برقرار است. معیارهای دیگری نیز برای این عامل از جمله تعداد دانشجویان بومی هر استان نیز لحاظ شده است، زیرا به جز برخی از کلان‌شهرها مانند تهران دانشجویان غیربومی تمایلی برای سکونت در شهر محل تحصیل خود ندارند و این مورد آمار فوق را دستخوش تغییر می‌نماید.

برای ارزیابی توان نیروی انسانی استان‌ها با استفاده از آمار فوق طبق رابطه (۲۴) عمل می‌کنیم.

$$score = a * 0.15 + b * 0.10 + c * 0.05 + d * 0.30 + e * 0.40 \quad (۲۴)$$

که در آن a درصد پذیرفته‌شدگان کارشناسی، b درصد پذیرفته‌شدگان کارشناسی‌ارشد، c درصد پذیرفته‌شدگان دکتری، d درصد دانشجویان استان به کل دانشجویان و e درصد دانشجویان بومی آن استان می‌باشد. در نهایت رتبه‌بندی شکل ۱۰ را برای استان‌ها خواهیم داشت.



شکل ۱۰- نمودار رتبه‌بندی براساس منابع انسانی تحصیل کرده

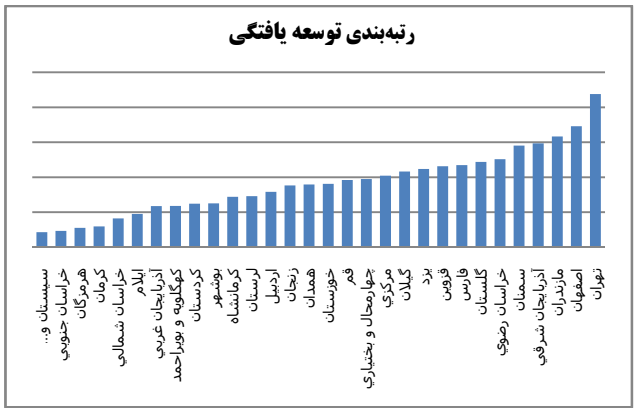
البته آماری همچون درصد فارغ‌التحصیلان بدون شغل و درصد متخصصان IT جویای شغل نیز می‌تواند مفید واقع شود که متأسفانه این آمار در دسترس نیست.

۴-۴- توسعه‌یافتگی

علاوه بر عوامل فوق در انتخاب یک شهر (یا استان)، عوامل دیگری نیز مانند طول بزرگراه‌ها، تعداد فرودگاه‌ها، طول خطوط راه‌آهن و غیره وجود دارند که توجه به آن‌ها مناسب است. این عوامل برای امنیت و کاهش خطرات مربوط به جنبه‌های عملیاتی یک مرکز داده اهمیت دارد. در این بخش برای بررسی این عوامل از سطح توسعه‌یافتگی استان‌های کشور استفاده شده که در آن مناطق توسعه‌یافته و محروم مشخص می‌شوند. هرچه سطح توسعه‌یافتگی شهر بیشتر باشد پتانسیل بیشتری برای پذیرش یک مرکز داده خواهد داشت زیرا دسترسی به خدمات ساده‌تر و ارزان‌تر خواهد بود.

در [۱] ۴۰ شاخص مختلف شامل شاخص اقتصادی (درصد اشتغال، درصد کارگاه‌ها و غیره)، آموزشی (درصد باسوادی، تعداد مدارس، تعداد کتابخانه، ...)، درمانی (تعداد مؤسسات درمانی، تعداد پزشک، ...) و زیربنایی (طول بزرگراه‌ها، طول راه‌های آسفالته، گاز لوله‌کشی، ...) با استفاده از چهار روش تاسیس، تاکسونومی عددی، موریس و شاخص‌بندی ترکیبی ارزیابی شده است. در این بخش از این رتبه‌بندی موجود با انتخاب روش تاپسیس[۱۶] که یکی از روش‌های مرسوم و پرکاربرد در میان روش‌های تصمیم‌گیری چند شاخصه است و برمبنای محاسبه فاصله گزینه‌ها از راه‌حل ایده‌آل مثبت و ایده‌آل منفی استوار است، استفاده شده است. گزینه برتر کمترین فاصله را از راه‌حل ایده‌آل مثبت و بیشترین فاصله را از راه‌حل ایده‌آل منفی خواهد داشت.

براساس [۱] رتبه‌بندی شهرها از نقطه نظر توسعه‌یافتگی به صورت شکل ۱۱ است.



شکل ۱۱- نمودار رتبه‌بندی استان‌ها براساس شاخص توسعه‌یافتگی

۵- حوادث غیرطبیعی

علاوه بر حوادثی که به دست طبیعت رخ می‌دهد و می‌تواند باعث اخلال کار یک مرکز داده شود، حوادث غیرطبیعی که به دست بشر رخ می‌دهد نیز می‌تواند باعث اخلال کار مرکز داده شود. این حوادث عمدتاً شامل وقوع جنگ یا عملیات تروریستی است. در این بخش شهرهای کشور از نقطه نظر آسیب‌پذیری در برابر این حوادث بررسی خواهند شد. البته فاصله شهر از انبارهای مهمات و همچنین بیماری‌های همه‌گیر نیز می‌توانند باعث اخلال کار یک مرکز داده شوند. انفجار زاغه مهمات سپاه در سال ۱۳۹۰ در حاشیه تهران یا انفجار پادگان امام علی خرم‌آباد که

موج آن تا چند ده کیلومتر گزارش شده و باعث کشته شدن چندین نفر شد، نمونه‌ای از حوادثی است که در صورت نزدیکی به یک مرکز داده ممکن است باعث اخلال در کار مرکز داده شود. اما این معیار در اینجا به دلیل مسائل امنیت ملی و عدم دسترسی به آمار دقیق نادیده گرفته خواهد شد.

۵-۱- تروریسم

در دهه‌های اخیر تروریسم در جهان افزایش پیدا کرده است و یکی از عوامل ایجاد خطر است. در ایران در دهه‌های پیش از انقلاب اسلامی شرایطی به وجود آمد که به گسترش رویکردهای تروریستی در بخشی از گروه‌های سیاسی کمک کرد. پس از انقلاب نیز نه تنها این پدیده تداوم یافت، بلکه به دلایل تازه تأسیس بودن حکومت انقلابی- مردمی و عدم‌اشراف آن بر تمام تحولات کشور، گسترش پیدا کرد.

شهرهای بزرگ و پرجمعیت در هر کشور به‌عنوان اهداف عملیات تروریستی قرار می‌گیرند. البته عوامل دیگری از جمله نزدیکی به تأسیسات هسته‌ای و مرزی بودن شهر نیز وجود دارند [۲۳]. هنگام در نظر گرفتن این عامل، داشتن امکانات امنیتی و گروه‌های نظامی در مقابله با تروریسم برای یک شهر ضروری به نظر می‌رسد. بیشتر مراکز داده در مکان‌هایی ساخته می‌شوند که با معابر عمومی فاصله چندانی ندارند و یا در ساختمان‌های چندمنظوره با پارکینگ‌های عمومی قرار دارند. به دلایل امنیتی یک مرکز داده باید در ساختمانی مستقل و با فاصله‌ای مناسب از اماکن عمومی قرار گیرد.

روش‌ها و مدل‌هایی برای رتبه‌بندی شهرها بر حسب احتمال عملیات تروریستی وجود دارد [۲۳، ۲۹] که از فاکتورهای متعددی برای رتبه‌بندی استفاده می‌کنند.

در این بخش برای رتبه‌بندی این عامل ابتدا گروهک‌های تروریستی ایران و شهرهایی که قرارگاه آنان هستند را مشخص می‌کنیم. گروهک‌های تروریستی کومله در کردستان، پژاک در آذربایجان غربی و کردستان، حزب مردم بلوچستان، سازمان زرمبش، جندالله در سیستان و بلوچستان، سازمان مجاهدین خلق (منافقین) ایران در استان دیاله عراق هم مرز با استان کرمانشاه مهم‌ترین گروهک‌های تروریستی مخالف جمهوری اسلامی ایران هستند. استقرار و مجاورت این گروهک‌ها در برخی استان‌ها باعث می شود این استان‌ها از لیست مکان های مناسب مرکز داده ملی خارج شوند.

ضمناً شهرهای تهران، قم، مشهد، اصفهان، تبریز، خوزستان و شیراز به دلیل اینکه جزء شهرهای پرجمعیت ایران هستند می‌توانند به عنوان هدف عملیات تروریستی قرار گیرند و گزینه‌های مناسبی نمی‌باشند. هرچند با فاصله دادن مرکز داده از اماکن عمومی همانطور که قبلاً گفته شد، می‌توان از برخی حوادث پیشگیری کرد ولی نمی‌توان از این گزینه چشم‌پوشی کرد زیرا که این شهرها هدف هستند و عملیات تروریستی قاعده‌ای برای خود ندارند.

وجود تأسیسات هسته‌ای در یک شهر نیز می‌تواند به عنوان عاملی برای حملات تروریستی باشد و تأسیسات هسته‌ای ایران همواره مورد توجه دشمنان بوده است.

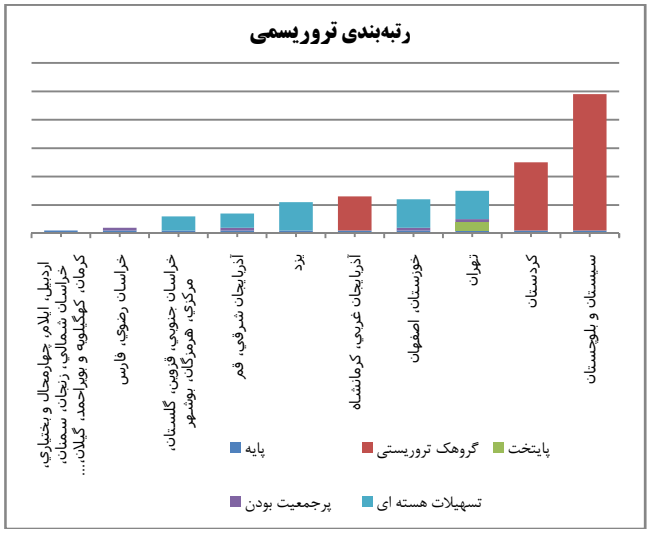
با استفاده از آمار جمع‌آوری شده در مورد تأسیسات هسته‌ای [۱۴، ۲۸] و آمار گروهک‌های تروریستی، در شکل ۱۲ استان‌ها براساس احتمال حملات تروریستی رتبه‌بندی شده‌اند.

برای این ارزیابی از رابطه (۲۵) استفاده شده است.

$$\text{score} = \left(a * 0.6 + b * 0.25 + \frac{c}{100} * 0.0 + d * 0.05 + e * 0.1 + \in\right) \qquad (۲۵)$$

$$\in > 0$$

که در آن a تعداد گروهک تروریستی استان، b تعداد تأسیسات هسته‌ای، c درصد آمادگی نیروی انتظامی استان، d شاخص پرجمعیت بودن، e شاخص پایتخت بودن و ∈ ضریب حمله تروریستی می‌باشد که برای همه استان‌ها مقداری ثابت و غیر صفر در نظر گرفته شده است.



شکل ۱۲- نمودار رتبه‌بندی بر حسب خطر تروریسم

ضرایب تأثیر نیز به صورت جدول ۳ انتخاب شده است.

جدول ۳- ضریب تأثیر شاخص‌های مربوط به تروریسم

شاخص	ایده آل	وزن مؤثر
تعداد گروهک تروریستی	۰	٪۶۰
تعداد تأسیسات هسته‌ای	۰	٪۲۵
جمعیت زیاد	نداشته باشد	٪۵
پایتخت	نباشد	٪۱۰

۵-۲- جنگ

جنگ از دو منظر نافی وجود مرکز داده است. اول اینکه تأثیرات جنگ می‌تواند به طور غیرمستقیم کار یک مرکز داده را مختل کند و دوم اینکه مرکز داده می‌تواند به طور مستقیم به عنوان یک هدف جنگی محسوب شود که در صورت تصرف شدن می‌توان صدمات جبران ناپذیری به امنیت ملی وارد آورد. بنابراین معقول است مرکز داده در مکانی برپا شود که از آثار جنگ به دور باشد.

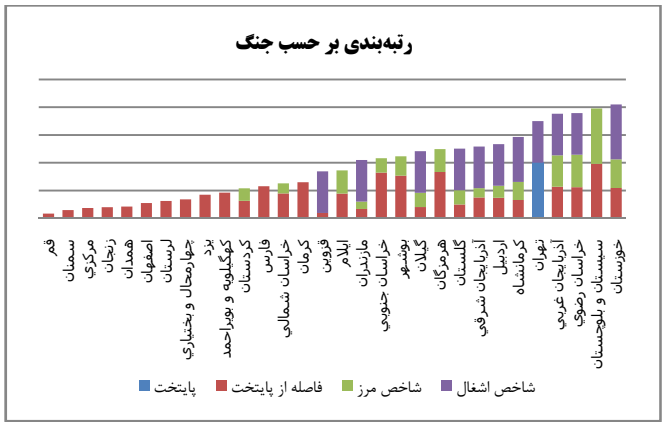
در طول تاریخ ایران همواره مورد هجوم کشورهای مختلف و همسایگان خود بوده است و جنگ تحمیلی بارزترین آن است. مرزهای سیاسی ایران همواره پرتنش و به‌عنوان مهم‌ترین مسئله سرزمینی و حاکمیت ایران محسوب شده است. همسایگی با پانزده کشور و مرزی به طول ۸ هزار کیلومتر و تنوع در نحوه تعامل با هر یک از همسایگان و وضعیت نابسامان آن‌ها می‌تواند پیامدهای امنیتی در سطوح مختلف همچون جنگ به همراه داشته باشد.

در اینجا استان‌های کشور بر حسب آسیب‌پذیری در جنگ رده بندی خواهند شد. برای این کار از معیارهایی همچون طول مرز مشترک با همسایگان، نزدیکی به پایتخت، سابقه اشغال شدن در قرن اخیر، وضعیت کشور همسایه و تعامل با آن‌ها استفاده خواهد شد.

در قرن اخیر بسیاری از شهرهای ایران در جنگ جهانی دوم به مدت ۵ سال (۶۰ ماه) در اشغال متفقین بود. برخی از شهرهای استان خوزستان در جنگ تحمیلی به مدت ۵۷۸ روز (۱۹ ماه) و برخی از شهرهای استان کرمانشاه در حمله منافقین به مدت ۵ روز در اشغال بوده است.

در مورد فاصله از پایتخت باید گفت هرچه شهرها به پایتخت نزدیک‌تر باشند، امن‌تر خواهند بود، هرچند شهرهایی که در مسیر پایتخت قرار دارند، هنگام اشغال مورد تجاوز قرار خواهند گرفت که این مورد در شاخص اشغال گنجانده شده است. خود پایتخت نیز از این قاعده مستثنی است چنانچه در جنگ تحمیلی همواره مورد حملات هوایی بوده است.

با استفاده از آمار جمع آوری شده و شاخص‌ها، رتبه‌بندی شکل ۱۳ را خواهیم داشت.



شکل ۱۳- نمودار رتبه‌بندی استان‌ها بر حسب خطر تروریسم

این رتبه‌بندی براساس رابطه (۲۶) انجام شده است:

$$\text{score} = A * 0.3 + B * 0.4 + C * 0.2 + D * 0.1 \quad (۲۶)$$

که در آن A شاخص مرز استان، B شاخص اشغال، C شاخص فاصله از پایتخت و D شاخص پایتخت می‌باشد.

جدول ۴- شاخص وضعیت کشورها

کشور	وزن شاخص
جمهوری آذربایجان	۱.۱
ارمنستان	۱
ترکیه	۱
عراق	۱.۵
پاکستان	۱.۱
افغانستان	۱.۲
ترکمنستان	۱
دریای خزر	۱.۲
دریای عمان	۱.۳
خلیج فارس	۱.۵

A، B، C و D طبق رابطه‌های (۲۷) تا (۳۰) محاسبه می‌شوند:

$$A = \frac{2a*e_a+b*e_b}{\max\{2a*e_a+b*e_b\}} \quad (۲۷)$$

$$B = c + d + k \quad (۲۸)$$

$$C = \frac{f}{1600} \quad (۲۹)$$

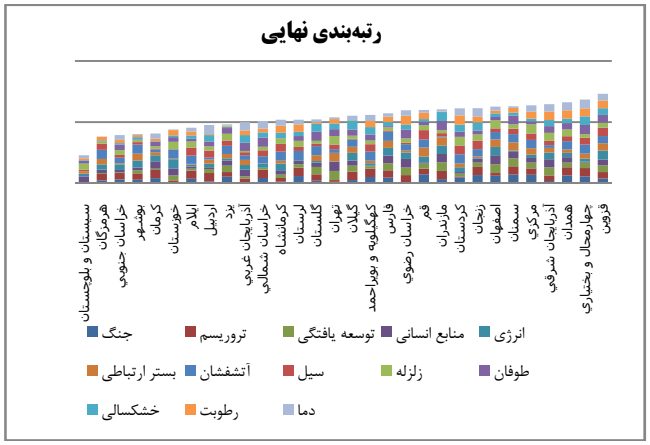
$$D = \{\text{تهران} = 1, \text{دیگر شهرها} = 0\} \quad (۳۰)$$

که در آن a طول مرز خشکی و b طول مرز آبی، e_i شاخص وضعیت کشور همسایه (طبق جدول ۴)، c مدت اشغال (در واحد ماه) در جنگ جهانی دوم، d مدت اشغال (در واحد ماه) در جنگ تحمیلی، k مدت اشغال (در واحد روز) در حمله منافقین به ایران و f فاصله از پایتخت می‌باشد.

۶- سیستم پیشنهادی

پس از ارزیابی جداگانه معیارهای انتخاب مکان یک مرکز داده، برای انتخاب بهترین مکان برای یک مرکز داده سبز نیاز به ادغام معیارها و ارزیابی نهایی است. برای رتبه‌بندی نهایی می‌توان از روش‌های متفاوتی همچون روش‌های تصمیم‌گیری چند شاخصه و یا رأی‌گیری استفاده کرد. در سیستم پیشنهادی از روش رأی‌گیری اکثریت[۱۷] استفاده شده است. برای این کار به هر استان در ازای هر معیار رتبه‌ای بین ۱ تا ۳۰ براساس رتبه‌بندی‌های صورت گرفته در بخش‌های قبلی داده می‌شود. در نهایت هر استان که بیشترین رأی با رتبه بالاتر را داشته باشد، یعنی در اکثر معیارها رتبه بالایی (نه لزوماً بالاترین) کسب کرده باشد، در رتبه‌بندی نهایی توسط سیستم انتخاب خواهد شد.

رتبه‌بندی شکل ۱۴ نتیجه حاصل از سیستم پیشنهادی (بدون وزن) است:



شکل ۱۴- نمودار رتبه‌بندی نهایی براساس تمامی شاخص‌ها

اما این رتبه‌بندی نمی‌تواند دقیق باشد زیرا تمام معیارها دارای تأثیر یکسانی هستند. به عنوان مثال تأثیر آتشفشان با دما یا انرژی یکی نیست.

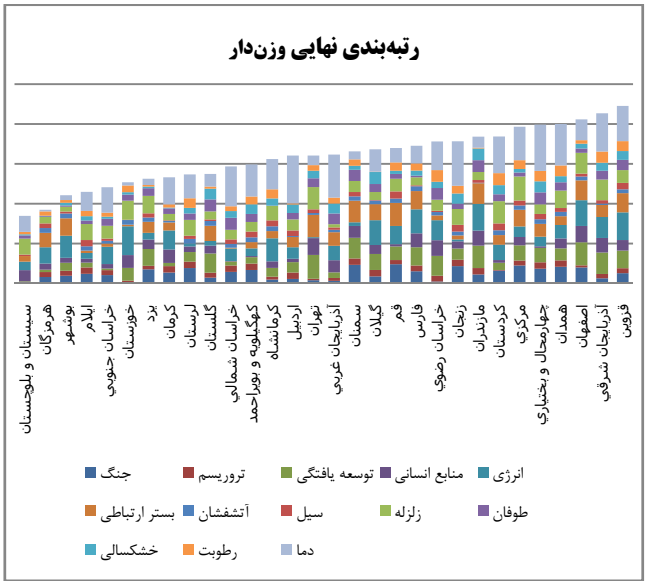
بنابراین برای بالا بردن دقت و کاربردپذیری در روش پیشنهادی، تأثیرگذاری معیارهای مرکز داده توسط متخصصان سنجیده شده و با استفاده از فرآیند تحلیل سلسله مراتبی وزن هر یک از معیارها محاسبه می‌شود. فرآیند تحلیل سلسله مراتبی یک روش مبتنی بر دانش کارشناسی است و وزن‌ها با توجه به اهمیتی که هر یک از کارشناسان برای هر یک از معیارها قائل هستند، به دست می‌آید. در این مقاله، اهمیت معیارها نسبت به یکدیگر براساس توزیع و جمع‌آوری پرسشنامه بین چندی از متخصصین رشته نرم‌افزار و فناوری اطلاعات به دست آمده است. پایایی پرسشنامه براساس ضریب آلفای کرونباخ و با آستانه ۰.۷ سنجیده شده است. برای محاسبه وزن نسبی پس از تحلیل سلسله مراتبی از روش‌های مختلفی استفاده

می‌شود [۲۴] که در این مقاله از روش تقریبی میانگین حسابی برای وزن‌دهی استفاده شده است. وزن نسبی معیارهای محاسبه شده در جدول ۵ نشان داده شده است.

بنابراین برای تشکیل یک مرکز داده سبز، اگر وزن دهی معیارها طبق جدول ۵ صورت پذیرد، سیستم پیشنهادی، رتبه‌بندی شکل ۱۵ را تعیین خواهد کرد.

جدول ۵- وزن پیشنهادی معیارها

دسته	زیر دسته	معیار		ضریب تأثیر
۱	۱	آب و هوا	دما	٪۲۰
	۲		رطوبت	٪۵
۲	۳	حوادث طبیعی	زمین‌لرزه	٪۱۰
	۴		خشکسالی	٪۵
	۵		طوفان	٪۵
	۶		سیل	٪۳
	۷		آتشفشان	٪۲
۳	۸	امکانات	بستر ارتباطی	٪۱۰
	۹		تأمین انرژی	٪۱۲
	۱۰		منابع انسانی	٪۷
	۱۱		توسعه‌یافتگی	٪۱۰
۴	۱۲	حوادث غیرطبیعی	تروریسم	٪۳
	۱۳		جنگ	٪۸



شکل ۱۵- نمودار رتبه‌بندی نهایی به صورت وزنی براساس تمامی شاخص‌ها

در صورت تغییر وزن هر یک از معیارها بر حسب نیاز در سیستم پیشنهادی، یک رتبه‌بندی متفاوت به دست خواهد آمد. بنابراین شناخت نیازهای اساسی و وزن‌دهی مناسب به هر یک از اولویت‌ها یک اصل خواهد بود که باید رعایت شود. به عنوان مثال اگر طرحی برای استفاده از سرمایش رایگان نداریم می‌توان وزن کمتری برای دما تعیین کرد. این رتبه‌بندی وزن‌دار سیستم پیشنهادی برای تمام حالات جایابی برای یک مرکز داده کاربرد خواهد داشت.

در نهایت قابل ذکر است که در نتیجه نهایی، از فاصله یکسانی برای امتیازات شهرها استفاده شده است (یک واحد فاصله بین ۱ تا ۳۰ به ترتیب) و به عنوان یکی از کارهای آتی در این پژوهش می‌توان به کارگیری فاصله مناسب برای امتیازدهی نهایی شهرها را بیان کرد که در آن امتیاز شهرها براساس فاصله معیارها حساب می‌شود که تک تک به ازای هر معیار برای تمام شهرها حساب می‌شود تا دقت بالاتری داشته باشیم و وزن هر معیار تأثیر خود را بهتر نشان دهد.

۷- نتیجه‌گیری

پس از آشنایی با معیارهای انتخاب یک مرکز داده، چگونگی انتخاب بهترین مکان برای یک مرکز داده فرآیندی است که نیاز به تعیین دقیق اهداف مرکز داده توسط مدیران دارد.

تعریف دقیق نیازمندی‌های مرکز داده برای پیشبرد فرآیند انتخاب حیاتی است. هر فرد یا تجارت یا هدفی که مرکز داده برای آن یا براساس آن تأسیس می‌شود، رتبه‌های خاص خود و معیارهای خاص خود را دارد که می‌تواند باعث برتری یک عامل بر عامل دیگر شود و باید مدنظر قرار گیرد.

ارزیابی‌های صورت گرفته در این مقاله بر حسب استان صورت گرفته که دانه‌بندی صحیحی برای مکان برپایی یک مرکز داده نیست و به عنوان یکی از کارهای آتی در این پروژه قصد داریم این معیارها را به صورت دقیق‌تر برای شهرهای کشور به جای استان‌ها انجام دهیم. همچنین در این پژوهش سعی شده است آمار کامل و دقیقی از معیارها داشته باشیم که البته با مشکلات و کمبودهایی در این راه روبرو بوده‌ایم و دستیابی به جزئیات بیشتر، دقیق‌تر و به‌روزتر از وضعیت معیارهای مختلف با استفاده از تعامل کامل‌تر با مراکز آمار را می‌توان به عنوان یکی دیگر از کارهای آتی در این پژوهش معرفی کرد.

مراجع

[۱] ر. ش. بیگلو، "شناسایی مناطق محروم ایران با استفاده از رتبه‌بندی ترکیبی،" مجله پژوهش و برنامه‌ریزی شهری، سال دوم، شماره هفتم، صفحه ۵۳-۷۰، ۱۳۹۰.

[۲] ا. یوسفان، و ا. یوسفیان، "خوشه‌بندی استان‌های ایران بر پایهٔ معیارهای شکاف دیجیتال به کمک روش K-Means،" نشریه علمی ترویجی محاسبات نرم، سال اول، شماره اول، صفحه ۳۲-۴۵، ۱۳۹۱.

[۳] ح. خورسندی، غ. فقیری، و ع. کلانتر،"راهنمای ارزیابی خسارت سیلاب"، نشریه وزارت نیرو، مدیریت منابع آب ایران، شماره ۲۹۶-الف، ۱۳۸۵.

[۴] س. م. ا. رشتیان، ش. مهره‌کش، ز. عباسی، ح. ر. خدابنده، و ن. یزدانی، "ارزیابی وضعیت توسعه فن‌آوری اطلاعات در استان‌های کشور با استفاده از شاخص‌های بین‌المللی"، دومین کنفرانس بین‌المللی فناوری اطلاعات و دانش، تهران، دانشگاه امیرکبیر، ۱۳۸۴.

[۵] ا. ص. عمران، ا. عالیشوندی، و ب. کاوه‌ئی، "بررسی سیاست پذیرش دانشجوی بومی در دانشگاه‌های ایران"، اولین همایش بین‌المللی مدیریت، آینده‌نگری، کارآفرینی و صنعت در آموزش عالی، ۱۳۹۰.

[۶] نتایج آمارگیری از کاربران اینترنت-۱۳۸۹، نشریه مرکز آمار ایران، ۱۳۹۰.

[۷] سالنامه آماری کشور، نشریه مرکز آمار ایران، ۱۳۹۰.

[۸] فصلنامه آماری، نشریه مرکز آمار و اطلاعات راهبردی، سال چهارم، شماره ۱۰، صفحه ۷۲-۸۱، ۱۳۹۱.

[۹] ا. یعقوبی، م. پورتندرست، و همکاران، آمار تفصیلی صنعت برق ایران ویژه تولید نیروی برق درسال ۱۳۹۰، وزارت نیرو، شرکت مادر تخصصی توانیر، ۱۳۹۱.

[۱۰] ا. یعقوبی، م. پورتندرست، و همکاران، آمار تفصیلی صنعت برق ایران ویژه توزیع نیروی برق درسال ۱۳۹۰، وزارت نیرو، شرکت مادر تخصصی توانیر، ۱۳۹۱.

[۱۱] وزارت نیرو، دفتر بهبود بهره‌وری و اقتصاد برق و انرژی، تعرفه و قوانین و مقررات فروش برق، http://tariff.moe.org.ir.

[۱۲] وزارت راه و شهرسازی، سازمان هواشناسی کشور، خلاصه رخدادهای حدی اقلیمی ایران در سال ۲۰۱۲، http://www.cri.ac.ir.

[۱۳] پایگاه ملی داده‌های علوم زمین کشور، نقشه میانگین بارندگی سالیانه، http://www.ngdir.ir/maps/PAverageAnnualPrecipitationMap.asp.

[۱۴] دویچه وله فارسی، مهمترین مراکز اتمی ایران، http://dw.de/p/16Mya.

[۱۵] پژوهشگاه بین‌المللی زلزله‌شناسی و مهندسی زلزلهٔ ایران، http://www.iiees.ac.ir.

[۱۶] مرکز لرزه‌نگاری کشور، مرکز ژئوفیزیک دانشگاه تهران، http://irsc.ut.ac.ir.

[۱۷] سازمان زمین‌شناسی و اکتشافات معدنی کشور، http://www.gsi.ir.

[۱۸] مرکز ملی اقلیم‌شناسی، پژوهشکده اقلیم‌شناسی، http://www.cri.ac.ir.

[19] ASHRAE Technical Committee 9.9, "*Thermal Guidelines for Data Processing Environments*," ASHRAE-American Society of Heating, Refrigerating and Air-Conditioning Engineers, White Paper, 2012.

[20] S. Strutt, and et. al., "*Data Center Efficiency and IT Equipment Reliability at Wider Operating Temperature and Humidity Ranges,*" The Green Grid, 2011.

[21] H. S. Kinvig, A. Winson, and J. Gottsmann, "Analysis of volcanic threat from Nisyros Island, Greece, with implications for aviation and population exposure," *Natural Hazards and Earth System Science*, vol. 10, no. 6, pp. 1101-1113, 2010.

[22] L. G. Mastin, and et. al., "*Preliminary Spreadsheet of Eruption Source Parameters for Volcanoes of the World,*" U.S. Geological Survey [Online]. Available: http://pubs.usgs.gov/of/2009/1133/.

[23] H. H. Willis, and et. al., "*Terrorism Risk Modeling for Intelligence Analysis and Infrastructure Protection*," RAND Center for Terrorism Risk Management Policy, Department of Homeland Security, 2007.

[24] T. L. Saaty, *The Analytic Hierarchy Process*, McGrawHill, 1980.

[25] J. Rath. *Data Center Site Selection* [Online]. Available: http://rath-family.com/rc/DC_Site_Selection.pdf.

[26] California Data Center Design Group (CDCDG). *Site Selection Criteria for a Critical Data Center* [Online]. Available: http://www.cdcdg.com/index.php/article-4.

[27] International Telecommunication Union, *"Measuring the Information Society,"* ITU, Geneva, Switzerland, 2012.

[28] A. Koch, and J. Wolf, "*Iran's Nuclear Facilities: a Profile,"* Center for Nonproliferation Studies, in Monterey, California, 1998.

[29] L. Evans. (2003). *RAND Tries to Model Risks of Terrorist Attacks*, UCLA [Online]. Available: http://www.international.ucla.edu/article.asp?parentid=3713.

[30] *Volcanoes of Iran (8 volcanoes)* [Online]. Available: http://www.volcanodiscovery.com/iran.html.

مجید حاجی بابا دانشجوی دکتری سازمان پژوهش‌های علمی و صنعتی می‌باشد. او مدرک کارشناسی‌ارشد خود را از دانشگاه علم و صنعت ایران اخذ نموده است. موضوع تز دکتری ایشان درباره مدل‌های پردازشی موازی و توزیع شده برای ترازبندی دنباله‌ها در نسل جدید توالی یابی می‌باشد. از موضوعات مورد علاقه ایشان محاسبات توزیع شده، محاسبات ابری، پردازش جریان‌های داده، داده‌های بزرگ و محاسبات کارا می‌باشد. از دیگر موضوعات مورد علاقه ایشان می‌توان به کامپایلرها و زبان‌های برنامه‌نویسی اشاره کرد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

hajibaba.m@irost.org

سعید گرگین مدرک کارشناسی و کارشناسی‌ارشد خود را در سال‌های ۱۳۸۱ و ۱۳۸۴ از دانشگاه آزاد اسلامی اخذ نمود و در سال ۱۳۸۹ موفق به اخذ مدرک دکتری از دانشگاه شهید بهشتی گردید. در حال حاضر او استادیار پژوهشکده برق و فناوری اطلاعات سازمان پژوهش‌های علمی و صنعتی ایران است. همچنین به عنوان محقق در پژوهشکده کامپیوتر پژوهشگاه دانش‌های بنیادی مشغول به فعالیت می‌باشد. در کنار موضوعات تحقیقات کاربردی در حوزه فناوری اطلاعات، سایر زمینه‌های مورد علاقه او عبارتند از: حساب کامپیوتری، طراحی VLSI، سیستم‌های پردازش سریع و پردازش موازی.

آدرس پست‌الکترونیکی ایشان عبارت است از:

gorgin@irost.org

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۸/۱۰
تاریخ اصلاح: ۱۳۹۴/۱۰/۱۷
تاریخ قبول شدن: ۱۳۹۴/۱۱/۰۱
نویسنده مرتبط: دکتر سعید گرگین، پژوهشکده برق و کامپیوتر، سازمان پژوهش‌های علمی و صنعتی ایران، تهران، ایران.

---

[1] Green Data Center
[2] Economizer
[3] Liquid Cooling
[4] PDSI: Palmer Drought Severity Index
[5] SPI: Standardized Precipitation Index
[6] National Volcano Early Warning System
[7] Conservative Score (CS)
[8] Extreme Score (ES)
[9] Stratovolcanoe
[10] Maars
[11] Indicator
[12] Index
[13] American Society for Public Administration
[14] Economic Intelligence Unit
[15] United Nations Conference on Trade and Development
[16] TOPSIS
[17] Major Voting

علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۱۴-۲۲
مقاله عادی

ارائه روش بهبود یافته زمان‌بندی کارها در محیط ابر با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات

فاطمه عبادی‌فرد احمد اکبری

دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران

چکیده

محاسبات ابری یکی از زمینه‌هایی است، که در سال‌های اخیر توجه کاربران زیادی را به خود جلب کرده است، این امر به دلیل مزایای قابل توجهی است، که سرویس‌های ابر از لحاظ هزینه و کارایی در اختیار مصرف‌کنندگان قرار می‌دهند. مسئله ی زمان بندی کارها یکی از مهمترین مسائل جهت استفاده مناسب از قابلیت های محیط ابر می باشد. به طور کلی مسئله زمان‌بندی کارها در محیط ابر یک مسئله ان‌پی-سخت می‌باشد [۱۶]. در مسئله‌ی زمان‌بندی کارها از یک طرف باید نیازمندی‌های کاربران در نظر گرفته شود و از طرف دیگر باید از منابع موجود حداکثر استفاده شود. در این مقاله روشی را برای زمان‌بندی کارها با استفاده از رویکرد بهبود یافته‌ی الگوریتم بهینه‌سازی ازدحام ذرات[۱] به کمک استفاده از الگوریتم تعادل بار برای تولید نسل اولیه مناسب‌تر، ارائه کرده‌ایم که روش پیشنهادی علاوه بر آنکه بار کاری را بین ماشین‌های مجازی موجود متعادل کرده، با انتخاب تابع برازندگی مناسب، سبب کاهش زمان تکمیل تمامی کارها[۲]، همچنین استفاده‌ی حداکثر از منابع، می‌شود. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با روش بهینه‌سازی ازدحام ذرات بدون استفاده از تعادل بار، به‌میزان ۳۳ درصد بهبود در زمان تکمیل تمامی کارها و به میزان ۲۲ درصد بهبود در میزان کارایی منابع دارد.

کلمات کلیدی: محاسبات ابری، زمان‌بندی، بهینه‌سازی ازدحام ذرات، تعادل بار، بهره‌وری منابع.

۱- مقدمه

امروزه محاسبات ابری یکی از تکنولوژی‌های جدیدی است که به طور کامل بر پایه اینترنت می‌باشد و در آن همه‌ی برنامه‌ها و فایل‌ها بر روی ابری قرار می‌گیرد، که شامل هزاران کامپیوتر به هم متصل می‌باشد. محاسبات ابری ترکیبی از محاسبات توزیع شده و موازی به‌منظور فراهم کردن منابع اشتراکی از قبیل سخت افزار، نرم‌افزار و اطلاعات از دیگر دستگاه‌ها می‌باشد [۱]. از دید توسعه‌دهندگان نرم‌افزار، خدماتی که از طریق ابر ارائه می‌شود؛ می‌توان به سه گروه زیرساخت به عنوان خدمات[۳]، سکو به عنوان خدمات[۴] و نرم‌افزار به عنوان خدمات[۵] طبقه‌بندی کرد.

در سطح زیرساخت به عنوان خدمات، فراهم‌کننده‌ی ابر منابع سخت‌افزاری را برای اجرای خدمات با استفاده از فناوری مجازی‌سازی در اختیار کاربران قرار می‌دهد. در سطح سکو به عنوان خدمات، لایه‌ای از نرم‌افزار را برای ایجاد سرویس‌های سطح بالاتر فراهم می‌کند. این بستر توسعه‌ای، به کاربران برای طراحی، توسعه، ارزیابی و میزبانی کردن برنامه‌های کاربردی در سطح ابر کمک می‌کند. در سطح نرم‌افزار به عنوان سرویس کاربران نرم‌افزار مورد نیاز خود را از محیط ابر درخواست می‌کنند.

در میان همه‌ی سطوح سرویس‌دهی محیط‌های ابر، سطح نرم‌افزار به عنوان سرویس در محیط ابر، توجه‌ی کاربران زیادی را به خود جلب کرده است. این سطح روش‌های سنتی استفاده از نرم‌افزار را که شامل نصب نرم‌افزار بر روی ماشینی که قصد استفاده از نرم‌افزار را دارد، با میزبانی نرم‌افزار روی سرورهای راه دور تغییر داده است و سبب شده تا محدودیت‌هایی را که مانع نصب برخی از نرم‌افزارهای حجیم و یا مانع تهیه‌ی نرم‌افزار برای کاربران بوده است را از بین ببرد.

محیط ابر بستری از سرورها را در مرکز داده فراهم می‌کند تا هنگام درخواست منابع توسط کاربران آنها را به صورت اشتراکی در اختیارشان قرار دهد. فراهم‌کنندگان سرویس‌های مختلف[۶] می‌توانند با اجاره ماشین‌های مجازی از

فراهم‌کنندگان ابر، سرویس‌های مختلف خود را به کاربران خود ارائه دهند. کاربران برنامه مورد علاقه خود را از فراهم‌کننده سرویس درخواست و آن را دریافت می‌کنند.

از آنجایی که فراهم‌کنندگان سرویس منابع خود را از فراهم‌کنندگان ابر اجاره می‌کنند، با توجه به پراکندگی جغرافیایی مختلف کاربران و ویژگی‌های خاص برنامه‌های مختلف که توسط کاربران درخواست داده می‌شود، چالش اساسی در این زمینه ارائه روشی کارا برای تخصیص منابع مورد نیاز به درخواست‌های کاربران به صورتی می‌باشد، که بتواند پارامترهای کیفیت سرویس مورد نیاز متناسب با نیازمندی‌های آن برنامه را برای کاربران به همراه داشته باشد [۲].

زمان‌بندی کارها تکنیکی است که وظایف کاربران به ماشین‌های مجازی برای اجرا اختصاص می‌دهد. از دید مشتری الگوریتم زمان‌بندی مناسب باید بتواند وظایف مورد تقاضای او را با کمترین زمان بر روی ماشین مجازی اجرا کند، از طرفی فراهم‌کننده ی سرویس نیازمند نوعی زمان‌بندی است که بتواند در عین رضایت‌مندی مشتری از منابع حداکثر استفاده را بکند. این مساله نیازمندی فراهم کننده‌ی سرویس را به انتخاب روش مناسب برای زمان‌بندی درخواست‌ها بیشتر می‌کند.

در این مقاله با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات [۱۲] به همراه الگوریتم تعادل بار، یک روش برای زمان‌بندی درخواست‌ها بر روی ماشین‌های مجازی ارائه کرده‌ایم که علاوه بر کاهش زمان تکمیل تمامی کارها، سبب افزایش بهره‌وری از منابع[۷] می‌شود.

مزیت الگوریتم ارائه شده در روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات در این است که در روش پایه‌ی بهینه‌سازی ازحام ذرات موقعیت اولیه هر ذره به صورت تصادفی انتخاب می‌شود در صورتی که در روش ارائه شده موقعیت هر ذره در ابتدا به صورت تصادفی تعیین شده و سپس با استفاده از روش تعادل بار[۸] برای توزیع عادلانه‌ی درخواست‌ها بین ماشین‌های مجازی مختلف، جا به جا می‌شود. با استفاده از این روش هر ذره‌ی اولیه خود یک ذره‌ی مناسب می‌باشد که با جابه‌جایی آن به جواب مناسب‌تر می‌رسیم. نتایج شبیه‌سازی نشان می‌دهد که روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات، بهبود بهتری را در زمان تکمیل تمامی کارها و میزان بهره‌وری منابع دارد. به طور خلاصه می‌توان گفت تمرکز اصلی ما در این مقاله شامل موارد زیر می‌باشد.

۱- ارائه‌ی روشی مناسب برای زمان‌بندی درخواست‌ها با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات، با هدف کاهش زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع

۲- استفاده از الگوریتم تعادل بار به همراه الگوریتم بهینه‌سازی ازدحام ذرات برای توزیع عادلانه‌ی درخواست‌ها

۳- تحلیل و ارزیابی هدفمند برای نشان دادن ترکیب موثر الگوریتم بهینه‌سازی ازدحام ذرات با الگوریتم تعادل بار برای بهبود زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع

باقی مقاله از بخش‌های زیر تشکیل شده است: در بخش ۲ مروری بر کارهای مرتبط داریم در بخش ۳ پس از معرفی الگوریتم بهینه‌سازی ازدحام ذرات به بیان مدل ریاضی الگوریتم پرداخته سپس جزئیات روش پیشنهادی به تفصیل بیان می‌شود و در بخش چهارم به شبیه‌سازی و ارزیابی روش پیشنهادی می‌پردازیم و در نهایت در بخش پنجم نتیجه‌گیری و کارهای آینده بیان می‌شود.

۲- کارهای مرتبط

زمان‌بندی در پردازش ابری به معنی تخصیص بهینه‌ای از درخواست‌ها به منابع محاسباتی موجود در مراکز داده می‌باشد. در زمان‌بندی، کارها را به انواع متفاوتی از منابع با توجه به محدودیت‌های مشخص شده، نسبت می‌دهیم. محدودیت‌ها به دو دسته تقسیم‌بندی می‌شود: محدودیت‌هایی که می‌تواند از طرف کاربر باشد و شامل محدودیت در اتمام کار در زمان مشخص، تعیین سررسید (مثلاً مهلت زمانی) خاص و اختصاص هزینه محدود به کار است. دسته دوم محدودیت‌ها مربوط به فراهم‌کنندگان سرویس‌های ابری می‌باشد که شامل پاسخگویی به کار کاربران در زمان اندک و بیشینه شدن سود حاصل از انجام کار و یا افزایش بهره وری منابع می‌باشد. توجه به این محدودیت‌ها بسیار لازم و ضروری است.

در رابطه با زمان‌بندی کارهای متفاوتی انجام شده است که ما آنها را در دسته‌بندی زیر ارائه می‌کنیم: الف: روش‌هایی که برای زمان‌بندی کارها از الگوریتم‌های اکتشافی استفاده کرده است. در این روش‌ها از الگوریتم‌های اکتشافی مختلف مانند ژنتیک، بهینه‌سازی ازدحام ذرات، کلونی مورچگان و سایر الگوریتم‌های اکتشافی [۴-۶] برای حل مسئله زمان‌بندی استفاده شده است. از آنجایی که پایه کار ما بر روی الگوریتم بهینه‌سازی ازدحام ذرات می‌باشد، در ادامه برخی از مهمترین روش‌هایی که از این الگوریتم استفاده شده است را بیان می‌کنیم:

پاندی و دیگر همکاران [۹] یک روش زمان‌بندی برای جریان گردش کار بر پایه الگوریتم بهینه‌سازی ازدحام ذرات ارائه کرده است. در روش پیشنهادی هدف کاهش هزینه‌ی منابع و هزینه‌ی انتقال داده می‌باشد و برای رسیدن به این هدف در ابتدا براساس متوسط هزینه محاسباتی تمامی کارها بر روی تمامی منابع موجود و متوسط هزینه انتقال داده برای هر منبع و هر کار یک وزن اختصاص می‌دهد و سپس براساس وزن اختصاص یافته، تابع هدف را برای الگوریتم بهینه‌سازی ازدحام ذرات تعیین می‌کند و الگوریتم را اجرا می‌کند. در ادامه روش ارائه شده را با روش انتخاب بهترین منبع[۹] مقایسه کرده و نتایج شبیه‌سازی نشان می‌دهد که الگوریتم ارائه شده نسبت به روش انتخاب بهترین منبع کاهش هزینه بهتری را در بر دارد. الگوریتم پیشنهادی اگرچه سبب کاهش هزینه شده است، اما برخی از معیارهای کیفیت سرویس مانند زمان انجام کار که سبب رضایت‌مندی مشتری می‌شود را در نظر نگرفته است. در سال ۲۰۰۸ لی زنگ و دیگر همکاران [۸] با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات الگوریتمی را برای زمان‌بندی بهینه‌ی کارها در محیط گرید ارائه دادند که هدف در آن کاهش زمان اتمام کار بود. تابع هدف در الگوریتم پیشنهادی کمترین زمان اجرای کار بود و در نهایت روش پیشنهادی را در شرایط مشابه با الگوریتم ژنتیک مقایسه کرده است .سپس براساس مقایسات نشان داده است که کارایی الگوریتم بهینه‌سازی ازدحام ذرات از ژنتیک بیشتر است. این مقاله اگرچه معیارهای کیفیت سرویس کمی مورد توجه قرار داده است و تابع هدف مناسبی را انتخاب نکرده است ولی با ایجاد شرایط مشابه کارایی دو الگوریتم اکتشافی بهینه‌سازی ازدحام ذرات و ژنتیک را مقایسه کرده و برتری الگوریتم بهینه‌سازی ازدحام ذرات را نشان داده است.

گیو و دیگر همکاران [۷] در سال ۲۰۱۲ روشی را برای زمان‌بندی کارها در محیط ابر بر مبنای الگوریتم بهینه‌سازی ازدحام ذرات به‌منظور کاهش زمان تکمیل کار ارائه کردند، تابع هدف در روش پیشنهادی زمان پردازش به همراه زمان انتقال داده می‌باشد. در نتیجه در روش ارائه شده هر دو مقدار زمان پردازش کار و زمان انتقال داده با هم در نظر گرفته می‌شد. روش ارائه شده باعث کاهش زمان تکمیل کار می‌شود ولی تابع هدف آن مناسب نبوده و برخی از معیارهای کیفیت سرویس متناسب را در نظر نمی‌گیرد.

همانطور که بیان شد الگوریتم بهینه‌سازی ازدحام ذرات الگوریتم مناسبی برای زمانبندی درخواست‌ها می‌باشد که می‌تواند با استفاده از تابع هدف مناسب نیازمندی‌های کاربران و فراهم‌کنندگان سرویس را برآورده کند. برخی از روش‌های ارائه شده سعی در بهبود الگوریتم بهینه‌سازی ازدحام ذرات گرفته و برای این کار این الگوریتم را با یک الگوریتم دیگر ترکیب کرده‌اند. در ادامه در دسته دوم

روش‌هایی که از ترکیب الگوریتم بهینه‌سازی ازدحام ذرات با سایر الگوریتم‌ها برای حل مسئله زمان‌بندی استفاده کرده‌اند، را بیان می‌کنیم.

در سال ۲۰۱۲، زن [۱۱] یک روش بهبود یافته‌ی بهینه‌سازی ازدحام ذرات را با هدف کاهش متوسط زمان اجرا و افزایش سرعت همگرایی ارائه داد. ایده‌ی پیشنهادی این الگوریتم ترکیب الگوریتم سرد و گرم فلزات (SA) به همراه الگوریتم بهینه‌سازی ازدحام ذرات، می‌باشد. الگوریتم سرد و گرم فلزات به صورت محلی جستجو را انجام می‌دهد، ترکیب یک الگوریتم اکتشافی مبتنی بر جمعیت مثل بهینه‌سازی ازدحام ذرات، با الگوریتم جستجوی محلی همانند الگوریتم سرد و گرم فلزات سبب همگرایی سریع‌تر برای رسیدن به جواب بهینه می‌شود. روش پیشنهادی با سایر الگوریتم‌های ژنتیک، سرد و گرم فلزات و کلونی مورچگان مقایسه شده و کارایی آن از بقیه بیشتر است. در روش پیشنهادی فقط پارامتر زمان اجرا بررسی شده و دیگر معیارهای کیفیت سرویس در نظر گرفته نشده است علاوه بر این استفاده از دو الگوریتم اکتشافی سبب افزایش پیچیدگی در الگوریتم ارائه شده می‌شود.

در سال ۲۰۱۴، عبدی [۱۴] و دیگر همکارانش یک روش زمان‌بندی با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات با هدف کاهش زمان تکمیل تمامی کارها ارائه کردند که در روش پیشنهادی از الگوریتم اختصاص کوتاه‌ترین کار به سریع‌ترین پردازنده، برای بهبود الگوریتم بهینه‌سازی ازدحام ذرات استفاده شده است. در این روش به جای تولید تصادفی نسل اولیه در الگوریتم بهینه‌سازی ازدحام ذرات، با قرار دادن وظایف کوتاه‌تر بر روی پردازنده سریعتر، نسل اولیه را مناسب‌تر انتخاب کردند و سپس با استفاده از جابه‌جایی این ذرات جمعیت و ارزیابی آن با تابع هدف جمعیت جدید را تولید می‌کند. در نهایت روش ارائه شده را با الگوریتم ژنتیک و بهینه‌سازی ازدحام ذرات، مقایسه کرده‌اند. همچنین در سال ۲۰۱۴، کوار [۱۳] و دیگر همکاران روشی را با هدف افزایش سود حاصل از منابع و بهره‌وری از منابع به کمک ترکیب الگوریتم سرد و گرم فلزات و بهینه‌سازی ازدحام ذرات، ارائه کردند.

از آنجایی که در الگوریتم پیشنهادی از ترکیب الگوریتم بهینه‌سازی ازدحام ذرات و تعادل بار استفاده شده است، مروری بر روش‌های تعادل بار انجام داده‌ایم: در رابطه با تعادل بار کارهای متفاوتی انجام شده است، به طور کلی الگوریتم‌های تعادل بار به دو دسته ایستا[۱۰] و پویا[۱۱] تقسیم‌بندی می‌شوند. در روش ایستا تخصیص وظایف به ماشین‌های مجازی، براساس قابلیت‌های ماشین مجازی و وضعیت اولیه هر ماشین، می‌باشد، برخلاف الگوریتم‌های ایستا، در روش‌های پویا توزیع‌کننده علاوه بر قابلیت‌های اولیه هر ماشین مجازی، براساس وضعیت حال حاضر آن ماشین و بارکاری موجود بر آن، وظایف را به ماشین‌های مجازی اختصاص می‌دهد. روشی را که ما برای تعادل بار استفاده کرده‌ایم الهام گرفته از مقاله‌ای است که در سال ۲۰۱۳ در [۱۶] بر پایه رفتار زنبور عسل ارائه شده است، که هدف در آن کاهش زمان تکمیل تمامی کارها بود و در آن ابتدا درخواست‌ها به صورت نوبت گردشی به ماشین‌های مجازی اختصاص داده می‌شد، سپس درخواست‌ها از ماشین دارای اضافه بار به ماشین دارای بار کم انتقال داده می‌شد.

همانطور که گفتیم در رابطه با الگوریتم بهینه‌سازی ازدحام ذرات کارهای قبلی زیادی انجام شده است اما تفاوتی که در کار ما با سایر کارهای انجام شده وجود دارد این است که در روش پیشنهادی ما با ترکیب الگوریتم بهینه‌سازی ازدحام ذرات و الگوریتم تعادل بار، به منظور تولید نسل اولیه مناسب‌تر و انتخاب تابع هدف مناسب، از طرفی توانسته‌ایم با کاهش زمان تکمیل تمامی کارها و پاسخگویی سریع به درخواست‌های مشتری رضایت‌مندی مشتری را به همراه داشته باشیم و از طرف دیگر با افزایش بهره‌وری منابع،سبب افزایش رضایت‌مندی فراهم‌کننده‌ی سرویس شویم. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی در مقایسه با روش بهینه‌سازی ازدحام ذرات بدون استفاده از تعادل بار و روش بهینه‌سازی ازدحام ذراتی که نسل اولیه آن به کمک الگوریتم SJFP[۱۲] بهینه شده است، به صورت کاراتر عمل می‌کند.

۳- روش پیشنهادی

۳-۱- بهینه‌سازی ازدحام ذرات کلاسیک

الگوریتم بهینه‌سازی ازدحام ذرات یک الگوریتم بهینه‌سازی فرا اکتشافی است که از حرکت گروهی پرندگان (و دیگر حیواناتی که به شکل گروهی زندگی می‌کنند) الگو گرفته است. در این الگوریتم هر پاسخ مساله به صورت یک ذره که دارای یک مقدار و همچنین میزان تناسب است، مدل می‌شود. الگوریتم بهینه‌سازی ازدحام ذرات اولین بار توسط راسل و کندی در سال ۱۹۹۵ ارائه شد [۱۲]. الگوریتم بهینه‌سازی ازدحام ذرات بر مبنای حرکت و هوش ذرات کار می‌کند. در این الگوریتم هر ذره در حال جستجو برای نقطهٔ بهینه می‌باشد در نتیجه نیازمند آن است که در حال جابه‌جایی باشد، زیرا در غیر این‌صورت نمی‌تواند به فرایند جستجو ادامه دهد.

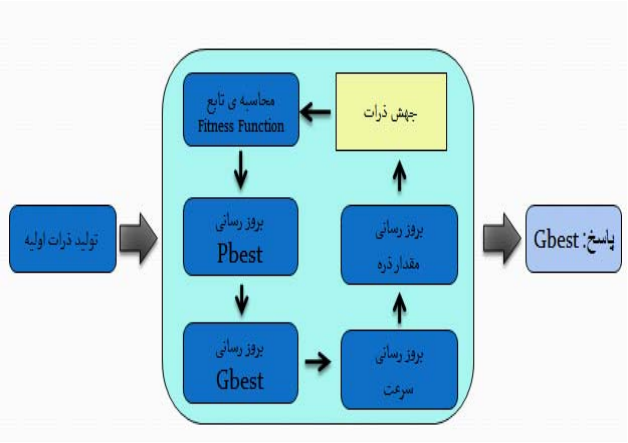
حرکت هر ذره به سه عامل بستگی دارد:

۱- موقعیت فعلی ذره.

۲- بهترین موقعیتی که تا کنون ذره داشته است. (بهترین موقعیت محلی[۱۳])

۳- بهترین موقعیتی که کل مجموعه‌ی ذرات تاکنون داشته‌اند. (بهترین موقعیت سراسری[۱۴])

در الگوریتم بهینه‌سازی ازدحام ذرات کارایی هر ذره براساس تابع برازندگی آن می‌باشد. هر ذره در هر مرحله، موقعیتی را که بهترین نتیجه را در آن داشته به خاطر می‌سپارد. (بهترین موقعیت فردی هر ذره) همچنین ذرات در گروه ذرات با هم همیاری می‌کنند. ذرات اطلاعاتی که دربارهٔ موقعیتی که در آن هستند را با هم تبادل می‌کنند. در نتیجه‌ی این تبادل اطلاعات بهترین موقعیت کل مجموع ذرات را می‌توان نگه‌داری کرد. در هر مرحله بر اساس اطلاعاتی که از بهترین موقعیتی که تاکنون ذره داشته است و بهترین موقعیتی که کل مجموعه‌ی ذرات تاکنون داشته‌اند می‌توانیم مقدار سرعت را به‌روزرسانی کنیم و سپس براساس سرعت به روز شده مقدار ذره را به روزرسانی کرده و سرانجام مقدار جهش ذره را تعیین کنیم. شکل ۱ مراحل الگوریتم بهینه‌سازی ازدحام ذرات را نشان می‌دهد.



شکل ۱- مراحل الگوریتم بهینه‌سازی ازدحام ذرات

در هر نسل سرعت و موقعیت ذره با استفاده از روابط (۱) و (۲) به روز می‌شود:

$$V_i^{k+1} = V_i^k + c_1.\mathrm{rand}_1 \times \left(\mathrm{pbest}_i - X_i^k\right) + c_2.\mathrm{rand}_2 \times \left(\mathrm{gbest}_i - X_i^k\right) \quad (۱)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (۲)$$

هر یک از متغیرهای روابط (۱) و (۲) مطابق جدول ۱ تعریف می‌گردد.

جدول ۱- تعریف متغیرهای الگوریتم PSO

v_i^k	سرعت ذره i در مرحله k
v_i^{k+1}	سرعت ذره i در مرحله k+1
w	وزن داخلی
c_j	ضریب شتاب j=1,2
$rand_i$	عدد تصادفی بین ۰و۱ i=1,2
x_i^k	موقعیت فعلی ذره i در مرحله k
$pbest_i$	بهترین موقعیت ذره i
$gbest_i$	بهترین موقعیت در بین تمامی ذرات
x_i^{k+1}	موقعیت فعلی ذره i در مرحله k+1

در ادامه روش پیشنهادی برای استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به همراه الگوریتم تعادل بار، برای حل مسئله‌ی زمان‌بندی، بیان می‌شود.

۳-۲- استفاده از الگوریتم بهینه‌سازی ازدحام ذرات به‌همراه الگوریتم تعادل بار در مسئله‌ی زمان‌بندی

محاسبات ابری با تخصیص مجموعه‌ای از وظایف محاسباتی بر روی مجموعه‌ای از ماشین‌های مجازی بر طبق نیازمندی کاربران و ویژگی‌های هر یک از ماشین‌های مجازی سروکار دارد، درخواست‌های کاربران برای سرویس‌های مختلف بر طبق سیاست‌های زمان‌بندی مختلف به هر یک از ماشین‌های مجازی هدایت می‌شود. امروزه الگوریتم‌های مختلفی برای حل مسئله‌ی زمان بندی با اهداف مختلف ارائه شده است. هدف ما در این مقاله کاهش زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع می‌باشد.

۳-۲-۱- مدل ریاضی الگوریتم

فرض کنید $VM = \{VM_1, VM_2, \dots, VM_m\}$ مجموعه‌ی ماشین‌های مجازی باشند که برای میزبانی درخواست‌های کاربران استفاده می‌شوند. همچنین $Task = \{T_1, T_2, \dots, T_n\}$ مجموعه‌ای از وظایفی می‌باشد که قصد اجرا بر روی ماشین‌های مجازی را دارد. هدف از الگوریتم کاهش در میزان زمان تکمیل تمامی کارها و افزایش متوسط بهره‌وری منابع می‌باشد که زمان تکمیل تمامی کارها را، طبق رابطه‌ی (۳) [۱۹] تعریف می‌کنیم:

$$\text{Makespan} = \max_{1 \le i \le m} \sum_{j=1}^{n} PT_{ij} * x_{ij} \qquad (۳)$$

به‌طوریکه PT_{ij} در رابطه (۳) زمان تکمیل وظیفه j بر روی ماشین مجازی i می‌باشد. همچنین در صورتی‌که درخواست j بر روی ماشین مجازی i اجرا شده باشد مقدار xij برابر ۱ و در غیر این صورت برابر ۰ می‌باشد.

طبق رابطه (۳) از بین ماشین‌های مجازی ۱ تا m، ماشینی که مجموع زمان تکمیل تمامی وظایف محوله به آن، از همه بیشتر است، عامل تعیین makespn بوده و زمان تکمیل تمامی وظایف آن ماشین به عنوان makespan در نظر گرفته می‌شود.

درصورتی که مقدار بهره‌وری از هر منبع طبق رابطه‌ی (۴) [۱] محاسبه شود، متوسط بهره‌وری طبق رابطه‌ی (۵) محاسبه می‌شود.

$$\text{Utilization}_i = \frac{\sum_{j=1}^{n} PT_{ij} * xij}{\text{Makespan}} \qquad (۴)$$

به‌طوریکه PT_{ij} در رابطه (۴) زمان پردازش وظیفه j بر روی ماشین i می‌باشد.

در رابطه (۴) میزان بهره‌وری از منبع i برابر مجموع زمان پردازش تمام وظایف محول شده به ماشین مجازی i، به نسبت زمان پردازش ماشینی است که مجموع زمان پردازش تمامی وظایف آن از باقی ماشین‌ها بیشتر می‌باشد. بدیهی است که مقدار بهره‌وری از هر ماشین مجازی کمتر و یا مساوی عدد یک می‌باشد.

$$Average\ Utilization = \frac{\sum_{i=1}^{m} Utilization_i}{m} \qquad (۵)$$

در نتیجه تابع برازندگی با هدف کاهش زمان تکمیل تمامی کارها و افزایش بهره‌وری از منابع به صورت رابطه‌ی (۶) تعریف می‌شود:

$$Fitness\ Function = \frac{Makespan}{average\ Utilization} \qquad (۶)$$

با توجه به رابطه‌ی (۶) می‌توان نتیجه گرفت که هر چه قدر مقدار تابع برازندگی کمتر باشد میزان مطلوبیت ذره از جهت کاهش زمان تکمیل تمامی کارهاو افزایش بهره‌وری از منابع بیشتر است. در ادامه جزئیات الگوریتم پیشنهادی را به تفصیل بیان کرده و در انتها شبه کد آن را قرار می‌دهیم.

۳-۲-۲- جزئیات الگوریتم پیشنهادی تعادل بار با استفاده از الگوریتم ازدحام ذرات

- انتخاب ذرات اولیه و سرعت اولیه

ابتدا اندازه‌ی هر ذره را به تعداد وظایف قرار داده و برای هر ذره یک مکان تصادفی و یک سرعت اولیه تصادفی ایجاد می‌کنیم. برای مثال در صورتی که تعداد شش وظیفه و سه ماشین مجازی داشته باشیم ممکن است به صورت تصادفی وظایف طبق جدول ۲ بر روی ماشین‌های مجازی قرار بگیرند.

جدول ۲- نحوه قرارگیری مکان اولیه ذرات در الگوریتم PSO

وظیفه ۱	وظیفه ۲	وظیفه ۳	وظیفه ۴	وظیفه ۵	وظیفه ۶
ماشین مجازی ۲	ماشین مجازی ۱	ماشین مجازی ۱	ماشین مجازی ۳	ماشین مجازی ۲	ماشین مجازی ۲

- استفاده از روش تعادل بار برای متعادل کردن هر ماشین مجازی و بهبود مکان ذرات

در این مرحله میزان بار هر ماشین مجازی را طبق رابطه‌ی (۷) محاسبه کرده و ماشین‌های مجازی را به سه دسته‌ی ماشین های مجازی دارای افزونگی بار، ماشین‌های مجازی دارای کم‌باری و ماشین‌های مجازی دارای بارکاری متعادل تقسیم‌بندی می‌کنیم.

$$L_{V,M_i,t} = \frac{N(T,t)}{S(VM_i,t)} \qquad (۷)$$

بار هر ماشین مجازی براساس نسبت تعداد درخواست‌های موجود در صف سرویس‌دهی ماشین مجازی i در زمان t، برروی نرخ سرویس‌دهی ماشین i در زمان t محاسبه می‌شود.

برای این کار میزان بارکاری ماشین مجازی را در مقایسه با کل بار کاری موجود در تمامی ماشین‌های مجازی بررسی می‌کنیم، سپس براساس نتایج حاصل از مقایسه ماشینی که بارکاری آن از متوسط بارکاری سیستم بالاتر بود دارای افزونگی بار، در صورتی که در محدوده‌ی متوسط بارکاری سیستم بود دارای تعادل بار و در غیراین صورت دارای کم باری می‌باشد. سپس درخواست‌ها را از ماشین‌هایی که دارای افزونگی بار هستند یکی یکی برداشته و بر روی ماشین‌های کم بار قرار می‌دهیم. این کار را تا جایی ادامه می‌دهیم که یکی از مجموعه‌های ماشین‌های دارای افزونگی بار و یا ماشین‌های دارای کم باری خالی شود.

- محاسبه‌ی تابع برازندگی و انتخاب مقادیر Pbest, Gbest

در این مرحله مقدار تابع برازندگی را برای هر یک از ذرات طبق رابطه‌ی (۶) محاسبه می‌کنیم. مقدار بهترین موقعیت محلی را مقدار اولیه هر ذره قرار می‌دهیم و از بین ذرات، ذره‌ای که دارای بهترین مقدار تابع برازندگی می‌باشد را به عنوان بهترین موقعیت سراسری انتخاب می‌کنیم.

- به‌روزرسانی مکان ذرات و سرعت آنها

در این مرحله طبق رابطه‌ی (۱) ابتدا میزان سرعت اولیه براساس اطلاعاتی که از سرعت قبلی، بهترین مکان فعلی ذره و بهترین مکان در کل ذرات داریم، به‌روز می‌شود، سپس مکان ذره براساس سرعت به‌روز شده طبق رابطه‌ی (۲) تعیین می‌شود . سپس تابع برازندگی برای موقعیت جدید ذره محاسبه شده و در صورتی که مقدار آن از موقعیت قبلی ذره کمتر بود بهترین موقعیت محلی به ذره‌ی جدید تغییر می‌یابد. در بین گروه جدید مکان ذرات در صورتی کمترین مقدار تابع برازندگی از مقدار بهترین موقعیت سراسری قبلی کمتر بود، مقدار بهترین موقعیت سراسری به‌روزرسانی می‌شود.

- تکرار مراحل فوق به اندازه‌ی تعداد مراحل[۱۵] الگوریتم

به تعداد مراحل الگوریتم، موقعیت هر ذره به روزرسانی شده و همچنین براساس موقعیت هر ذره و گروه ذرات مقادیر بهترین موقعیت محلی و بهترین موقعیت سراسری محاسبه می‌شود. در نهایت ذره‌ای که کمترین مقدار بهترین موقعیت سراسری را دارد به عنوان پاسخ مساله در نظر گرفته می‌شود. در ادامه شبه کد الگوریتم پیشنهادی در الگوریتم ۱ آورده شده است.

الگوریتم ۱- شبه‌کد الگوریتم پیشنهادی

```
Input:VM = {VM_1, VM_2, ...., VM_m},
Task = {T_1, T_2, ...., T_n}
Output:  best position of Tasks on the VM(Gbest)
Start:
1: Set particle dimensionas equal to the size of ready tasks
2: Initialize particles position randomly and velocity vi
randomly.
3: for  each particle run load balancing algorithm for
balance particles position
4: For each particle, calculate its fitness value as in
Equation(3-6)
     If (fitness value <  previousbestpbest)
        set the current fitness value as the newpbest
5: After Steps 4 for all particles, select the best particle as
gbest
6: For all particles, calculate velocity using Equation (3-1)
           and update their positions using Equation (3-2)
7: If (stopping criteria or maximum iteration is not
satisfied)
       repeat from Step 4.
    else
        Return gbest
Stop.
```

با استفاده از الگوریتم ارائه شده می‌توان درخواست‌ها را به بهترین صورت ممکن به ماشین‌های مجازی اختصاص دهیم به‌طوری که میزان زمان تکمیل تمامی کارها، را کاهش داده و حداکثر استفاده از منابع را داشته باشیم. مزیت الگوریتم ارائه شده در روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات در این است، که در روش پایه‌ی بهینه‌سازی ازدحام ذرات موقعیت اولیه هر ذره به صورت تصادفی انتخاب می‌شود در صورتی که در روش ارائه شده موقعیت هر ذره در ابتدا به صورت تصادفی تعیین شده و سپس با استفاده از روش تعادل‌بار، برای توزیع عادلانه‌ی درخواست‌ها بین ماشین‌های مجازی مختلف، جابه‌جا می‌شود. با استفاده از این روش هر ذره‌ی اولیه خود یک ذره‌ی مناسب می‌باشد که با جابه‌جایی آن به جواب مناسب‌تر می‌رسیم. نتایج شبیه‌سازی نشان می‌دهد که روش فوق نسبت به روش پایه‌ی بهینه‌سازی ازدحام ذرات، بهبود قابل توجه‌ای را در زمان پاسخ و میزان بهره‌وری منابع دارد.

۴- شبیه‌سازی و ارزیابی روش پیشنهادی

در این بخش به جزئیات شبیه‌سازی الگوریتم معرفی شده در بخش قبل خواهیم پرداخت. سپس از طریق رسم نمودار به ارزیابی روش‌های مذکور می‌پردازیم. ذکر این نکته که محیط مورد بررسی ما یک محیط نرم‌افزار به‌عنوان سرویس است و همچنین ابزار ما برای شبیه‌سازی نرم‌افزار کلودسیم [۱۷] است، مفید خواهد بود. این شبیه‌ساز، به ما اجازه‌ی ایجاد یک محیط مجازی‌سازی شده را می‌دهد و از تخصیص منابع براساس تقاضا پشتیبانی می‌کند. در واقع ما هسته این شبیه‌ساز را برای مدل کردن الگوریتم ارائه شده گسترش داده‌ایم.

برای ارزیابی این بخش یک مرکز داده رایانش ابری را شبیه‌سازی کرده‌ایم که از سه میزبان تشکیل شده که قابلیت مجازی‌سازی[۱۶] دارند. در واقع فرض شده است که روی آن‌ها مجازی سازهایی مثل Xen نصب شده است،که می‌تواند منابع را به اشتراک بگذارد. مشخصات هر یک از میزبان‌ها مطابق با جدول ۳ می‌باشد.

جدول ۳- مشخصات میزبان‌ها

شماره میزبان	تعداد هسته‌های پردازنده	سرعت پردازنده (تعداد دستورالعمل در ثانیه)	حافظه داخلی (مگا بیت)	حافظه جانبی (مگا بیت)	پهنای باند (مگا بیت بر ثانیه)
۱	۴	۵۰۰۰۰	۲۰۴۸۰۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
۲	۲	۲۵۰۰۰	۱۰۲۴۰۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
۳	۱	۱۰۰۰۰	۵۱۲۰۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰

روی این مرکز داده ۱۶ ماشین مجازی با مشخصات متفاوت قرار داده ایم. هرکدام از ماشین‌های مجازی چند برنامه کاربردی با تعداد دستورالعمل‌های متغیر بین ۵۰۰ تا ۴۵۰۰ دستورالعمل را اجرا می‌کنند. همانطورکه قبلا هم اشاره کردیم هدف ارائه‌ی الگوریتم جامع و مناسب برای زمان‌بندی درخواست‌ها در بستر ابر می‌باشد به طوری‌که زمان تکمیل تمامی کارها[۱۷] در آن کمترین بوده و حداکثر بهره‌وری از منابع را در بر داشته باشد. برای شبیه‌سازی الگوریتم بهینه‌سازی ذرات ده ذره را در نظر گرفته‌ایم که طی ۱۰۰ مرحله تکرار[۱۸] به جواب مناسب می‌رسیم. پارامترهای لازم برای شبیه‌سازی الگوریتم ارائه شده مطابق با جدول ۴ می‌باشد.

در بخش اول مقایسه‌ای جهت بررسی زمان تکمیل تمامی کارها، بین روش‌های نوبت گردشی که بدون استفاده از روش‌های تعادل بار به توزیع درخواست‌ها می‌پردازد و سپس روش استفاده از الگوریتم بهینه‌سازی ازدحام ذرات پایه و روش تعادل بار بدون استفاده از بهینه‌سازی ذرات و الگوریتم بهینه‌سازی ازدحام ازدحام ذرات بهبود یافته انجام می‌شود. نمودار شکل ۲ این مقایسه را

انجام می‌دهد. در این نمودار محور افقی تعداد وظایف و محور عمودی زمان تکمیل تمامی کارها، را نشان می‌دهد.

جدول ۴- پارامترهای لازم برای شبیه‌سازی

الگوریتم بهینه‌سازی ازدحام ذرات	اندازه جمعیت اولیه	۱۰
	تعداد تکرار	۱۰۰
	C_1,C_2	۱.۴۹۴۴۵
	r1 و r2	عدد تصادفی بین ۰ تا ۱
	W	w=0.99*w, Initial w=1
الگوریتم تعادل بار	محدوده متوسط بار کاری	meanload±0.05*meanload

برای شبیه‌سازی و مقایسه نتایج، از بارکاری حقیقی استفاده شده است این بار کاری مربوط به مرکز تحقیقات ناسا[۱۹] می‌باشد که از اول ماه اکتبر[۲۰] تا ۳۱ دسامبر[۲۱] اندازه‌گیری شده است و شامل ۴۲۲۴۰ کار می باشد [۱۸].

هر کار موجود در این بار کاری شامل اطلاعات زیر می‌باشد:

- شناسه‌ی کار
- زمان ورود کار در سیستم
- زمان اجرای کار
- تعداد پردازنده‌های مورد نیاز برای یک کار
- تعداد کل پردازنده‌های مورد نیاز
- شناسه ی کاربری که کار را ایجاد کرده است
- شناسه‌ی گروه کاربرانی که کار را ایجاد کرده‌اند
- آرایه موقتی برای ذخیره‌ی فیلدهای هر کار

زمان اجرای هر کار به صورت متغیر از ۱ تا ۱۰۰۰۰ می‌باشد. برای استفاده از این فایل، آن را از ورودی گرفته و هر یک از کارها را به یک وظیفه در شبیه‌ساز کلودسیم به نام کلادلت تبدیل می‌کنیم. به این صورت که تعداد دستورالعمل‌های کلادلت را حاصلضرب زمان اجرای کار بر نرخ پردازنده[۲۲] در نظر گرفته و یک کلادلت را می‌سازیم. برای ارزیابی از چهار ماشین فیزیکی با مشخصات جدول ۵ استفاده می‌شود:

جدول ۵- مشخصات ماشین فیزیکی مورد استفاده

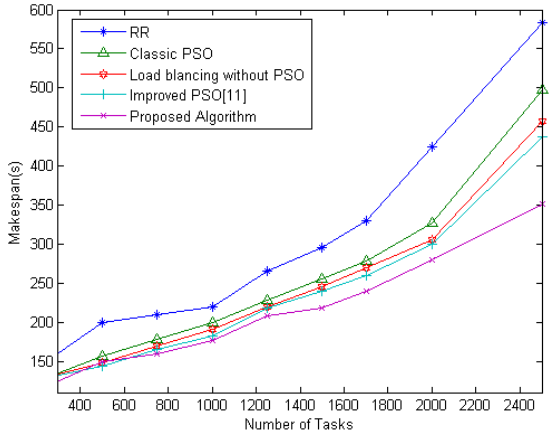
پردازنده	تعداد هسته‌های پردازنده	سرعت پردازنده (تعداد دستورالعمل در ثانیه)	حافظه داخلی (مگا بیت)	حافظه جانبی (مگا بیت)	پهنای باند (مگا بیت بر ثانیه)
Core_2_Extreme _X6800	۲	۲۷۰۷۹	۲۰۴۸۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
Core_i7_Extreme _Edition_X3960	۶	۱۷۷۷۳۰	۱۰۲۴	۱۰۴۸۵۷۶	۱۰۲۴۰۰
Core_i7_Extreme _Edition_980X	۶	۱۴۷۶۰۰	۲۰۴۸۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰
Core_i7_875K	۴	۹۲۱۰۰	۲۰۴۸۰	۱۰۴۸۵۷۶	۱۰۲۴۰۰

همچنین به تعداد ۴۰ ماشین مجازی با مشخصات جدول ۶ برای ارزیابی روش ارائه شده استفاده می‌کنیم.

جدول ۶- مشخصات ماشین‌های مجازی مورد استفاده برای ارزیابی

پردازنده	تعداد هسته‌های پردازنده	سرعت پردازنده (تعداد دستورالعمل در ثانیه)	حافظه داخلی (مگا بیت)	حافظه جانبی (مگا بیت)	پهنای باند (مگا بیت بر ثانیه)
Core_i4_Extreme_Edition	۱	۹۷۲۶	۵۱۲	۱۰۲۴۰	۱۰۲۴

در نمودار شکل ۲ با ثابت نگه داشتن تعداد ماشین مجازی به میزان ۴۰ ماشین مجازی و افزایش تعداد درخواست‌ها از ۳۰۰ تا ۲۵۰۰ درخواست مقدار زمان تکمیل تمامی کارهارا در روش ارائه شده با سایر روش‌ها مقایسه کرده‌ایم.



شکل ۲- مقایسه makespan بین روش‌های مختلف

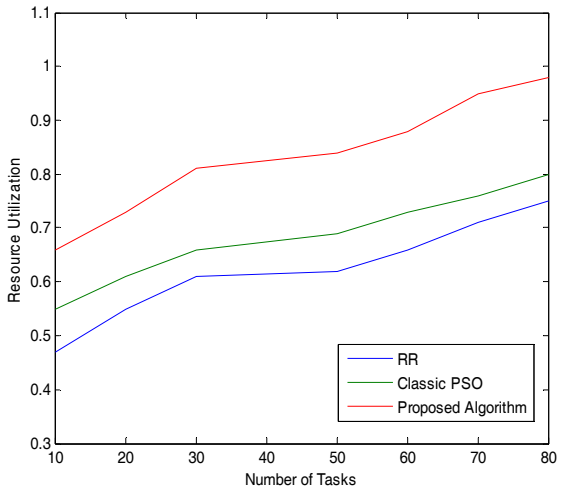
همانطور که در مجموعه‌ی نمودارهای شکل ۲ مشخص است استفاده از الگوریتم بهینه‌سازی ذرات باعث بهبود خوبی در زمان تکمیل تمامی کارها نسبت به روش پایه نوبت گردشی می‌شود. هم‌چنین استفاده از روش‌های تعادل بار نیز باعث کاهش زمان تکمیل تمامی کارها، نسبت به روش نوبت گردشی و روش پایه‌ی استفاده از الگوریتم بهینه‌سازی ذرات می‌شود. که این امر به دلیل تشخیص ماشین‌های مجازی دارای اضافه بار و کاهش بار آنها با جابه جایی بار بر روی ماشین‌های کم بار می باشد.

اما روش بهینه‌سازی ذرات که از ابتدا هر ذره در جایگاه مناسب قرار بگیرد از روش استفاده از الگوریتم تعادل بار مناسب تر است زیرا همانطور که در بخش قبل هم گفتیم کارایی الگوریتم تعادل بار وابسته به نحوه‌ی جایگزینی اولیه‌ی درخواست‌های آن مربوط می‌شود. در صورتی که درخواست‌ها را از ابتدا به‌صورت مناسب بر روی ماشین‌های مجازی قرار دهیم در هر لحظه جلوی افزونگی بار روی ماشین‌های مجازی گرفته می‌شود و بهبود قابل ملاحظه‌ای را نسبت به روش تعادل باری که جایگزینی اولیه درخواست‌ها به صورت نوبت گردشی باشد به دست می‌آوریم.

در روش پیشنهادی ابتدا در هر یک از ده ذره که یک جواب می‌باشد درخواست‌ها به‌صورت تصادفی بر روی ماشین‌ها قرار گرفته و سپس با استفاده از روش تعادل بار آنها را به‌منظور تعدیل بار و کاهش ماشین‌هایی که دارای افزونگی بار هستند، جابه‌جا می‌کنیم.

همانطور که در نمودار شکل ۲ مشخص است کارایی این روش با افزایش درخواست‌ها بهتر نیز می‌شود این امر به این دلیل می‌باشد که در درخواست‌های کم، بارکاری کم می‌باشد در نتیجه تعداد ماشین‌هایی که دارای افزونگی بار هستند، کمتر است، علاوه بر این مقدار زمان تکمیل تمامی کارها، بیشتر وابسته به تعداد درخواست‌ها و میزان زمان اجرای آن‌ها می‌باشد تا نحوه‌ی قراردهی درخواست‌ها بر روی ماشین‌های مجازی، در نتیجه با افزایش درخواست‌ها این بهبود به‌طور قابل ملاحظه افزایش می‌یابد.

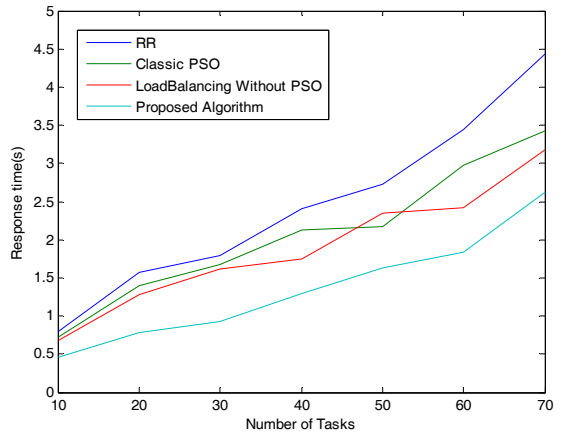
در ادامه به مقایسه‌ی بهره‌وری منابع بین روش‌های پایه، روش بهینه‌سازی ازدحام ذرات کلاسیک و روش بهینه‌سازی ازدحام ذرات بهبودیافته می‌پردازیم. شکل ۳ نمودار مقایسه‌ی بهره‌وری منابع را بین روش‌های بیان شده نشان می‌دهد. محور افقی در این نمودار تعداد درخواست‌ها و محور عمودی میزان بهره وری منابع می‌باشد.



شکل ۳- بهره‌وری منابع بین روش‌های PSO و PSO ترکیبی با روش تعادل بار

همانطور که از نمودار شکل ۳ مشخص است در هر دو روش با افزایش درخواست‌ها به دلیل تعادل بار ایجاد شده توسط هر یک از روش‌های فوق و افزایش بار کاری، بهره‌وری منابع به نسبت روش پایه بیشتر می‌شود، اما در روش بهینه‌سازی ذرات بهبود یافته نسبت به روش پایه چون میزان تعادل بار نسبت به روش پایه بیشتر است، میزان بهبود در بهره‌وری منابع بیشتر می‌شود.

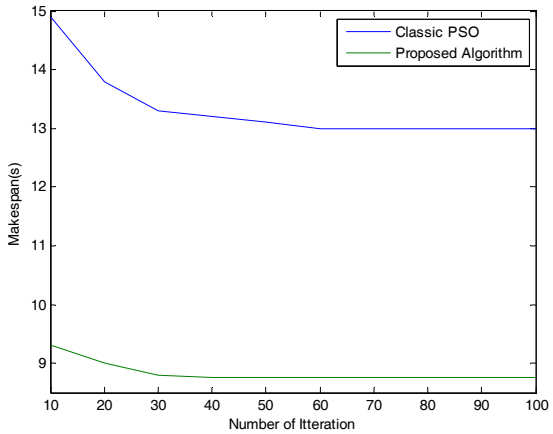
در ادامه متوسط زمان پاسخ را بین روش‌های مختلف مقایسه کرده‌ایم. شکل ۴ نمودار مقایسه‌ی متوسط زمان پاسخ را بین روش‌های بیان شده نشان می‌دهد. محور افقی در این نمودار تعداد درخواست‌ها و محور عمودی میزان متوسط زمان پاسخ می‌باشد.



شکل ۴- مقایسه زمان پاسخ بین روش‌های مختلف

نمودار شکل ۵ میزان همگرایی تا رسیدن به جواب مناسب را با تعداد تکرار در هر مرحله نشان می‌دهد، تعداد درخواست‌ها ۱۰۰ درخواست بوده و مشخصات شبیه‌سازی همانند شکل ۲ می‌باشد. همانطور که مشخص است، با افزایش تعداد مراحل تکرار، مقدار زمان تکمیل تمامی کارها، کاهش می‌یابد.

همان‌طور که در نمودارهای شکل ۵ مشخص است در روش بهینه‌سازی ازدحام ذرات چون جایگاه اولیه درخواست‌ها به صورت تصادفی انتخاب شده‌است با تعداد تکرار ۳۰ تا ۴۰ مرحله می‌توانیم به جواب مناسب دست یابیم. اما در روش بهبود یافته‌ی بهینه‌سازی ازدحام ذرات چون جایگاه اولیه درخواست‌ها به صورت مناسب انتخاب شده است، علاوه بر کاهش زمان تکمیل تمامی کارها با تعداد تکرار ۲۵ تا ۳۰ مرحله و به طور سریعتر می‌توانیم به جواب مناسب برسیم.



شکل ۵- رابطه‌ی همگرایی و تعداد تکرارها

۵- نتیجه‌گیری و کارهای آینده

در این مقاله ما یک روش زمان‌بندی کارها بر پایه ترکیب الگوریتم بهینه‌سازی ازدحام ذراتبه همراه الگوریتم تعادل بار ارائه کرده‌ایم. این الگوریتم نه تنها با متعادل کردن درخواست‌های ماشین‌های مجازی سبب کاهش زمان تکمیل تمامی کارها شده است، همچنین با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات توانسته با انتخاب تابع برازندگی مناسب، سبب افزایش بهره‌وری منابع شود. ما روش پیشنهادی خود را با روش پایه نوبت گردشی و هریک از الگوریتم‌های تعادل بار و الگوریتم بهینه‌سازی ازدحام ذرات به تنهایی مقایسه کرده‌ایم نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی بهبود خوبی را در زمان تکمیل تمامی کارها و افزایش بهره‌وری منابع می‌دهد. در آینده قصد داریم تا این کار را برای جریان گردش کار نیز انجام دهیم همچنین معیارهای دیگر مثل قابلیت تحمل خرابی برای کاربر و کاهش هزینه را برای فراهم کننده‌ی سرویس در نظر بگیریم.

مراجع

[1] K. Krishnasamy, "Task Scheduling Algorithm Based on Hybrid Particle Swarm Optimization in Cloud Computing Environment," Journal of theoretical & applied information technology, vol. 54, 2013.

[2] P. Kumar, and A. Verma, "Scheduling Using Improved Genetic Algorithm In Cloud Computing for Independent Tasks," Proc. International Conference on Advances in Computing, Communications and Informatics, pp. 137-142, 2012.

[3] C. Zhao, S. Zhang, Q. Liu, J. Xie, and J. Hu, "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing," 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'09), pp. 1-4, 2009.

[4] L. Wang, and L. Ai, "Task Scheduling Policy Based on Ant Colony Optimization in Cloud Computing

[16] P. V. Krishna, "Honey Bee Behavior Inspired Load Balancing of Tasks in Cloud Computing Environments," Applied Soft Computing, vol. 13, pp. 2292-2303, 2013.

[17] R. N. Calheiros, and et. al., "Cloudsim: A Toolkit For Modeling And Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software: Practice and Experience, vol. 41, pp. 23-50, 2011.

[18] D. G. Feitelson, and B. Nitzberg, "Job Characteristics of a Production Parallel Scientific Workload on the NASA Ames Ipsc/860," in Job Scheduling Strategies for Parallel Processing. Berlin, Germany: Springer Berlin Heidelberg, 1995, pp. 337-360.

[19] K. Gomathi, and B. Krishnasamy, "Task Scheduling Algorithm Base on Hybrid Particle Swarm Optimization in Cloud Computing Environment," Journal of Theoretical and Applied Information Technology, vol. 55, 2013.

فاطمه عبادی‌فرد مدرک کارشناسی مهندسی کامپیوتر گرایش نرم‌افزار را از دانشگاه قم اخذ کرده و در رشته مهندسی فناوری اطلاعات گرایش شبکه‌های کامپیوتری در مقطع کارشناسی‌ارشد، از دانشگاه علم و صنعت تهران فارغ‌التحصیل شده است و در حال حاضر دانشجوی دکتری مهندسی کامپیوتر در دانشگاه کاشان می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

f_ebadifardr@gmail.com

احمد اکبری دارای دکتری برق و کامپیوتر از دانشگاه رن فرانسه بوده و در حال حاضر دانشیار و عضو هیئت علمی دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران می‌باشد. ایشان پژوهش‌های فعالی را در زمینه‌های شبکه‌های کامپیوتری و امنیت و نیز پردازش سیگنال هدایت می‌کنند. همچنین مسئولیت ریاست دانشکده مهندسی کامپیوتر این دانشگاه و مرکز تحقیقات فناوری اطلاعات دانشگاه را نیز برعهده دارند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

akbari@iust.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۷/۲۰

تاریخ اصلاح: ۱۳۹۴/۰۹/۱۸

تاریخ قبول شدن: ۱۳۹۴/۱۰/۱۵

نویسنده مرتبط: فاطمه عبادی‌فرد، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران.

Environment," in LISS. Berlin, Germany: Springer Berlin Heidelberg, 2013, pp. 953-957.

[5] S. K. Chaharsooghi, and A. H. M. Kermani, "An Effective Ant Colony Optimization Algorithm (ACO) for Multi-Objective Resource Allocation Problem (MORAP)," Applied Mathematics and Computation, vol. 200, pp. 167-177, 2008.

[6] E. Pacini, C. Mateos, and C. G. Garino, "Balancing Throughput and Response Time in Online Scientific Clouds via Ant Colony Optimization," Advances in Engineering Software, vol. 84, pp. 31-47, 2015.

[7] L. Guo, S. Zhao, S. Shen, and C. Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm," Journal of Networks, vol. 7, pp. 547-553, 2012.

[8] L. Zhang, and et. al., "A Task Scheduling Algorithm Based on PSO for Grid Computing," International Journal of Computational Intelligence Research, vol. 4, pp. 37-43, 2008.

[9] S. Pandey, and et. al., "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," Proc. 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 400-407, 2010.

[10] S. Wang, and B. Meng, "Resource Allocation and Scheduling Problem Based on Genetic Algorithm and Ant Colony Optimization," in Advances in Knowledge Discovery and Data Mining. Berlin, Germany: Springer Berlin Heidelberg. 2007, pp. 879-886.

[11] S. Zhan, and H. Huo, "Improved PSO-based Task Scheduling Algorithm in Cloud Computing," Journal of Information & Computational Science, vol. 9, pp. 3821-3829, 2012.

[12] R. C. Eberhart, and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," Proc. sixth International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.

[13] G. Kaur, and E. S. Sharma, "Optimized Utilization of Resources Using Improved Particle Swarm Optimization Based Task Scheduling Algorithms in Cloud Computing," International Journal of Emerging Technology and Advanced Engineering, vol. 4, 2014.

[14] S. A. M. S. Abdi, and S. Sharifian, "Task Scheduling Using Modified PSO Algorithm in Cloud Computing Environment," Proc. International Conference on Machine Learning, Electrical and Mechanical Engineering, Dubai (UAE), 2014.

[15] K. A. Nuaimi, and et. al., "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," Proc. Second Symposium on Network Cloud Computing and Applications (NCCA), pp. 137-142, 2012.

[1] Paticle Swarm Optimization
[2] Makespan
[3] Infrastructure As A Service (Iaas)
[4] Platform As A Service (Paas)
[5] Software As A Service (SaaS)
[6] Multimedia Application Provider

[7] Resource Utilizatio
[8] Load Balancing
[9] Best Resource Selection
[10] Static
[11] Dynamic
[12] Shortest Job to Fastest Processor Algorithm
[13] Pbest
[14] Gbest
[15] Iteration
[16] Virtualization
[17] Makespan
[18] Iteration
[19] NASA Ames Research Center
[20] October
[21] December
[22] CPU Rating

علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۲۳-۳۱
مقاله عادی

افزایش طول‌عمر حافظه‌ی نهان سطح آخر غیرفرار با کمک بلوک‌های ذخیره

محمدرضا جوکار[۱و۳] محمد ارجمند[۲و۳] حمید سربازی‌آزاد[۳و۴]

[۱]دانشکده علوم کامپیوتر، دانشگاه شیکاگو، ایلینوی، امریکا
[۲]دانشکده مهندسی برق و کامپیوتر، دانشگاه ایالتی پنسیلوانیا، پنسیلوانیا، امریکا
[۳]دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران
[۴]پژوهشکده علوم کامپیوتر، پژوهشگاه دانش‌های بنیادی (IPM)، تهران، ایران

چکیده

تکنولوژی حافظه‌های غیرفرار همچون STT-RAM، دارای چگالی سلول بالا بوده و همچنین توان نشتی تقریبا صفر دارند. در نتیجه می‌توانند به عنوان یک جایگزین مناسب برای حافظه‌های نهان متداول امروزی همچون SRAM، در نظر گرفته شوند. در مقابل مزایای ذکر شده، این حافظه‌ها دارای قابلیت تحمل نوشتن محدود هستند که می‌تواند منجر به طول‌عمر پایین آنها شود. در این مقاله یک راهکار برای افزایش طول‌عمر این نوع حافظه‌ها ارائه می‌شود که مبتنی بر اضافه کردن بلوک‌های ذخیره به ازاء هر مجموعه از حافظه‌ی نهان است. با خرابی یک بلوک از یک مجموعه، راهکار پیشنهادی به صورت هوشمند و بدون تاثیر منفی بر کارایی سیستم، بلوک خراب را از آن مجموعه خارج کرده و یک بلوک ذخیره را به آن مجموعه اضافه می‌کند.

کلمات کلیدی: حافظه‌ی غیرفرار، STT-RAM، تحمل تعداد نوشتن، طول‌عمر.

۱- مقدمه

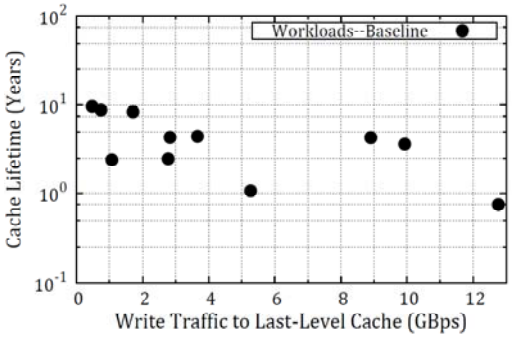
سیستم‌های محاسباتی مدرن امروزی نیازمند حافظه‌ی نهان روی تراشه‌ی بزرگ هستند تا بخش بیشتری از بلوک‌های مورد نیاز برنامه‌های در حال اجرا را در روی تراشه نگه داشته و نیاز به دسترسی خارج از تراشه (که دارای سربار کارایی و انرژی است) راکاهش دهند. نیاز به حافظه‌ی نهان روی تراشه‌ی بزرگ با افزایش تعداد هسته‌های پردازشی و بزرگ شدن مجموعه‌ی کاری برنامه‌های امروزی [۱]، رو به افزایش است. از طرفی مقیاس‌پذیری حافظه‌های نهان متداول امروزی همچون SRAM و DRAM نهفته بنا به دلایلی همچون چگالی سلول و توان نشتی آن‌ها با مشکل مواجه شده است. مطالعات نشان می‌دهد که بخشی زیادی از توان مصرفی حافظه‌ی نهان SRAM (حدود ۸۰٪) مربوط به توان نشتی است [۲]. در این راستا راهکارهای متعددی ارائه شده است که یکی از بهترین آن‌ها استفاده از تکنولوژی حافظه‌های غیرفرار[۱] به عنوان جایگزین مناسبی برای حافظه‌های نهان متداول امروزی است چرا که دارای توان نشتی تقریبا صفر و همچنین چگالی سلول بالا هستند.

در میان حافظه‌های غیرفرار، حافظه‌ی مغناطیس-جریان چرخش قطبی[۲] (STT-RAM) دارای ویژگی‌های جالبی برای جایگزینی حافظه‌ی نهان سطح آخر SRAM است.

حافظه‌ی STT-RAM دارای چگالی سلول بالا (حدود ۴ برابر SRAM [۳]) است. همچنین به عنوان سریع‌ترین حافظه‌ی غیرفرار در نظر گرفته می‌شود و زمان عملیات خواندن در آن تقریبا با SRAM برابری می‌کند. به علاوه، سلول‌های این نوع حافظه در مقایسه با دیگر حافظه‌های غیرفرار دارای بالاترین تحمل نوشتن[۳] (حدود 10^{12} [۴]) هستند.

اگرچه STT-RAM دارای بیشترین تحمل نوشتن بین حافظه‌های غیرفرار است، اما همچنان تحمل نوشتن آن در مقایسه با SRAM ($3 * 10^{16}$) بسیار کم‌تر است و تحمل نوشتن محدود سلول‌های آن، می‌تواند منجر به طول‌عمر پایین در این نوع حافظه‌ها شود. همچنین بررسی‌های انجام شده بر روی الگوی نوشتن در

حافظه‌ی نهان سطح آخر، نشان می‌دهد که ترافیک نوشتن وارد شده به بلوک‌های مختلف آن، یکسان نبوده و اختلاف تعداد نوشتن بین بلوک‌های مجموعه‌های مختلف حافظه‌ی نهان[۴] (InterV) و همچنین بین بلوک‌های یک مجموعه از آن[۵] (IntraV) وجود دارد [۵]. در نتیجه بلوک‌هایی که ترافیک نوشتن بیشتری را دریافت می‌کنند بسیار سریع‌تر از دیگر بلوک‌ها به حد نوشتن قابل تحمل خود رسیده و با خرابی زودهنگام خود موجب کاهش کارایی و خرابی حافظه‌ی نهان می‌شوند. همان‌طور که شکل ۱ نشان می‌دهد، حافظه‌ی نهان سطح آخر STT-RAM با حجم ۲ مگابایت، در برنامه‌های وابسته به حافظه که به سیستم حافظه استرس وارد می‌کنند (جزئیات پیکربندی سیستم و روش‌های ارزیابی را در فصل ۵ مشاهده بفرمائید.)، می‌تواند طول‌عمری در حد چندین ماه داشته باشد.



شکل ۱- طول‌عمر حافظه‌ی نهان سطح آخر غیرفرار

از دیگر مشکلات حافظه‌های STT-RAM زمان نوشتن طولانی و توان مصرفی نوشتن بالا است که کارهای گذشته به صورت گسترده به بررسی آن پرداخته‌اند [۶-۹] اما تاکنون توجه کمی به مشکل تحمل نوشتن محدود و طول‌عمر پایین در این نوع حافظه‌ها شده است. کارهای انجام شده با هدف افزایش طول‌عمر، تلاش در یکنواخت‌سازی ترافیک نوشتن وارد شده به بلوک‌های مختلف حافظه‌ی نهان دارند؛ در نتیجه نیازمند ایجاد تغییراتی در عملکرد معمول حافظه‌ی نهان هستند. با توجه به این‌که زمان دسترسی در حافظه‌ی نهان از اهمیت ویژه‌ای برخوردار است، تغییرات ایجاد شده در آن بایستی به طور هوشمند باشد چرا که در غیر این صورت موجب افت شدید کارایی سیستم و همچنین افزایش مصرف انرژی خواهد شد. متاسفانه، تکنیک‌هایی که به یکنواخت‌سازی ترافیک نوشتن در سطح حافظه‌ی نهان می‌پردازند [۵ و ۱۰]، با تغییرات غیرهوشمند و در نتیجه سربارهای غیرقابل تحمل از جهت کارایی، همراه هستند.

در مقاله‌ی [۱۰]، نرخ نوشتن به بلوک‌های داده و همچنین مجموعه‌های حافظه‌ی نهان در جدول‌هایی نگه داشته می‌شود و جابه‌جایی داده بین بلوک‌های سرد و داغ درون یک مجموعه و همچنین جابه‌جایی نگاشت آدرس بین مجموعه‌های سرد و داغ انجام می‌شود. این روش نیازمند جابه‌جایی‌های مکرر داده است و در نتیجه موجب افزایش تعداد نوشتن‌ها در حافظه‌ی نهان می‌شود که این خود می‌تواند عامل کاهش طول‌عمر بوده و همچنین موجب مصرف انرژی زیاد، اختلال در دسترسی به حافظه‌ی نهان به دلیل افزایش تعداد نوشتن‌ها (عملیات نوشتن در حافظه‌های غیرفرار کند بوده و می‌تواند موجب مسدود شدن عملیات خواندنی که به دنبال آن می‌آیند، شود) و همچنین عدم قطعیت در زمان دسترسی به حافظه‌ی نهان شود. به‌علاوه سربار ذخیره‌سازی این روش زیاد است. مقاله‌ی [۵] که مدرن‌ترین مقاله در زمینه‌ی افزایش طول‌عمر حافظه‌های نهان غیر فرار است، نوشتن اضافی به حافظه‌ی نهان تحمیل نمی‌کند و همچنین سربار ذخیره‌سازی کمی دارد اما به دلیل پس‌نویسی‌های مکرر حافظه‌ی نهان سطح آخر به حافظه‌ی اصلی، موجب هدر رفتن پهنای‌باند حافظه‌ی اصلی و همچنین افزایش نرخ نقصان حافظه‌ی نهان سطح آخر می‌شود. استفاده از این روش در برنامه‌های وابسته به حافظه، می‌تواند منجر به کاهش شدید کارایی و افزایش مصرف انرژی سیستم شود؛ درنتیجه گزینه‌ی مناسبی برای این نوع برنامه‌ها نیست.

در این مقاله یک روش نوین مبتنی بر اضافه کردن بلوک‌های ذخیره، برای افزایش طول‌عمر حافظه‌های نهان غیرفرار ارائه شده است. در این روش، به ازاء هر مجموعه از حافظه‌ی نهان غیرفرار، تعدادی بلوک ذخیره وجود دارد؛ با خرابی بلوک‌های یک مجموعه، یکی از بلوک‌های ذخیره‌ی آن، به صورت هوشمند به آن مجموعه اضافه شده و بلوک خراب از آن مجموعه حذف می‌شود. تا قبل از خرابی بلوک‌های حافظه‌ی نهان، روش پیشنهادی تغییری در عملکرد معمول حافظه‌ی نهان ایجاد نمی‌کند و تاثیری بر روی کارایی سیستم ندارد. همچنین با توجه به این‌که توان نشتی در حافظه‌های غیرفرار تقریبا صفر است، وجود بلوک‌های ذخیره که هنوز به حافظه‌ی نهان اضافه نشده‌اند، اثری بر روی توان سیستم نخواهد داشت. پس از خرابی بلوک‌های حافظه‌ی نهان، تغییرات جزئی در حافظه‌ی نهان ایجاد می‌شود، اما این تغییرات اثری بر روی تاخیر عملیات خواندن (که در مسیر بحرانی قرار دارد) و در نتیجه کارایی سیستم ندارد. همچنین این تغییرات موجب افزایش تاخیر عملیات نوشتن نیز نمی‌شوند.

سربار این روش، مساحت اشغالی بلوک‌های ذخیره است. طبق آزمایشات صورت گرفته در این مقاله، مساحت اشغالی حافظه‌ی نهان سطح آخر مورد ارزیابی قرار گرفته با حجم دو مگابایت از نوع SRAM، ۴۶.۶۱ میلی‌متر مربع است در حالی که مساحت اشغالی حافظه‌ی نهان غیرفرار STT-RAM با همین حجم، ۱۴ برابر کمتر یعنی ۳.۲۱ میلی‌متر مربع است. با توجه به این‌که بهبود مساحت حاصل شده در استفاده از حافظه‌های غیرفرار زیاد است، می‌توان بخشی از آن را صرف افزایش طول‌عمر آن‌ها کرد.

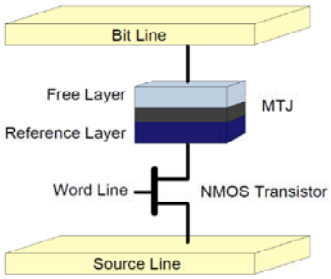
ادامه‌ی این مقاله به این صورت است که ابتدا در بخش ۲ به بررسی حافظه‌ی STT-RAM پرداخته و در بخش ۳ کارهای پیشین مرتبط با طول‌عمر این نوع حافظه‌ها مورد بررسی قرار می‌گیرد. در بخش ۴ روش پیشنهادی به صورت دقیق ارائه می‌شود. بخش ۵ به ارزیابی روش پیشنهادی و مقایسه‌ی آن با کارهای پیشین پرداخته و بخش ۶ به کارهای آینده و جمع‌بندی مقاله اختصاص می‌یابد.

۲- حافظه‌ی STT-RAM

المان اصلی در حافظه‌های مغناطیسی با زمان دسترسی تصادفی[۶] (MRAM)، اتصال تونل مغناطیسی[۷] (MTJ) است که برای ذخیره‌ی مقدار دودویی استفاده می‌شود. بر خلاف حافظه‌های معمول که از شارژ الکتریکی به عنوان حامل اطلاعات استفاده می‌کنند، MTJ براساس ذخیره‌ی مغناطیسی کار می‌کند؛ MTJ از دو لایه‌ی فرومغناطیسی و یک لایه‌ی تونل مانع[۸] تشکیل شده است. جهت مغناطیسی یکی از لایه‌های فرومغناطیسی ثابت است (لایه‌ی مرجع[۹]) در حالی‌که جهت لایه‌ی فرومغناطیسی دیگر، می‌تواند با اعمال یک جریان تغییر کند (لایه‌ی آزاد[۱۰]). جهت مغناطیسی نسبی این دو لایه‌ی فرومغناطیسی، مقاومت MTJ را تعیین می‌کند. زمانی که دو لایه جهت یکسان دارند مقاومت MTJ کم است و بیانگر مقدار دودویی یک است؛ در صورتی که جهت دو لایه متفاوت باشد، مقاومت MTJ زیاد است و بیانگر مقدار دودویی صفر است [۳].

حافظه‌ی STT-RAM، نسل دوم حافظه‌های MRAM بوده و شهرت زیادی به دلیل مقیاس‌پذیری بالا، سرعت بالا و مصرف توان پایین دارد. معروف‌ترین ساختار سلول‌های STT-RAM، ساختار یک ترانزیستور و یک MTJ است. در این ساختار همانطور که در شکل ۲ نشان داده شده است، MTJ با یک ترانزیستور NMOS سری شده است. ترانزیستور NMOS در زمان خواندن و نوشتن، روشن می‌شود. برای عمل خواندن، یک اختلاف ولتاژ منفی بین خط بیت[۱۱] و خط منبع[۱۲] برقرار می‌شود. این ولتاژ منفی که مقدار آن خیلی کوچک است، باعث ایجاد جریان

در MTJ می‌شود. مقدار جریان به وجود آمده عموماً به مقاومت MTJ بستگی دارد. یک تقویت کننده‌ی حسی[۱۳]، این جریان را با یک جریان مرجع مقایسه می‌کند و تشخیص می‌دهد که مقدار دودویی ذخیره شده در سلول صفر یا یک است. مقدار اختلاف ولتاژ اعمال شده در عمل خواندن، باید به قدری کم باشد که موجب نوشتن ناخواسته نشود. در عمل نوشتن، زمانی که می‌خواهیم مقدار دودویی صفر را بنویسیم، یک اختلاف ولتاژ مثبت بین خط منبع و خط بیت اعمال می‌کنیم و زمانی که می‌خواهیم مقدار یک را بنویسیم، اختلاف ولتاژ منفی اعمال می‌کنیم. جریان مورد نیاز برای تغییر جهت لایه فرومغناطیسی آزاد، به اندازه MTJ و مدت زمان پالس نوشتن بستگی دارد. به هر اندازه که اندازه MTJ کوچک‌تر باشد و مدت زمان پالس نوشتن طولانی‌تر باشد، جریان کمتری مورد نیاز است [۳].



شکل ۲- یک سلول حافظه‌ی STT-RAM

در آخرین تکنولوژی حافظه‌های MRAM یعنی STT-RAM، جهت مغناطیسی لایه‌ی آزاد از طریق گذراندن یک جریان چرخشی قطبی از MTJ به دست می‌آید. در مقایسه با نسل قبلی حافظه‌های MRAM که از یک میدان مغناطیسی خارجی برای تغییر وضعیت MTJ استفاده می‌کردند، STT-RAM دارای مقیاس‌پذیری بیشتری است و حداقل جریان مورد نیاز برای تغییر وضعیت MTJ، با کاهش اندازه‌ی MTJ کاهش می‌یابد [۳].

هر سلول STT-RAM، قابلیت تحمل تعداد مشخصی عمل نوشتن (حدود 10^{12} [۲]) را در خود دارد و پس از آن، لایه‌ی آزاد قابلیت تغییر جهت مغناطیسی را از دست می‌دهد و در نتیجه موجب از کار افتادن سلول می‌شود.

۳- کارهای پیشین

در این بخش به کارهای مرتبط پیشین در زمینه‌ی استفاده از حافظه STT-RAM در سلسله مراتب حافظه‌ی نهان می‌پردازیم و اثر هر کار بر روی طول‌عمر این نوع حافظه‌ها را بررسی می‌کنیم. کارهای انجام شده در زمینه‌ی STT-RAM را می‌توان به سه دسته‌ی کلی برای حل مشکلات انرژی مصرفی نوشتن بالا، زمان نوشتن بالا و طول‌عمر پایین تقسیم کرد که در ادامه به آن‌ها پرداخته می‌شود.

۳-۱- کاهش انرژی مصرفی نوشتن

برخی کارهای پیشین همچون مقاله‌ی [۹]، به کاهش انرژی مصرفی عملیات نوشتن در حافظه‌های STT-RAM پرداخته‌اند.در این مقاله، یک تکنیک سطح مدار برای کاهش انرژی نوشتن در این نوع حافظه‌ها ارائه شده است که اثر منفی روی کارایی سیستم ندارد. این مقاله در ارائه‌ی ایده خود، از این واقعیت که در عمل نوشتن احتمال زیادی وجود دارد که محتوای سلول عوض نشود، استفاده می‌کند؛ قبل از هر عمل نوشتن، یک بار عمل خواندن انجام می‌شود و تنها در صورتی که مقدار قبلی سلول با مقدار جدید متفاوت باشد، عمل نوشتن ادامه می‌یابد، در غیر این صورت عمل نوشتن قطع می‌شود. با توجه به این‌که زمان نوشتن بسیار طولانی‌تر از زمان خواندن است و تغییر در مقاومت MTJ در انتهای پالس نوشتن انجام می‌شود، سلول در ابتدای پالس، مقدار قبلی خود را دارد؛ پس با محاسبه مقاومت MTJ از طریق جریان جاری شده در آن می‌توان در ابتدای پالس نوشتن، مقدار قبلی سلول را اندازه گرفت و در صورت یکسان بودن با مقدار جدید عمل نوشتن را متوقف کرد. به این صورت تعداد عملیات نوشتن کاهش می‌یابد و انرژی مصرفی حاصل از نوشتن و در نتیجه انرژی کل سیستم کاهش می‌یابد. این تکنیک با هدف افزایش طول‌عمر ارائه نشده اما با کاهش تعداد دفعات نوشتن، تا حدی موجب افزایش طول‌عمر حافظه‌ی نهان غیرفرار می‌شود.

۳-۲- افزایش کارایی

در کارهایی همچون مقاله‌ی [۷]، به حل مشکل زمان نوشتن بالا در حافظه‌های STT-RAM پرداخته شده است. این مقاله، یک تکنیک سطح معماری برای افزایش کارایی سیستم‌هایی که از حافظه‌ی STT-RAM در حافظه نهان سطح آخر استفاده می‌کنند، ارائه می‌دهد. در حافظه‌ی نهان سطح آخر STT-RAM، یک عمل نوشتن به دلیل تاخیر زیاد، می‌تواند موجب مسدود شدن پورت حافظه و تاخیر عملیات خواندنی که به دنبال آن می‌آیند، شود و با توجه به این که عملیات خواندن در مسیر بحرانی قرار دارند، کارایی سیستم به شدت پایین می‌آید. به علاوه زمانی که حافظه STT-RAM در سیستم‌های چند هسته‌ای به صورت مشترک استفاده می‌شود، یک عمل نوشتن درخواست شده توسط یک هسته، می‌تواند عملیات خواندن هسته‌های دیگر را نیز به تاخیر بیاندازد و به این طریق کارایی کل سیستم را به شدت پایین بیاورد. در معماری مطرح شده در آن مقاله یک نظارت روی درخواست‌هایی که به حافظه نهان سطح آخر می‌آیند، انجام می‌شود و برای هر پردازش در سیستم، دو پارامتر محاسبه می‌شود. یکی زمان دسترسی دستورات در صورتی که درخواست‌های نوشتن به حافظه نهان سطح آخر را انجام دهیم و دیگری زمان دسترسی دستورات، در صورتی که درخواست‌های نوشتن را مستقیما به حافظه‌ی اصلی هدایت کنیم. در صورتی که مقدار پارامتر اول از پارامتر دوم بیشتر باشد، به این معنی است که نوشتن داده‌های این پردازش در حافظه‌ی نهان سطح آخر، نه تنها سودی ندارد بلکه باعث مسدود شدن عملیات خواندن پس از آن می‌شود. چنین پردازش‌هایی به عنوان پردازش مسدودکننده علامت زده می‌شوند و داده‌های آن‌ها در حافظه نهان سطح آخر نوشته نمی‌شود و مستقیما از حافظه نهان سطح بالاتر به حافظه اصلی و یا برعکس هدایت می‌شود. این تکنیک با هدف افزایش کارایی ارائه شده اما با توجه به این که تعداد عملیات نوشتن در حافظه‌ی نهان سطح آخر را کم می‌کند، تا حدی موجب افزایش طول‌عمر آن می‌شود.

۳-۳-افزایش طول‌عمر

کارهایی که تا کنون بررسی شدند، دارای هدف اصلی افزایش طول‌عمر نبودند اما تا حدی به آن کمک می‌کردند. در این بخش کارهای با هدف اصلی افزایش طول‌عمر، با انجام یکنواخت‌سازی عملیات نوشتن، مورد بررسی قرار می‌گیرند.

در مقاله‌ی [۱۰]، به ازاء هر بلوک و همچنین به ازاء هر مجموعه از حافظه‌ی نهان غیرفرار، یک شمارنده وجود دارد که با هر عمل نوشتن بهآن بلوک یا مجموعه یکی می‌شمارد. برای کاهش IntraV، محتوای بلوک‌های داغ (دارای شمارنده‌ی بلوک اشباع‌شده) و بلوک‌های سرد (دارای شمارنده‌ی بلوک با کمترین مقدار) در هر مجموعه جابه‌جا می‌شود. برای کاهش InterV، بخشی از بلوک‌های مجموعه‌های داغ (دارای شمارنده‌ی مجموعه‌ی اشباع‌شده) به مجموعه‌های سرد (دارای شمارنده‌ی مجموعه با کمترین مقدار) فرستاده می‌شود. این تکنیک با

جابه‌جایی‌های متعدد همراه بوده و تا حد زیادی تعداد نوشتن درون حافظه‌ی نهان غیرفرار را افزایش می‌دهد. افزایش عملیات زمان‌بر نوشتن در حافظه‌های غیرفرار، موجب تاخیر عملیات خواندنی که به دنبال آن‌ها می‌آیند شده و می‌تواند موجب کاهش شدید کارایی در برنامه‌های وابسته به حافظه شود. ضمن این‌که سربار شمارنده‌های این روش نیز از نظر تاخیر، مساحت اشغالی و انرژی مصرفی زیاد است.

در مقاله‌ی [۵]، روش i2WAP با دو تکنیک برای یکنواخت‌سازی تعداد نوشتن‌ها در حافظه‌های نهان غیرفرار، یکی برای کاهش InterV و دیگری برای کاهش IntraV، ارائه شده است. در تکنیک بین مجموعه، یک شمارنده به ازاء کل حافظه‌ی نهان سطح آخر وجود دارد و با هر عمل نوشتن در آن، یکی می‌شمارد؛ زمانی که این شمارنده به یک مقدار آستانه رسید، یک جابه‌جایی نگاشت بین دو مجموعه‌ی مجاور در حافظه‌ی نهان غیرفرار، انجام می‌شود اما محتوای این دو مجموعه با هم جابه‌جا نمی‌شوند؛ تمام بلوک‌های آن دو مجموعه نامعتبر شده و در صورت کثیف بودن، به حافظه‌ی سطح پایین‌تر پس‌نویسی می‌شوند و از این پس، درخواست‌های هرکدام ازاین دو مجموعه به دیگری هدایت می‌شوند. در تکنیک درون مجموعه، به ازاء کل حافظه‌ی نهان سطح آخر، یک شمارنده وجود دارد که با هر اصابت نوشتن، یکی می‌شمارد و زمانی که مقدار شمارنده به یک مقدار آستانه رسید، اصابت نوشتن بعدی انجام نمی‌شود؛ بلوک مورد اصابت نامعتبر شده و درخواست نوشتن به حافظه‌ی سطح پایین‌تر هدایت می‌شود. از آنجا که به احتمال زیاد این بلوک داغ بوده و با پذیرفتن عملیات اصابت نوشتن متوالی، موجب اشباع شدن شمارنده شده است، با ارسال آن به حافظه‌ی سطح پایین‌تر، این فرصت را می‌یابد که در صورت نیاز مجدد و آورده شدن به سطح حافظه‌ی فعلی، در یک محل فیزیکی دیگر در مجموعه‌ی نگاشت‌شده‌ی خود قرار بگیرد.

در این روش که مدرن‌ترین روش برای یکنواخت‌سازی ترافیک نوشتن در سطح حافظه‌ی نهان است، نوشتن‌های اضافی بر روی حافظه‌ی نهان وجود ندارد در عین حال که سربار ذخیره‌سازی کمی نیز دارد؛ اما از چند جهت سربارهای غیرقابل تحملی را به طرح نهایی اعمال می‌کند. این روش پهنای باند محدود حافظه اصلی را به میزان قابل توجهی هدر می‌دهد و همچنین مصرف انرژی زیادی را به دلیل پس‌نویسی مکرر بلوک‌های داده به حافظه‌ی اصلی متحمل می‌شود.

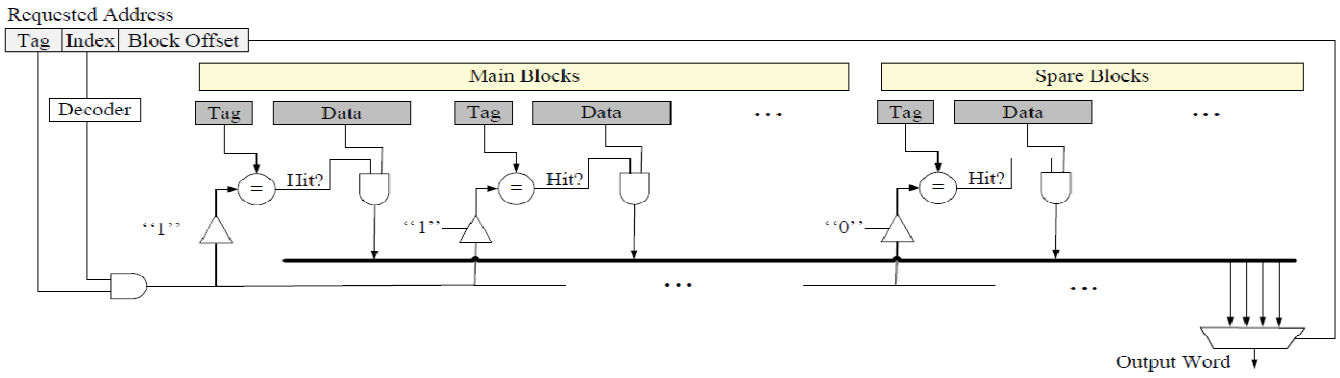
در تکنیک درون مجموعه، بلوک‌های داغ به حافظه‌ی اصلی پس‌نویسی می‌شوند؛ در مشاهداتی که ما انجام دادیم، در سیستم با دو سطح حافظه‌ی نهان (جزئیات پیکربندی را در بخش ۵ مشاهده بفرمایید.)، درصد زیادی از این بلوک‌ها (به طور میانگین ۹۷.۷٪) به دلیل داغ بودن، در آینده‌ی نزدیک درخواست شده (به طور میانگین در حدود ۴۰۰۰ سیکل پردازنده) و نرخ نقصان حافظه‌ی نهان سطح آخر را تا ۳۶.۷ ٪ (۱۳.۵ ٪ به طور میانگین) نسبت به سیستم پایه افزایش می‌دهند. افزایش نرخ نقصان حافظه‌ی نهان سطح آخر، نه تنها موجب کاهش کارایی سیستم شده بلکه موجب هدر رفتن پهنای باند محدود حافظه‌ی اصلی و مصرف انرژی نیز می‌شود. همچنین علاوه بر افزایش نرخ نقصان، این تکنیک پس‌نویسی به حافظه‌ی اصلی را تا ۴۴.۸ ٪ (۱۷.۷٪ به طور میانگین) نسبت به سیستم پایه افزایش می‌دهد و از این جهت نیز موجب هدر رفتن پهنای باند حافظه‌ی اصلی می‌شود.

تکنیک بین مجموعه، با توجه به این‌که در هر جابه‌جایی نگاشت، تمامی بلوک‌های دو مجموعه را نامعتبر کرده و در صورت کثیف بودن به حافظه‌ی اصلی پس‌نویسی می‌کند، می‌تواند نرخ نقصان حافظه‌ی نهان سطح آخر و همچنین پس‌نویسی به حافظه‌ی اصلی را به میزان زیادی افزایش دهد. در این مقاله، میزان آستانه‌ی شمارنده‌ی بین مجموعه عدد بسیار بزرگی در نظر گرفته شده است که موجب کاهش نرخ جابه‌جایی‌های نگاشت می‌شود. نویسندگان این مقاله برای توجیه داشتن این شمارنده‌ی بزرگ، بیان می‌کنندکه هر چند این شمارنده در زمان شبیه‌سازی موجب می‌شود که تعداد ناچیزی جابه‌جایی نگاشت انجام شود اما در دراز مدت اثر خود را نشان خواهد داد.

حتی با در نظر گرفتن شمارنده‌ی بزرگ برای تکنیک بین مجموعه، همچنان این روش سربار زیادی از نظر نرخ نقصان حافظه‌ی نهان سطح آخر و پس‌نویسی به حافظه‌ی اصلی دارد (سربار تکنیک درون مجموعه) که یکی از موانع اساسی در جهت به‌کارگیری آن برای برنامه‌های وابسته به حافظه است. همچنین بایستی توجه داشت که با داشتن شمارنده‌ی بزرگ برای تکنیک بین مجموعه، در صورت وجود برنامه‌های بدخواه که به صورت مکرر در یک آدرس مشخص از فضای حافظه عمل نوشتن را انجام دهند، طول‌عمر حافظه‌ی نهان با چالش جدی مواجه خواهد شد. در نتیجه برای داشتن الگوریتم یکنواخت‌سازی با بهره‌وری مناسب، نمی‌توان شمارنده‌ی بزرگی برای تکنیک بین مجموعه در نظر گرفت. شمارنده‌ی کوچک از جهت دیگر، موجب می‌شود نرخ پس‌نویسی به حافظه‌ی اصلی به شدت افزایش یافته و غیرقابل تحمل شود.

۴- راهکار پیشنهادی

راهکار پیشنهادی این مقاله برای افزایش طول‌عمر حافظه‌های نهان غیرفرار، اضافه کردن بلوک‌های ذخیره به ازاء هر مجموعه است به گونه‌ای که این بلوک‌های ذخیره موجب افزایش مصرف توان و کاهش کارایی نشوند. فرض کنید که هر مجموعه از حافظه‌ی نهان، دارای x بلوک اصلی بوده و ما در روش خود به ازاء هر مجموعه، y بلوک ذخیره قرار داده‌ایم (S = y)؛ در این صورت می‌توان گفت که یک حافظه‌ی نهان با x + y بلوک به ازاء هر مجموعه داریم که در ابتدا تنها x بلوک اصلی قابل دسترسی است و بقیه بلوک‌ها تنها در صورت خرابی یک بلوک از مجموعه، به آن اضافه می‌شوند.



شکل ۳- معماری یک مجموعه از حافظه‌ی نهان غیرفرار در راهکار پیشنهادی

اولین سؤالی که ممکن است به ذهن برسد این است که با توجه به چگالی سلول زیاد در حافظه‌های غیرفرار و امکان ساخت حافظه‌های نهان با حجم بالا، چرا از همان ابتدا یک حافظه‌ی نهان با x + y بلوک اصلی به ازاء هر مجموعه نداشته باشیم. نکته‌ای که وجود دارد این است که با اضافه شدن تعداد بلوک‌ها به ازاء هر مجموعه، هزینه‌ی انرژی و تاخیر جستجوی یک بلوک در آن مجموعه افزایش می‌یابد. در نتیجه با وجود امکان ساخت از نظر مساحت، نمی‌توان تا حد زیادی تعداد بلوک‌های هر مجموعه را افزایش داد.

نکته‌ی مهم بعدی که نیاز به بررسی دارد توان مصرفی بلوک‌های ذخیره در زمانی که هنوز به مجموعه اضافه نشده‌اند است. در حافظه‌های نهان SRAM، با توجه به اینکه بخش زیادی از مصرف توان مربوط به توان نشتی است، برخی کارهای گذشته همچون مقاله‌ی [۱۱]، در مجموعه‌هایی که از بلوک‌های خود بهره‌ی کافی نمی‌برند، اقدام به خاموش کردن برخی بلوک‌ها برای کاهش توان نشتی می‌کنند. مقاله‌ی [۱۱] از تکنیک سطح مدار ارایه شده در مقاله‌ی [۱۲] برای خاموشی سلول‌های حافظه SRAM در زمان لازم استفاده می‌کند. با توجه به ساختار سلول حافظه‌ی SRAM که از دو گیت معکوس‌کننده تشکیل شده است که برای حفظ مقدار ذخیره شده در خود به منبع تغذیه وصل هستند، این حافظه حتی در حالت ایستا مصرف توان نشتی زیادی دارد.

مقاله‌ی [۱۲] برای خاموشی سلول‌های حافظه‌ی SRAM از یک ترانزیستور برای قطع ارتباط دو گیت معکوس‌کننده با منبع تغذیه استفاده می‌کند. مسلماً با قطع ارتباط با منبع تغذیه مقدار ذخیره شده در سلول دیگر قابل دسترس نیست اما به این طریق می‌توان باعث قطع مصرف توان نشتی بلوک‌های حافظه در زمانی که کارایی لازم را ندارند شد. در نتیجه در صورت داشتن بلوک‌های ذخیره در حافظه‌های SRAM بایستی آن‌ها را خاموش کرده و در زمان نیاز روشن کرد. اما در حافظه‌های نهان غیرفرار بر خلاف حافظه‌های نهان SRAM، توان نشتی صفر است و در نتیجه در راهکار پیشنهادی این مقاله نیازی به عملیات اضافی (همچون خاموشی بلوک‌های ذخیره) برای قطع مصرف توان نشتی نیست و این حافظه‌ها در حالت ایستا توان مصرف نمی‌کنند. همچنین با توجه به این که این بلوک‌ها هنوز به حافظه‌ی نهان اضافه نشده‌اند، موجب افزایش مصرف توان پویا نیز نمی‌شوند.

در ادامه عملکرد راهکار پیشنهادی در سناریوهای مختلف خواندن و نوشتن مورد بررسی قرار داده می‌شود:

اصابت خواندن: شکل ۳، معماری یک مجموعه از حافظه‌ی نهان پس از اضافه کردن بلوک‌های ذخیره را نشان می‌دهد. درخواست خواندن در ساختار معمول حافظه‌ی نهان، به این صورت است که ابتدا به کمک فیلد اندیس آدرس درخواستی، مجموعه‌ی مورد نظر انتخاب می‌شود؛ پس از آن، نشانه‌ی آدرس درخواستی با نشانه‌ی بلوک‌های آن مجموعه مقایسه شده و در صورت برابر بودن، یک اصابت رخ می‌دهد. سپس توسط فیلد آفست آدرس،کلمه‌ی درخواستی جدا شده و در اختیار پردازنده قرار می‌گیرد. با توجه به این که این عملیات در مسیر بحرانی قرار دارد، اضافه شدن بلوک‌های ذخیره نبایستی اثر منفی بر روی تاخیر این عملیات داشته باشد.

بدین منظور از بافرهای سه‌حالته به ازاء هر بلوک استفاده می‌کنیم؛ به این صورت که در زمان بررسی نشانه‌ی آدرس درخواستی، در صورتی که بافرهای سه‌حالته‌ی بلوکی فعال باشد، نشانه‌ی آن بلوک مقایسه می‌شود؛ در غیر این صورت مقایسه انجام نشده و گویی بلوک مورد نظر در ساختار آن مجموعه وجود ندارد. در ابتدا، بافرهای سه‌حالته در بلوک‌های اصلی فعال هستند و پس از خرابی یک بلوک از یک مجموعه، بافرهای آن غیرفعال شده و بافرهای یک بلوک ذخیره فعال می‌شود. در نتیجه بلوک‌های یک مجموعه به طور هوشمند به آن اضافه شده یا از آن خارج می‌شوند و همیشه به تعداد بلوک‌های اصلی جستجو انجام می‌شود.

جدول ۱- پیکربندی سیستم پایه

Cores	4-cores, Alpha ISA, out-of-order, 3GHz
Coherency	MOESI directory
L1 caches	32KB, private, 8-way, 64B, LRU, write-back, hit latency: 2 cycles
L2 cache	2MB STT-RAM, UCA, shared, non-inclusive, 8-way, 64B, LRU, write-back, 4banks, 8 entries write buffer, read latency: 11 cycles, write latency: 19 cycles
DRAM	4GB, 128-entry write buffer, 200-cycle

به عنوان مثال فرض کنید که در یک مجموعه ۸ بلوک اصلی و ۴ بلوک ذخیره وجود دارد. در ابتدا بافرهای سه حالته در ۸ بلوک اصلی فعال و در ۴ بلوک ذخیره غیرفعال هستند. زمانی که یک درخواست خواندن برای این مجموعه وجود دارد، با توجه به شکل ۳، نشانه‌ی آدرس درخواستی با نشانه‌های بلوک‌هایی که داری بافر سه حالته فعال هستند (۸ بلوک اصلی) به طور موازی مقایسه می‌شود و در صورت موجود بودن داده درخواستی، یک اصابت خواندن رخ می‌دهد. حال در صورتی که یک بلوک اصلی خراب شود، بافر سه حالته آن غیر فعال شده و بافر سه حالته یک بلوک ذخیره فعال می‌شود. از این پس در زمان درخواست خواندن، نشانه‌ی بلوک اصلی خراب شده با نشانه‌ی آدرس درخواستی مقایسه نمی‌شود و در عوض نشانه‌ی بلوک ذخیره اضافه شده مقایسه می‌شود. در نتیجه ما همیشه نشانه‌های ۸ بلوک را با نشانه‌ی آدرس درخواستی به طور موازی مقایسه می‌کنیم و اضافه کردن بلوک‌های ذخیره موجب افزایش تاخیر اصابت خواندن که در مسیر بحرانی است نمی‌شود.

اصابت نوشتن: در این حالت همچون حالت قبل مقایسه بین نشانه‌ی آدرس درخواستی و نشانه‌ی بلوک‌های داری بافر سه حالته فعال به طور موازی انجام شده و پس از اصابت، عملیات نوشتن انجام می‌شود. با استدلالی مشابه بخش قبل متوجه می‌شویم که اضافه کردن بلوک‌های ذخیره موجب افزایش تاخیر اصابت نوشتن نیز نمی‌شود. هرچند که بر خلاف اصابت خواندن، اصابت نوشتن در مسیر بحرانی قرار نداشته و افزایش مقداری تاخیر در آن، تاثیری در کارایی سیستم ندارد.

نقصان: در صورتی که پس ازجستجو در بلوک‌های یک مجموعه، بلوک مورد نظر یافت نشود، یک نقصان رخ داده است. در این حالت، الگوریتم جایگزینی، بایستی یک بلوک قربانی از بین بلوک‌های آن مجموعه انتخاب کند. در الگوریتم‌های جایگزینی قابل پیاده‌سازی همچون الگوریتم NRU (که در پردازنده‌ی UltraSPARC T2 [۱۳] استفاده شده است.)، بیت جایگزینی در تمام بلوک‌های یک مجموعه چک می‌شود و از بین بلوک‌هایی که اخیرا مورد دسترسی قرار نگرفته‌اند، یک بلوک به صورت تصادفی به عنوان بلوک قربانی انتخاب می‌شود. در ساختار معمول حافظه‌ی نهان که بلوک‌های مشخصی در هر مجموعه وجود دارد، کنترل‌کننده‌ی حافظه‌ی نهان، بیت جایگزینی این بلوک‌ها را برای انتخاب بلوک قربانی بررسی می‌کند.در راهکار پیشنهادی، با توجه به اینکه بلوک‌های مشخصی به ازاء هر مجموعه وجود ندارد و امکان حذف یک بلوک و اضافه شدن بلوک دیگر وجود دارد، بایستی تغییراتی در ساختار معمول حافظه‌ی نهان داده شود.

در ساختار پیشنهادی، یک بیت به نام بیت وضعیت و یک شمارنده به نام شمارنده‌ی خرابی به ازاء هر مجموعه و همچنین یک بیت خرابی به ازاء هر بلوک وجود دارد. بیت وضعیت که مشخص‌کننده‌ی وضعیت سلامت بلوک‌های اصلی است، در ابتدا یک بوده و به محض خرابی اولین بلوک اصلی از آن مجموعه، صفر می‌شود. با بررسی این بیت متوجه می‌شویم که حداقل یک بلوک در آن مجموعه خراب شده است. شمارنده‌ی خرابی در ابتدا صفر بوده و با خرابی هر بلوک از آن

مجموعه یکی می‌شمارد. بیت خرابی در ابتدا صفر بوده و با خرابی بلوک یک می‌شود. در زمان سالم بودن تمام بلوک‌های اصلی، بیت وضعیت یک است و کنترل‌کننده‌ی حافظه‌ی نهان، تنها بیت جایگزینی بلوک‌های اصلی را برای یافتن بلوک قربانی مورد بررسی قرار می‌دهد؛ با خرابی اولین بلوک، بیت وضعیت صفر می‌شود و می‌دانیم که از این پس حداقل یک بلوک ذخیره به مجموعه اضافه شده است؛ پس علاوه بر بلوک‌های اصلی سالم، بیت جایگزینی تعدادی از بلوک‌های ذخیره (به تعداد شمارنده‌ی خرابی) نیز مورد بررسی قرار می‌گیرد و بلوکی که اخیرا مورد دسترسی قرار نگرفته باشد، به عنوان بلوک قربانی انتخاب می‌شود.

برای پیدا کردن بلوک قربانی همواره بلوک‌هایی که دارای بیت خرابی صفر هستند (بلوک‌های سالم) بررسی می‌شوند. در نتیجه با خرابی یک بلوک اصلی و اضافه شدن یک بلوک ذخیره، تعداد بلوک‌هایی که در زمان پیداکردن بلوک قربانی بررسی می‌شوند افزایش نمی‌یابد.

۵- ارزیابی

برای ارزیابی راهکار پیشنهادی، از شبیه‌ساز سیستم کامل gem5 [۱۴] استفاده می‌کنیم. سیستم شبیه‌سازی شده، یک سیستم چهار هسته‌ای با معماری دستورالعمل ALPHA است که مجموعه‌ی وسیعی از برنامه‌های محک چند- نخی و چند-برنامگی بر روی آن اجرا می‌شود. برای به دست آوردن پارامترهای تاخیر و مساحت حافظه‌های نهان فرار و غیرفرار به ترتیب از ابزارهای تحلیلی CACTI [۱۵] و NVsim [۱۶] استفاده می‌کنیم. جزئیات پیکربندی سیستم پایه در جدول ۱ قابل مشاهده است. برنامه‌های محک استفاده شده همان‌طور که در جدول ۲ نشان داده شده است، شامل مجموعه‌ی کامل برنامه‌های چند- نخی PARSEC-2 [۱۷] و همچنین بارهای کاری چند- برنامگی تشکیل شده از برنامه‌های مجموعه‌ی SPEC CPU 2006 [۱۸] است. بارهای کاری چند- برنامگی، به صورت هوشمند انتخاب شده‌اند به طوری که رفتارهای متفاوتی از جهت فشار وارد شده به حافظه‌ی نهان را داشته باشند. هرچه ترافیک نوشتن، IntraV و InterV در حافظه‌ی نهان غیرفرار بیشتر باشد، طول‌عمر آن پایین‌تر خواهد بود. همان‌طور که جدول ۲ نشان می‌دهد، بارهای کاری Mix1 و Mix2 دارای ترافیک نوشتن بالا و اختلاف تعداد نوشتن پایین هستند؛ در نقطه‌ی مقابل، بارهای کاری Mix3 و Mix4 دارای ترافیک نوشتن پایین‌تر و اختلاف تعداد نوشتن بالا هستند و در نهایت بار کاری Mix5 دارای ترافیک نوشتن بالا و اختلاف تعداد نوشتن بالا است. در ادامه به ارزیابی راهکار پیشنهادی می‌پردازیم. ابتدا میزان افزایش طول‌عمر حافظه‌ی نهان سطح آخر غیرفرار را بررسی می‌کنیم. پس از آن، اثر راهکار پیشنهادی بر روی کارایی سیستم را مشاهده کرده و در نهایت به بررسی سربار ذخیره‌سازی و مساحت اشغالی آن می‌پردازیم.

۵-۱- ارزیابی طول‌عمر

در این بخش به ارزیابی بهبود طول‌عمر به دست آمده با اعمال راهکار پیشنهادی می‌پردازیم. معیاری که برای خرابی حافظه‌ی نهان در نظر گرفته شده است، خرابی تمام بلوک‌های یک مجموعه است. دو دلیل برای انتخاب این معیار وجود دارد: اول اینکه خرابی تعداد محدودی بلوک، موجب خرابی کل حافظه‌ی نهان نمی‌شود و دوم اینکه با خرابی یک مجموعه‌ی کامل تمام آدرس‌های نگاشت شده به آن مجموعه به طور مستقیم بین حافظه‌ی سطح پایین‌تر و حافظه‌ی اصلی جابه‌جا می‌شوند و این قضیه می‌تواند موجب افت شدید کارایی شود.

در جدول ۲، طول‌عمر حافظه‌ی نهان سطح آخر در سیستم پایه و در سیستم با اعمال راهکار پیشنهادی با اضافه کردن ۴ (S = 4)، ۸ (S = 8) و ۲۴ (S = 24) بلوک ذخیره به ازاء هر مجموعه و همچنین در سیستم با اعمال تکنیک i2WAP، نشان داده شده است. برخی برنامه‌ها همچون برنامه‌های چند- نخی blackscholes، canneal، fluidanimate، freqmine، raytrace، streamcluster و vips، وابسته به حافظه نبوده و به آن استرس وارد نمی‌کنند و حافظه‌ی نهان سطح آخر در آن‌ها، طول‌عمر بالایی دارد (حداقل ۱۸ سال) که برای سیستم‌های امروزی قابل قبول است. در مقابل، مابقی برنامه‌های چند-نخی و بارهای کاری چند-برنامگی انتخاب شده، به سیستم حافظه استرس وارد کرده و حافظه‌ی نهان سطح آخر، در برخی از آن‌ها طول‌عمری در حد چندین ماه را خواهد داشت. در ادامه تنها این برنامه‌ها را مورد بررسی قرار می‌دهیم.

جدول ۲- شخصات برنامه‌های محک مورد ارزیابی قرار گرفته

Workloads	Write Traffic [GBps]	InterV [%]	IntraV [%]	Cache Lifetime (Years)				
				Baseline	S = 4	S = 8	S = 24	i2WAP
PARSEC-2, 2009, Multi-Threaded								
blackscholes	0.05	173.9	111.19	48.59	72.89	97.62	195.28	50.75
bodytrack	1.11	222.34	48.84	2.42	3.63	4.59	9.11	3.07
caneal	1.76	16.08	9.89	18.49	27.73	36.61	74.05	20.77
dedup	1.72	24.24	26.52	8.36	12.55	20.48	33.29	10.83
facesim	0.75	35.6	12.27	8.78	10.17	12.40	25.69	10.67
ferret	5.24	47.54	22.47	1.07	1.60	2.22	8.04	1.85
fluidanimate	1.34	12.4	5.46	32.10	44.15	57.88	109.63	36.93
freqmine	0.15	99.05	103.49	58.73	88.10	111.24	215.57	61.92
raytrace	0.23	41.4	16.89	48.36	72.55	103.16	113.76	53.19
streamcluster	2.03	23.12	4.58	21.46	32.19	39.83	61.89	28.09
swaptions	2.72	56.93	107.27	2.47	3.71	4.15	11.79	3.02
vips	1.97	22.89	4.26	19.69	29.53	40.08	59.98	26.84
x264	0.51	73.48	47.09	9.61	14.41	28.18	76.34	10.18
SPEC CPU 2006, 4-Application Multi-Program								
Mix1: 2x libq, 2x leslie3D	8.9	14.2	3.11	4.31	6.46	7.84	16.60	4.65
Mix2: 2x leslie, 2x lbm	9.93	15.63	2.99	3.64	5.46	7.05	14.12	4.58
Mix3: 2x perlbench, 2x gcc	3.65	30.36	31.59	4.44	6.67	10.46	22.26	4.70
Mix4: 2x gromacs, 2x gobmk	2.82	41.49	25.56	4.34	5.51	6.23	17.11	4.76
Mix5: 2x deal, 2x gamess	8.16	25.32	64.72	0.75	1.12	2.31	5.64	1.06

در اکثر برنامه‌های مورد ارزیابی، طول‌عمر حافظه‌ی نهان سطح آخر در سیستم دارای ۴ بلوک ذخیره به ازاء هر مجموعه (S = 4)، نسبت به سیستم با اعمال تکنیک i2WAP بیشتر است. جدول ۳، میانگین طول‌عمر برنامه‌های وابسته به حافظه را نشان می‌دهد. طول‌عمر در سیستم پایه حدود ۴.۵ سال است که با اضافه کردن ۴ بلوک ذخیره به ازاء هر مجموعه، به حدود ۶.۵ سال رسیده است. این در حالی است که طول‌عمر در سیستم با اعمال تکنیک i2WAP، حدود یک سال کم‌تراست. همچنین با دو و چهار برابر کردن بلوک‌های هر مجموعه، طول‌عمر حدودا دو و چهار برابر می‌شود. در واقع، نسبت افزایش طول‌عمر با نسبت افزایش بلوک‌های مجموعه، برابر است.

جدول ۳- طول‌عمر برنامه‌های وابسته به حافظه

	Baseline	S = 4	S = 8	S = 24	i2WAP
Lifetime	4.56	6.48	9.63	21.82	5.40

۵-۲- ارزیابی کارایی

در این بخش به ارزیابی کارایی راهکار پیشنهادی می پردازیم. بدین منظور، از ابتدای شروع به کار سیستم تا زمان خرابی حافظه‌ی نهان سطح آخر را از لحاظ زمانی به هشت فاز تقسیم کرده و در هر فاز تعداد دستورالعمل در سیکل را محاسبه می‌کنیم. شکل ۴، میزان کاهش دستورالعمل در سیکل را در سیستم با اضافه کردن ۴ بلوک ذخیره به ازاء هر مجموعه و همچنین در سیستم با اعمال تکنیک i2WAP نشان می‌دهد. با توجه به اینکه این دو سیستم از لحاظ طول‌عمر بررسی شده در بخش قبل به یک‌دیگر نزدیک هستند، آن‌ها را برای مقایسه برگزیدیم. لازم به ذکر است که در این شکل مقادیر ارایه شده مربوط به میانگین برنامه‌های محک وابسته به حافظه است. این برنامه‌ها دارای ترافیک نوشتن و همچنین اختلاف تعداد نوشتن بالا هستند و در نتیجه بلوک‌های حافظه‌ی نهان تحت فشار بوده و زودتر خراب می‌شوند. این قضیه موجب می‌شود که در این برنامه‌ها برخلاف دیگر برنامه‌های محک ذکر شده در جدول ۲، روش ارایه شده در این مقاله (با توجه به خرابی زود هنگام برخی بلوک‌ها و نیاز به اضافه کردن بلوک ذخیره) و همچنین روش (i2WAP با توجه به ترافیک نوشتن و همچنین اختلاف تعداد نوشتن بالا که موجب اجرای تکنیک‌های این روش می‌شوند) با نرخ بالاتری به کار گرفته شوند و می‌توان تفاوت کارکرد با سیستم مبنا و تغییر احتمالی در کاهش کارایی را در آن‌ها مشاهده کرد.

جدول ۴- سربار مساحت راهکار پیشنهادی

	Baseline	S = 4	S = 8	S = 24
Area (mm^2)	3.21	4.14	5.72	10.02

در ابتدای کار سیستم دارای بلوک‌های ذخیره، هیچ سرباری از جهت کارایی ندارد؛ با خرابی بلوک‌های حافظه‌ی نهان به تدریج بلوک‌های ذخیره اضافه می‌شوند؛ پس از اضافه شدن تمام بلوک‌های ذخیره، خرابی‌های بعدی موجب کاهش بلوک‌های در دسترس مجموعه‌ها و در نتیجه کاهش کارایی می‌شود. در فاز هشتم، میزان کاهش دستورالعمل در سیکل برابر با ۱.۳۸ درصد است و بیشترین سربار کارایی را داریم. در نقطه‌ی مقابل، در سیستم با اعمال تکنیک i2WAP، در همان ابتدای کار سیستم،کاهش تعداد دستورالعمل در سیکل برابر با ۸.۴۲ درصد است که دلیل آن پس‌نویسی‌های مکرر بلوک‌های داغ به حافظه‌ی اصلی است. سربار کارایی، با خرابی بلوک‌های حافظه‌ی نهان بیشتر می‌شود. لازم به ذکر است که در فاز هشتم سیستم دارای ۸ و ۲۴ بلوک ذخیره به ازاء هر مجموعه نیز همچنان میزان کاهش تعداد دستورالعمل در سیکل کمتر از ۲ درصد است.

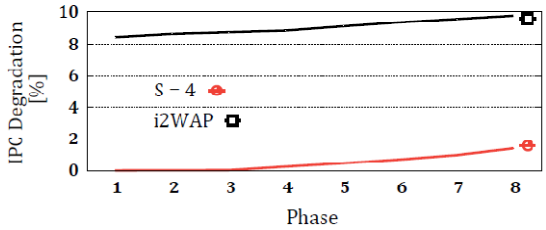
۵-۳- سربار ذخیره‌سازی و مساحت

در راهکار پیشنهادی، به ازاء هر مجموعه از حافظه‌ی نهان سطح آخر، یک بیت وضعیت و یک شمارنده‌ی خرابی مجموعه و همچنین به تعداد بلوک‌های اصلی و ذخیره، بیت خرابی وجود دارد. به عنوان مثال، در صورت اضافه کردن ۴ بلوک ذخیره به هر مجموعه از یک حافظه‌ی نهان ۸ راهه با حجم ۲ مگا بایت، سربار ذخیره‌سازی در مجموع ۷.۵ کیلو بایت است. با افزایش تعداد بلوک‌های ذخیره به ازاء هر مجموعه، تعداد بیت‌های خرابی بیش‌تر شده و همچنین سایز شمارنده‌ی خرابی افزایش می‌یابد؛ در نتیجه سربار ذخیره‌سازی افزایش می‌یابد. جدول ۴ میزان مساحت اشغالی حافظه‌ی نهان سطح آخر در سیستم پایه و همچنین در سیستم با اضافه کردن ۴، ۸ و ۲۴ بلوک ذخیره به ازاء هر مجموعه را نشان می‌دهد. با افزایش تعداد بلوک‌های ذخیره به ازاء هرمجموعه، مساحت اشغالی حافظه‌ی نهان بیش‌تر می‌شود به طوری که بیشترین مساحت اشغالی، مربوط به سیستم با اضافه کردن ۲۴ بلوک ذخیره به ازاء هر مجموعه است.

در سیستم با اعمال تکنیک i2WAP، با توجه به این‌که تنها از دو شمارنده استفاده می‌شود، سربار ذخیره‌سازی و مساحت تقریبا صفر است و مساحت این روش تقریبا برابر با سیستم پایه است. در نتیجه سربار مساحت راهکار پیشنهادی در مقایسه با تکنیک i2WAP قابل ملاحظه است اما در مقایسه با حافظه‌ی نهان SRAM با حجم یکسان، که مساحتی برابر با ۴۶.۶۱ میلی‌متر مربع دارد، ناچیز است؛ که البته لازم به ذکر است که این ناچیز بودن مساحت نسبت به حافظه‌ی نهان SRAM، به دلیل استفاده از حافظه‌ی STT-RAM حاصل می‌شود و از مزایای راهکار پیشنهادی این مقاله به حساب نمی‌آید.

۶- جمع‌بندی و کارهای آینده

روش‌های پیشین انجام شده در زمینه‌ی افزایش طول‌عمر حافظه‌های نهان غیرفرار، عموما دارای سربار کارایی و انرژی زیاد هستند. در این مقاله یک راهکار نوین جهت افزایش طول‌عمر این نوع حافظه‌ها ارائه شده است که بر مبنای اضافه کردن بلوک‌های ذخیره به ازاء هر مجموعه از حافظه‌ی نهان غیرفرار است؛ با خرابی یک بلوک از یک مجموعه، بلوک خراب به صورت هوشمند و بدون تاثیر بر روی کارایی سیستم، از آن مجموعه خارج شده و یک بلوک ذخیره به آن مجموعه اضافه می‌شود. سربار راهکار پیشنهادی مساحت آن است که این سربار با افزایش تعداد بلوک‌های ذخیره افزایش می‌یابد.



شکل ۴- سربار کارایی راهکار پیشنهادی در برنامه‌های وابسته به حافظه

در راهکار ارائه شده در این مقاله، به ازاء هر مجموعه، تعدادی بلوک ذخیره داریم و باخرابی یک بلوک از آن مجموعه، یکی از این بلوک‌های ذخیره به مجموعه

اضافه می‌شود. با توجه به این که ترافیک نوشتن وارد شده به مجموعه‌های حافظه‌ی نهان متفاوت است، برخی مجموعه‌ها بسیار سریع‌تر از بقیه دچار خرابی شده و بلوک‌های ذخیره‌ی خود را استفاده می‌کنند؛ این مجموعه‌ها، پس از خرابی بلوک‌های ذخیره، همچنان نیازمند بلوک‌های ذخیره‌ی اضافی هستند در حالی که دیگر مجموعه‌ها، ممکن است هیچ‌کدام از بلوک‌های ذخیره‌ی خود را استفاده نکرده باشند. در کارهای آینده می‌توان به جای در نظر گرفتن بلوک‌های ذخیره به ازاء هر مجموعه، به ازاء کل حافظه‌ی نهان، استخری از بلوک‌های ذخیره داشت که هر مجموعه در صورت خرابی از آن استفاده کند. در این حالت به صورت بهینه از بلوک‌های ذخیره استفاده می‌شود. تنها نکته‌ای که وجود دارد این است که مکانیزم اضافه شدن این بلوک‌های ذخیره بایستی به صورت هوشمند بوده و تاثیر منفی بر مسیر بحرانی و کارایی سیستم نداشته باشد.

مراجع

[1] O. J. Santana, A. Ramirez, and M. Valero, "Enlarging instruction streams," IEEE TC, vol. 56, no. 10, pp. 1342–1357, Oct. 2007.

[2] C. H. Kim, J.-J. Kim, S. Mukhopadhyay, and K. Roy, "A forward bodybiased- low-leakage SRAM cache: device and architecture considerations," in ISLPED, pp. 6–9, 2003.

[3] X. Dong, X. Wu, G. Sun, Y. Xie, H. Li, and Y. Chen, "Circuit and microarchitecture evaluation of 3d stacking magnetic RAM (MRAM) as a universal memory replacement," in DAC, pp. 554–559, 2008.

[4] M. Hosomi, H. Yamagishi, T. Yamamoto, K. Bessho, Y. Higo, K. Yamane, H. Yamada, M. Shoji, H. Hachino, C. Fukumoto, H. Nagao, and H. Kano, "A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram," in IEDM, pp. 459–462, 2005.

[5] J. Wang, X. Dong, Y. Xie, and N. P. Jouppi, "i2WAP: improving nonvolatile cache lifetime by reducing inter- and intra-set write variations," in HPCA, pp. 234–245, 2013.

[6] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STTRAM using early write termination," in ICCAD, pp. 264–268, 2009.

[7] J. Wang, X. Dong, and Y. Xie, "OAP: An obstruction-aware cache management policy for STT-RAM last-level caches," Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 847-852, 18-22 March 2013.

[8] K.-W. Kwon, S. H. Choday, Y. Kim, and K. Roy, "AWARE (asymmetric write architecture with redundant blocks): A high write speed STT-MRAM cache architecture," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 4, pp. 712–720, Apr. 2014.

[9] P. Zhou, B. Zhao, J. Yang, and Y. Zhang, "Energy reduction for STTRAM using early write termination," in ICCAD, pp. 264–268, 2009.

[10] A. Jadidi, M. Arjomand, and H. Sarbazi-Azad, "High-endurance and performance-efficient design of hybrid cache architectures through adaptive line replacement," in ISLPED, pp. 79–84, 2011.

[11] A. Bardine, M. Comparetti, P. Foglia, G. Gabrielli, and C. A. Prete, "Way adaptable D-NUCA caches," Int. J. High Perform. Syst. Archit., pp. 215-228, August 2010.

[12] M. Powell, Se-Hyun Yang, B. Falsafi, K. Roy and T. N. Vijaykumar, "Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories," ISLPED, pp. 90-95, 2000.

[13] Sun Microsystems, Inc., "UltraSPARC T2 supplement to the UltraSPARC architecture," Draft D1.4.3., 2007.

[14] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The Gem5 simulator," SIGARCH CAN, vol. 39, no. 2, pp. 1–7, May 2011.

[15] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0," in MICRO, pp. 3–14, 2007.

[16] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "NVSim: a circuit-level performance, energy, and area model for emerging nonvolatile memory," IEEE TCAD, vol. 31, no. 7, pp. 994–1007, 2012.

[17] C. Bienia, and K. Li, "PARSEC 2.0: A new benchmark suite for chipmultiprocessors," in MoBS, 2009.

[18] C. D. Spradling, "SPEC CPU2006 benchmark tools," SIGARCH CAN, vol. 35, no. 1, pp. 130–134, Mar. 2007.

محمدرضا جوکار در سال ۱۳۹۱ مدرک کارشناسی مهندسی سخت‌افزار خود را از دانشگاه شهید باهنر کرمان و در سال ۱۳۹۳ مدرک کارشناسی‌ارشد معماری کامپیوتر خود را از دانشگاه صنعتی شریف دریافت نمود. زمینه‌های تحقیقاتی مورد علاقه وی سلسله مراتب حافظه‌های غیرفرار، معماری سیستم‌های محاسباتی پردازش سریع و همچنین معماری سیستم‌های محاسباتی کم‌توان می‌باشد. وی هم‌اکنون دانشجوی دکتری دانشگاه شیکاگو در رشته‌ی علوم کامپیوتر می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

jokar@uchicago.edu

محمد ارجمند در سال ۱۳۸۵مدرک کارشناسی مهندسی سخت‌افزار خود را از دانشگاه شهید باهنر کرمان، در سال ۱۳۸۷ مدرک کارشناسی‌ارشد معماری کامپیوتر خود را از دانشگاه صنعتی شریف و در سال ۱۳۹۳مدرک دکتری معماری کامپیوتر خود را از دانشگاه صنعتی شریف دریافت نمود. زمینه‌های تحقیقاتی وی شامل جنبه‌های مختلفی از معماری کامپیوتر همچون روش شناسی طراحی ارتباطات روی تراشه برای چندپردازنده‌های سیستم بر روی تراشه، زمان‌بندی در حافظه‌ها و به کارگیری تکنولوژی‌های حافظه‌ی غیرفرار به‌عنوان جایگزینی برای DRAM می‌باشد. وی هم‌اکنون مشغول گذراندن دوره پسا دکتری خود در بخش مهندسی برق و

کامپیوتر دانشگاه ایالتی پنسیلوانیا می‌باشد و همچنین عضو دانشجویی ACM می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

arjomand@cse.psu.edu

حمید سربازی‌آزاد در سال ۱۳۷۱ مدرک کارشناسی مهندسی برق و کامپیوتر خود را از دانشگاه شهید بهشتی، در سال ۱۳۷۳ مدرک کارشناسی‌ارشد مهندسی کامپیوتر خود را از دانشگاه صنعتی شریف و در سال ۱۳۸۱ مدرک دکتری علوم رایانش خود را از دانشگاه گلاسکو انگلستان دریافت نمود. وی اکنون استاد مهندسی کامپیوتر در دانشکده‌ی مهندسی کامپیوتر دانشگاه صنعتی شریف و همچنین رئیس پژوهشکده علوم کامپیوتر پژوهشگاه دانش‌های بنیادی (IPM) می‌باشد. زمینه‌ی تحقیقاتی مورد علاقه‌ی وی شامل معماری پیشرفته کامپیوتر، سیستم و شبکه روی تراشه، سیستم حافظه و شبکه‌های اجتماعی می‌باشد که در این زمینه‌ها بیش از ۳۰۰ مقاله در مجلات و کنفرانس‌های معتبر مرتبط منتشر کرده است.

دکتر سربازی آزاد در سال ۱۳۸۵ جایزه‌ی بین‌المللی خوارزمی، و در سال ۱۳۸۶ جایزه‌ی دانشمند جوان TWAS در علوم مهندسی را دریافت کرد و به عنوان پژوهشگر برتر دانشگاه صنعتی شریف در سال‌های ۱۳۸۳، ۱۳۸۶، ۱۳۸۷، ۱۳۸۹ و ۱۳۹۲ معرفی شد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

azad@sharif.edu,azad@ipm.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۶/۰۶

تاریخ اصلاح: ۱۳۹۴/۰۷/۲۰

تاریخ قبول شدن: ۱۳۹۴/۱۰/۲۳

نویسنده مرتبط: محمدرضا جوکار، دانشکده علوم کامپیوتر، دانشگاه شیکاگو، ایلینوی، امریکا و دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران.

---

[1]Non-Volatile Memory (NVM)
[2]Spin-Transfer Torque RAM (STT-RAM)
[3]Write Endurance
[4]Inter-Set Variation (InterV)
[5]Intra-Set Variation (IntraV)
[6]Magnetic Random Access Memory
[7]Magnetic Tunnel Junction (MTJ)
[8]Tunnel Barrier
[9]Reference Layer
[10]Free Layer
[11]Bit Line
[12]Source Line
[13]Sense Amplifier

علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۳۲-۴۶
مقاله عادی

نگاشت و زمان‌بندی همزمان وظایف و ارتباطات انرژی‌آگاه بی‌درنگ در ساختارهای چندهسته‌ای

امین‌اله مه‌آبادی فاطمه عسگری بیدهندی

دانشکده فنی و مهندسی، دانشگاه شاهد، تهران، ایران

چکیده

در این مقاله یک متدلوژی نگاشت و زمان‌بندی بی‌درنگ انرژی‌آگاه برای برنامه‌ریزی همزمان وظایف و ارتباطات با هدف حل سریع با جواب نزدیک بهینه درتراشه‌های چند هسته‌ای ارایه می‌شود. متدلوژی پیشنهادی با برخورداری از ساختار نوین کروموزوم در الگوریتم ژنتیک و برخورداری از تابع جهش شبیه‌سازی گداخت، دارای قابلیت جلوگیری از تولید راه‌حل‌های غیرممکن جهت کاهش زمان تولید جواب نزدیک بهینه است. تحلیل ما از نتایج آزمایشات در فضای نانو تکنولوژی نشان می‌دهد که در نگاشت و زمان‌بندی همزمان نسبت به روش سنتی ژنتیک از سرعت همگرایی بسیار خوبی برخورداری است و به‌طور متوسط در ساختار زمان‌بندی حدود ۱۰٪ و در ساختار نگاشت بیش از ۹۰٪ بهبود سرعت در زمان اجرا، همراه با تولید جواب نزدیک بهینه را نشان می‌دهد.

کلمات کلیدی: شبکه بر تراشه، زمان‌بندی وظایف و ارتباطات، ژنتیک الگوریتم، تابع جهش، شبیه‌سازی گداخت، زمان‌بندی انرژی‌آگاه.

۱- مقدمه

تراشه‌ها آینده می‌توانند شامل صدها هسته پردازشی از پیش طراحی شده باشند که درون یک تراشه کنار هم قرار گرفته‌اند و یک تراشه با پیچیدگی بسیار بالا را ایجاد کنند [۱]. از مهم‌ترین مسائل چنین تراشه‌هایی، نگاشت و زمان‌بندی منابع درون تراشه است که با ترکیب ابعاد نگاشت و زمان‌بندی همزمان وظایف و ارتباطات از نوع مسایل سخت NP و نیاز تقلیل زمان حل آن است [۲]. در این بخش به مساله نگاشت و زمان‌بندی، چالش‌های مساله، راهبرد حل مساله، و نوآوری‌های ارایه شده می‌پردازیم.

۱-۱- نگاشت و زمان‌بندی

اکنون حل مساله نگاشت و زمان‌بندی همزمان وظایف و ارتباطات آنها با هدف کمینه‌سازی مصرف انرژی جهت وظایف بی‌درنگ در تکنولوژیِ زیرمیکرون مورد توجه قرار دارد و با نگاه به چالش‌های جدید مصرف انرژی [۳]، حل سریع و نزدیک بهینه آن برای تراشه‌های چندهسته‌ای از مساله‌های بسیار سخت محسوب می‌شود. بیشینه فرکانس عملیاتی یک پردازنده تک هسته‌ای می‌تواند از طریق توان نشتی و اثرات فرکانس رادیویی به آن آسیب بزند. این مشکل سازندگان را به محدود کردن بیشینه فرکانس پردازنده و به طراحی تراشه‌های چندهسته‌ای با فرکانس‌های کمتر سوق می‌دهد [۴] [۵] هر چند با افزایش تقاضای کارآیی کاربردهای نهفته پیچیده مدرن، افزایش فرکانس پردازنده تک‌هسته‌ای یا مشتری‌سازی آن پردازنده نمی‌تواند پاسخگو باشد لذا نیاز به پردازنده‌های با هسته‌های زیاد و با ارتباطات داده‌ای بسیار، یک ضرورت است [۶]. مفهوم اصلی آن تشریح نگاشت و زمان‌بندی وظیفه‌ها و ارتباطات گوناگون کاربردها است که به‌صورت کارآ بتواند بر روی چندین هسته به‌صورت همزمان به‌منظور افزایش کارآیی، اجرا شوند [۷] [۸].

افزایش تقاضا برای سیستم‌های با کارایی[1] بالا، سبب کوچک شدن زیاد تکنولوژی نیمه‌هادی[2] در سیستم‌های پیچیده الکترونیکی شده است.این پیشرفت

تکنولوژی طراحان را قادر به تجمیع پردازنده‌های همگن زیاد در یک تراشه می‌سازد که به سیستم چندپردازنده برتراشه[۳] (MPSoC) معروف هستند. گرچه برای هسته‌های بسیار در کنار توجه به پردازش وظیفه‌ها، برای افزایش قابلیت مقیاس‌پذیری و کارآیی ارتباطات آنها نیازمند زیرساخت مناسب شبکه برتراشه (NoC) هستیم [۹]. کوچکتر شدن عناصر سیستم نیز سبب افزایش توان[۴] و انرژی مصرفی آن می‌شود [۱۰]. مطابق تحقیقات مقاله [۷]، افزایش توان مصرفی نشتی[۵] در تکنولوژی‌های زیر میکرون مشکلات بسیاری دارد. از سوی دیگر میزان توان نشتی مصرفی تراشه در تکنولوژی زیرمیکرون[۶]، به ۶۰٪ کل توان مصرفی خواهد رسید و ضرورت توجه به ذخیره‌سازی مصرف انرژی در تراشه را نشان می‌دهد [۱۰].

فرآیند طراحی سیستم‌های نهفته[۷] معمولا دارای سه گام تقسیم‌بندی وظیفه‌های کاربرد، نگاشت وظیفه بر عنصر پردازش و و زمان‌بندی وظیفه‌ها و ارتباطات است [۱۱] [۱۲]. تقسیم‌بندی کاربرد[۸] براساس قیود[۹] و محدودیت‌های سیستم به وظیفه‌های سخت‌افزاری[۱۰] و نرم‌افزاری[۱۱]، نگاشت وظیفه‌های بر روی عناصر پردازشی موجود (نگاشت وظیفه‌های سخت‌افزاری بر روی مدارهای خاص منظوره (ASIC) و FPGAها، و وظیفه‌های نرم‌افزاری برروی پردازنده‌های همه‌منظوره، DSPها[۱۲] و شتاب‌دهنده‌ها[۱۳])، و زمان‌بندی همزمان وظیفه‌ها و ارتباطات برای دستیابی به کارایی بهینه جهت برآوردن قیود مختلف سیستم مانند زمان‌بندی بی‌درنگ و توان مصرفی انجام می‌شود [۱۳]. گرچه در فرآیند نگاشت بر بسترهای ناهمگن، به تعیین نوع هسته برای نگاشت وظیفه و هزینه نگاشت هر وظیفه روی هسته‌های متفاوت (یعنی هزینه پیاده‌سازی مانند کارآیی، توان مصرفی و اشتغال منابع) نیز نیاز است [۱۴].

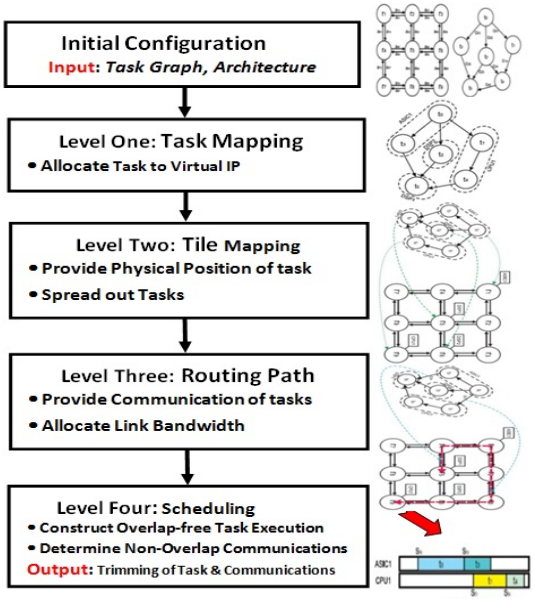
نگاشت وظیفه‌ها بر سیستم‌های چندهسته‌ای و بسیار هسته‌ای، شامل رعایت ترتیب اجرای وظیفه‌ها و ارتباطات آنها براساس بعضی معیارهای بهینه‌سازی مانند توان مصرفی و کارآیی محاسبات و ارتباطات است [۱]. یعنی ارتباطات وظیفه‌ها به‌منظور بهینه‌سازی تاخیرات ارتباطی و انرژی مصرفی یا روی یک هسته یا بر روی هسته‌ای نزدیک به‌یکدیگر نگاشت می‌شوند [۲]. بهینه‌سازی برای برآوردن قیود کارآیی اجرای کاربردها ضروری است. این ضرورت توسعه متدلوژی‌های کارآی نگاشت و زمان‌بندی را نشان می‌دهد که نیازمند مدل کاربرد، مدل بسترکاری، قیود (مانند کارآیی محاسبات و توان)، مدل کارآیی ارتباطات درونی (مانند زمان اجرا و توان مصرفی) و تخمین زمان بدترین اجرای وظیفه با پیاده‌سازی بر هسته‌های متفاوت (مانند پردازنده‌های خاص منظوره و همه‌منظوره) است [۱] [۱۱].

۱-۲- چالش‌های نگاشت و زمان‌بندی

چالش‌های حل مساله زمان‌بندی کاربردها عبارت از پیچیدگی جستجوی فضای راه‌حل، زمان طولانی پاسخ مساله، دقت جواب پاسخ، اهداف همزمان طراحی با قیود فرآوان، افزایش ابعاد وظیفه‌ها یا هسته‌های پردازشی، سرعت همگرایی و دقت روش‌های ارایه شده است. در روش‌های فرامکاشفه‌ای مانند الگوریتم ژنتیک [۱۵]، بهینه‌سازی گروهی ذرات [۱۶] و شبیه‌سازی گداخت [۱۷] فضای راه‌حل برای زمان‌بندی بهینه جستجو می‌شود. روش‌های مکاشفه‌ای، ترکیبی از تکنیک‌های جستجوی شبه تصادفی و بهینه‌سازی هستند که کشف فضای مساله را براساس تجارب انجام می‌دهند. این روش‌ها زمانی به‌کار می‌روند که جستجوی جامع و متدهای قطعی، بسیار سخت یا بکاربردن آن‌ها غیرممکن باشد و اگر زمان جستجو با افزایش ابعاد مساله به‌صورت نمایی رشد کند [۱۸].

روش‌های مکاشفه‌ای یک راه‌حل مطلوب با زمان نسبتا کوتاه فراهم می‌آورند. گرچه به‌دلیل نیاز به افزایش سرعت ممکن است پاسخ این روش‌ها مطلوب باشد ولی بهینه یا نزدیک به بهینه نباشد. برای جواب دقیق بهینه روش‌های برنامه‌ریزی خطی صحیح[۱۴]، برنامه‌ریزی خطی مشروط[۱۵]، برنامه‌ریزی غیرخطی[۱۶] و برنامه‌ریزی خطی صحیح مخلوط[۱۷] مثال‌های قابل ارایه با پیچیدگی چند جمله‌ای و غیر آن هستند [۹]. برنامه‌ریزی ریاضی، متدهای کمینه یا بیشینه‌سازی اهداف با ارضاء قیود مساله را برای جواب دقیق یا بهترین جواب دقیق فراهم می‌کنند.

پیچیدگی فضای جستجوی مساله با نیاز به جواب بهینه، افزایش وظیفه‌ها، همزمانی زمان‌بندی و نگاشت با پشتیبانی از عملیات همجوار پردازش و ارتباطات، و افزایش قیود آنها، ارایه روش‌های حل این مساله را بسیار سخت می‌کند [۱۹]. این روش‌ها می‌توانند خاص منظوره یا همه‌منظوره، سازنده یا براساس بهبود تکرار باشند. مکاشفه‌ای‌های سازنده راه‌حل جزیی می‌سازند تا به راه‌حل کامل برسند. مکاشفه‌ای‌های قابل تبدیل سعی بر تبدیل و بزرگ کردن فضای مساله تا رسیدن به کل جواب را دارند. زمان‌بندی فهرستی و الگوریتم ژنتیک مثال‌هایی از روش‌های مکاشفه‌ای همه‌منظوره از کلاس سازنده و قابل تبدیل هستند.



شکل ۱- متدلوژی پیشنهادی نگاشت و زمان‌بندی همزمان وظایف و ارتباطات

افزایش تعداد هسته‌ها، وظیفه‌ها، ارتباطات و همزمانی نگاشت و زمان‌بندی به سختی مساله کمک بسیار می‌کند که حل سریع آن با یافتن پاسخ بهینه یا نزدیک بهینه از محورهای تحقیقاتی نوین محسوب می‌شود [۱۰]. از آنجا که یافتن جواب بهینه و برآوردن تمامی قیود بسیار سخت و زمان‌بر است لذا طراحی راه‌حل مکاشفه‌ای با ارایه جواب مطلوب یک ضرورت همیشگی برای پیاده‌سازی سیستم‌های نهفته محسوب می‌شود [۲۰]. با مطالعه کارهای تحقیقاتی ارایه شده در حوزه نگاشت و زمان‌بندی وظیفه‌ها می‌توان بیان کرد که توجه بسیاری به متدلوژی‌های انرژی‌آگاه شده ولی در تسریع حل همزمان مساله همچنان مشکل کاهش زمان تولید محصول تا بازار نیز وجود دارد [۲۱] [۲۲] [۲۳] [۲۴].

۱-۳- راهبرد متدلوژی پیشنهادی

از آنجاکه یافتن تمامی نگاشت‌های و زمان‌بندی‌های ممکن با کاربردهای بزرگ پیچیده و در ابعاد بزرگ یک قالب کار در زمان محدود ممکن نیست لذا به داشتن راهبردهای تحلیل سریعتر با اهداف طراحی برای کشف نگاشت و زمانبندی کارا نیازمند هستیم. ما با استفاده از داده‌های اولیه، کروموزوم‌های مناسبی برای بهینه‌سازی تاخیر و انرژی مصرفی، در دو مرحله نگاشت و زمان‌بندی در معماری مش دو بعدی با در نظر گرفتن الگوریتم مسیریابی به‌عنوان راهبرد اولیه انتخابی

ارایه می‌کنیم. تمرکز ما بر تعریف کروموزوم با امکان جلوگیری از ایجاد راه‌حل‌های غیرممکن در الگوریتم ژنتیک پیشنهادی (بعد از مراحل جهش و تقاطع) به‌منظور کاهش تعداد جستجوها است. برای تضمین بهبود این کروموزوم بعد از مرحله جهش و به‌عنوان فاز جهش ژنتیک پیشنهادی از الگوریتم بهبود شبیه‌سازی گداخت استفاده شده است.

همچنین سعی داریم که از روش برنامه‌ریزی خطی عدد صحیح برای افزایش دقت جواب‌ها بهره ببریم. با این انتخاب‌ها، تعداد جستجو و در نتیجه زمان حل‌مساله کاهش و جواب بهینه حاصل می‌شود. معماری متدلوژی تکاملی ترکیبی پیشنهادی (مطابق شکل ۱) در سه مرحله "مقیدکردن وظایف به پردازشگرها"، "نگاشت وظایف و ارتباطات"، و "زمان‌بندی وظایف" با ساختار شبکه بر تراشه به حل مساله می‌پردازد. ورودی مساله گراف وظایف و معماری شبکه است. ابتدا وظایف با هدف بهینه‌سازی انرژی مصرفی مسیرها و هسته‌ها به هسته‌هایی که توانایی اجرای آن‌ها را دارند ملحق و نگاشت می‌گردند. سپس کار زمان‌بندی وظیفه‌ها و ارتباطات انجام می‌گیرد.

۱-۴- نوآوری

ما در این مقاله یک متدلوژی نگاشت و زمان‌بندی همزمان انرژی‌آگاه ایستا و شبه ایستا با در نظر گرفتن جواب مطلوب و نزدیک بهینه در سیستم‌های برپایه شبکه بر تراشه برای کاربردهای بی‌درنگ سخت ارایه می‌دهیم. براساس دانش ما، این مقاله اولین کاری نیست که با لحاظ کردن همزمان نگاشت و زمان‌بندی انرژی‌آگاه وظیفه‌ها را بیان می‌کند ولی سعی بر حل سریع و یافتن پاسخ نزدیک بهینه آن با کاهش فضای جستجو دارد. برای تقلیل زمان اجرای این قالب‌کار از الگوریتم ژنتیک، برای بهبود جواب‌های میانی از تکنیک شبیه‌سازی گداخت، و برای افزایش دقت جواب‌ها از برنامه‌ریزی صحیح بهره برده‌ایم که توانایی ارایه سریع جواب مطلوب، بهبود جواب‌ها در تکرار با هدف بهینه‌سازی انرژی مصرفی را برای حل همزمان پردازش و ارتباطات دارد. روش ارایه شده، مستقل از نوع الگوریتم زمان‌بندی انرژی‌آگاه است و هر الگوریتم جایگزین بهتر آن می‌تواند نتایج را بهبود بخشد. نتایج کار در تکنولوژی ۹۵ نانومتر ارزیابی شده و به‌طور خلاصه نوآوری‌های آن در این مقاله عبارتست از:

- ارایه یک قالب‌کار برای همزمانی تخصیص و زمان‌بندیِ وظایف و ارتباطات به‌صورت انرژی‌آگاه با افزایش سرعت همگرایی و دقت جواب و کاهش زمان تولید محصول،
- ارایه یک ساختار نوین کروموزوم الگوریتم ژنتیک با قابلیت جلوگیری از ایجاد راه‌حل‌های غیرممکن بعد از مراحل جهش و تقاطع،
- ارایه روش تضمین بهبود جواب هر نسل الگوریتم ژنتیک با فرار از تله بهینه‌های محلی جهت تولید بهینه سراسری،
- ارایه مدل دقیق حل مساله با ترکیبی از الگوریتم ژنتیک، شبیه‌سازی گداخت و برنامه‌ریزی خطی صحیح جهت ارایه سریع جواب مطلوب، بهبود جواب‌های مطلوب در تکرار جهت ارای جواب دقیق با هدف بهینه‌سازی انرژی مصرفی، و
- ایجاد فضای آزمون مناسب برای ارزیابی کارآیی الگوریتم در تکنولوژی ۹۵ نانومتر برای ساختارهای بر تراشه همگن[۱۸].

ما در ادامه مقاله و در بخش ۲ کارهای مرتبط با مساله را بررسی می‌کنیم. در بخش ۳ متدلوژی پیشنهادی را ارایه می‌دهیم. در بخش ۴، مدل نگاشت و زمان‌بندی پیشنهادی را برای نگاشت و زمان‌بندی همزمان انرژی‌آگاه تشریح می‌کنیم. نتایج شبیه‌سازی و آزمایشات را در بخش ۵ به‌طور مشروح ارایه می‌دهیم. نهایتاً در بخش ۶ نتیجه‌گیری از متدلوژی پیشنهادی را بیان می‌کنیم.

۲- کارهای مرتبط

نگاشت و زمان‌بندی وظایف روی چند هسته و پردازنده، مساله سخت[۱۹] است [۱]. لذا در ابعاد و اندازه بزرگ فقط می‌توان آن را با استفاده از روش‌های مکاشفه‌ای سازنده[۲۰] یا قابل تبدیل[۲۱] حل کرد. جواب مساله‌های با اندازه کوچک می‌تواند با روش‌های قطعی و با جواب بهینه یافت شود. روش‌های قطعی (مانند. روش شاخه و کران[۲۲]) به‌طور جامع راه‌حل‌های فضای مساله را کشف می‌کنند و بهترین جواب ارایه می‌دهند. برای حل مساله‌های بزرگ و یافتن سریع جواب نزدیک بهینه راه‌حل‌های مکاشفه‌ای[۲۳] زیادی مانند مکاشفه‌ای مبتنی بر لیست [۱۲] و فرامکاشفه‌ای [۱۱] ارایه شده‌اند.

جدول ۱- دسته‌بندی متدلوژی‌های زمان طراحی

مقاله	معماری	هدف بهینه‌سازی
ارسیلا و همکاران [۲۶]	همگن	زمان اجرا
راجیرو و همکاران [۳۱]	همگن	زمان اجرا
ساتیش و همکاران [۵۰]	همگن	زمان اجرا
بانی‌فتی و همکاران [۴۹]	همگن	زمان نگاشت و کیفیت
لین و همکاران [۸]	همگن	گذردهی، اشتغال منابع
وو و همکاران [۲۷]	همگن	انرژی مصرفی
راهی و همکاران [۳۴]	همگن	انرژی مصرفی
چن و همکاران [۱۸]	همگن	انرژی مصرفی
هو و همکاران [۲۲]	همگن	انرژی مصرفی، زمان اجرا
مارکون و همکاران [۲۳] [۲۴]	همگن	انرژی مصرفی، زمان اجرا
ایشیا و همکاران [۹]	همگن	انرژی مصرفی، زمان اجرا
می‌یر و همکاران [۲۵]	همگن	قابلیت اطمینان
تلی و همکاران [۵۱]	همگن	قابلیت اطمینان، دما
زانگ و همکاران [۳۸]	همگن	انرژی مصرفی
وو و همکاران [۵۲]	ناهمگن	زمان اجرا
مارکوسکی و همکاران [۵۳]	ناهمگن	زمان اجرا
چه و همکاران [۱۴]	ناهمگن	زمان اجرا
کاستریلون و همکاران [۱۳]	ناهمگن	زمان اجرا
مانولاچی و همکاران [۲۹]	ناهمگن	زمان کشف، دقت
جاوید و همکاران [۳۰]	ناهمگن	زمان کشف، دقت
وئو و همکاران [۵۲]	ناهمگن	انرژی مصرفی
هارتمن و همکاران [۲۰]	ناهمگن	قابلیت اطمینان
متدلوژی پیشنهادی	همگن	انرژی مصرفی، دقت، زمانِ اجرا

۲-۱- متدلوژی‌های نگاشت و زمان‌بندی

رده‌بندی‌هایی برای کلاس‌بندی متدلوژی‌های نگاشت مانند با اساس معماری هدف، با اساس معیارهای بهینه‌سازی، براساس بارکاری و غیره وجود دارد. متدلوژی‌های نگاشت در سطح زمان- طراحی و زمان اجرا براساس سناریوهای بارکاری ثابت و پویا، عمل بهینه‌سازی را انجام می‌دهند. براساس معماری هدف به سیستم‌های همگن و ناهمگن تقسیم می‌شوند.

زمان‌بندی زمان اجراء، نیازمند مدیری است که نگاشت زمان اجراء را انجام دهد و علاوه بر آن مسئولیت زمان‌بندی وظیفه [۲۵]، کنترل منابع، کنترل ساختار و مهاجرت وظیفه در زمان اجراء را بر عهده داشته باشد. این مدیر می‌تواند مدیریت متمرکز (استفاده از یک هسته به عنوان مدیر)، مدیریت توزیعی (تقسیم به نواحی کلاستری و استفاده از یک هسته در هر کلاستر به عنوان مدیر و ارتباط از طریق یک مدیر سراسری برای انتخاب بهترین کلاستر برای نگاشت) یا ترکیبی از هر دو داشته باشد.

متدلوژی‌های نگاشت زمان طراحی، مناسب سناریوهای بارکاری ثابت که برای مجموعه‌ای از کاربردهای از پیش مشخص با محاسبات و رفتار ارتباطی شناخته‌شده و دارای قالب‌کار ثابت بیان می‌شوند، قادر به پشتیبانی از پویایی زمان اجرا (مانند کاربردهای چند رسانه‌ای و شبکه‌ای) نیستند. لذا متدلوژی خاص خود را نیاز دارد که در راستای بررسی ما نیست.در متدلوژی‌های نگاشت زمان طراحی، نگاه سراسری به سیستم وجود دارد که تصمیم‌سازی بهتری برای استفاده از منابع سیستم صورت گیرد.

لذا در مقایسه با متدلوژی‌های نگاشت زمان اجرا که نگاه محلی و همسایگی دارند از کیفیت بهتر نگاشت و زمان‌بندی برخوردار است. بیشتر متدلوژی‌های نگاشت مربوط به زمان طراحی دارای قابلیت معماری همگن یا ناهمگن نیستند. در جدول ۱، کارهای اخیر نگاشت زمان طراحی، براساس معماری هدف و اهداف بهینه‌سازی ارایه شده است. بهینه‌سازی کارآیی محاسبات برای موفقیت ضرب‌الاجل‌ها یا کمینه‌سازی زمان اتمام کارها مهم است. کارآیی ممکن است به زمان اجراء، تاخیر، پریود، توان خروج و مانند آن که مرتبط با اطلاعات زمان‌بندی است اشاره کند.

روش‌های جستجوی متفاوتی به‌منظور یافتن نگاشت مطلوب بهینه یا نزدیک بهینه وظیفه‌ها مانند شبیه‌سازی گداخت [۸] [۲۶]، ژنتیک الگوریتم [۲۷]، جستجوی ممنوع [۲۹]، و برنامه‌ریزی خطی صحیح [۳۰] استفاده می‌شود. زمان اجرا و حافظه مصرفی در این کار بهینه‌سازی شده است. در کار دیگری مساله نگاشت وظیفه‌ها با توجه به بیشینه‌سازی گذردهی سیستم به بهبود ۲۰٪ دست یافت [۸]. در کار مشابهی با استفاده از GA برای اجرای کاربردهای جریان‌داده سنکرون روی یک سیستم چندهسته‌ای با توجه به تخصیص محدود حافظه به هر هسته صورت گرفت [۵۳]. در تحقیق دیگری با تمرکز بر فرموله کردن ILP مساله نگاشت حل شد. تمامی این متدلوژی‌ها با وجود دست‌یابی به پاسخ‌های مناسب، از هزینه بالا محاسباتی برای کاربردهای دارای وظیفه بسیار برخوردارند [۲۳].

هدف استراتژی‌های دیگر تمرکز بر فضای جستجو با هدف تقلیل هزینه محاسبات است. ترکیب برنامه‌ریزی خطی و برنامه‌ریزی قیود برای تسریع اجرا هدف است [۳۱]. معماری هدف این ساختار براساس ساختار باس بنا شده و مقیاس‌پذیر نبود. در کار دیگری از تکنیک تجزیه برای تسریع بهینه‌سازی قیود مساله استفاده شد [۵۰]. در تحقیقی دیگر کار بر روی بهینه‌سازی زمان ارتباطات و محاسبات صورت گرفت [۳۲]. متدهای بسیار دیگری که این راه را طی کردند گرچه دارای زمان کمتر جستجو بودند ولی از پاسخ‌های با کیفیت بالا برخوردا نبودند.

۲-۲- محدودیت‌های زمان طراحی

بیشتر متدلوژی‌های زمان طراحی، روش‌های جستجو مبنا (یعنی GA[۲۴]، ILP، SA[۲۵]) هستند که متحمل هزینه‌های محاسباتی بالا می‌شوند. گرچه آنها برای سیستم‌های کوچک پاسخ کارآ فراهم می‌کنند ولی در مقیاس بزرگ، ممکن است زمان ارزیابی آنها قابل‌پذیرش نباشد. این مساله در ترکیب ارتباطات و پردازش وظایف کار را پیچیده‌تر و این زمان را بیشتر می‌کند. این زمان می‌تواند بوسیله هرس فضای جستجو تقلیل یابد ولی ریسک این مساله، از دست دادن پاسخ‌های نگاشت با کیفیت بالا است. مقیاس‌پذیری حل این مساله با هدف جواب بهینه و ضرورت افزایش تعداد هسته‌ها در کنار نیازمندی به تکنولوژی شبکه بر تراشه یک ضرورت تحقیقاتی جدید است. مساله اصلی، کاهش فضای جستجو با هدف یافتن پاسخ‌های نگاشت و زمان‌بندی با همزمانی پردازش و ارتباطات و با کیفیت بالا یعنی سرعت پاسخ زیاد، جواب نزدیک بهینه، و سرعت همگرایی بالا است.

در تمامی سطوح فرآیند طراحی تکنولوژی زیرمیکرون، مساله همزمانی نگاشت و زمان‌بندی در طراحی انرژی‌آگاه ضروری است. کارهای قبلی یا فقط روی بهبود نگاشت تمرکز کرده‌اند و یا فقط به مساله زمان‌بندی توجه داشته‌اند. البته بیشتر تحقیقات یا تمرکز بر پردازش وظایف داشته‌اند و یا ارتباطات را مدل کرده‌اند و از همزمانی تمرکز بر مدل‌سازی پردازش وظایف و ارتباطات صرف‌نظر شده است. همچنین، ارایه بهبود دقت محاسبات برای ارایه جواب نزدیک بهینه در کارهای قبلی به اندازه کافی دقیق نیست و از همگرایی خوبی برخوردار نبوده است. این دلایل ما را بر آن داشت که در این مقاله روش جدید مکاشفه‌ای با مدل‌سازی همزمان ارتباطات در کنار محاسبات وظایف بی‌درنگ برای حل‌مساله همزمان نگاشت و زمان‌بندی انرژی‌آگاه تحت نگرش یافتن جواب سریع نزدیک به بهینه با ترکیب الگوریتم ژنتیک برای سرعت جواب، شبیه‌سازی گداخت برای بهبود تکرار و روش برنامه‌ریزی خطی صحیح برای افزایش دقت و یافتن جواب نزدیک بهینه ارایه دهیم.

۲-۳- متدلوژی‌های انرژی‌آگاه

مساله مهم در تحقیقات نهفته مدرن، بهینه‌سازی مصرف انرژی برای تداوم بیشتر باتری است و نیازمندی شدیدی به‌کار در زمان طراحی و تداوم زمان اجرا دارد. در کار مولی و همکاران، نتیجه ذخیره‌سازی ۵۴٪ توان مصرفی حاصلو متدلوژی با هدف برآوردن قیود کارآیی ارایه شد [۳۳]. نتیجه تحقیق راهی [۳۴] با اساس ILP بحث بهینه‌سازی نگاشت هسته‌ها در معماری توری به‌منظور کمینه کردن مصرف انرژی یا ازدحام NoC[۲۶] بود که ۸۱٪ ذخیره‌سازی انرژی حاصل شد. یک متدلوژی نگاشت چند مرحله‌ای برای بهینه‌سازی در [۱۸] ارایه شد. و همکاران یک روش مبتنی بر GA ارایه دادند که انرژی مصرفی را با روش ولتاژ مقیاسی پویا تا ۵۱٪ کاهش داد.

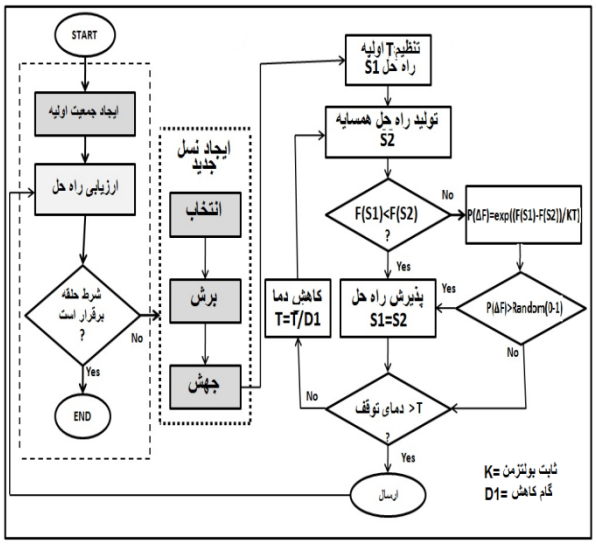
در بعضی از تحقیقات مساله بهینه‌سازی در دو بعد کارآیی محاسبات و انرژی مصرفی صورت گرفت [۹] [۲۲] [۲۴] در تحقیق هو و همکاران یک روش نگاشت برای تقلیل انرژی مصرفی از طریق کاهش انرژی ارتباطات در کنار تضمین کارآیی مورد نیاز با ۵۱٪ ذخیره انرژی ارایه شد [۲۲]. مارکون و همکاران با توسعه کار [۲۲] تکنیکی ارایه دادند که زمان‌بندی ارتباطات را علاوه بر میزان ارتباطات ارایه [۲۴] و علاوه بر تقلیل ۹۸٪ زمان اجرا به‌میزان قابل توجه ذخیره انرژی انجام دادند. اشیا و همکاران روشی برمبنای GA ارایه دادند که به پاسخ پرتو[۲۷] برای عوامل بهره‌وری و دقت دست یافتند در حالی که برای انرژی مصرفی و کارآیی نیز بهینه‌سازی انجام دادند [۹]. گرچه این کارها توانستند انرژی مصرفی را کاهش دهند ولی در زمینه دقت پاسخ نزدیک بهینه و سرعت تولید محصول تا بازار به نقطه خوبی دست نیافتند.

۳- متدلوژی پیشنهادی

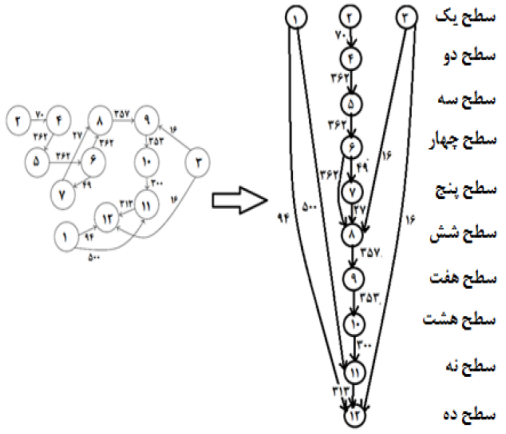
در این بخش به تعریف ارایه متدلوژی نگاشت و زمان‌بندی وظیفه‌ها می‌پردازیم. در ابتدا کروموزومی خوب برای مساله زمان‌بندی با امکان جلوگیری از ایجاد راه‌حل‌های غیرممکن بعد از مراحل جهش و تقاطع، ارایه می‌شود. سپس برای بهبود روش با الگوریتم ژنتیک در مرحله جهش، از الگوریتم شبیه‌سازی گداخت بهره‌برداری می‌شود تا بهبود کروموزوم بعد از مرحله جهش تضمین گردد. بعد مساله نگاشت با هدف کاهش انرژی با روش شبیه‌سازی خطی عدد صحیح فرموله می‌شود. نهایتاً برای کاهش جایگشت‌ها در حل قطعی مساله زمان‌بندی جهت جواب بهینه، راه‌حل پیشنهادی بیان می‌گردد.

۳-۱- کروموزم در زمان‌بندی

ما کروموزومی برای بخش زمان‌بندی طراحی کرده‌ایم که از ایجاد راه‌حل‌های غیر امکان‌پذیر پس از مراحل جهش و تقاطع جلوگیری می‌کند. برای این کار ابتدا گراف وظایف را به صورت زیر سطح بندی می‌کنیم.گره‌هایی که یال ورودی ندارند را در سطح اول قرار می‌دهیم. گره‌هایی که مبدا یال ورودی آن‌ها در سطح اول است در سطح دوم قرار می‌گیرند. به همین نحو سطح هر گره با توجه به بیشینه‌ی مبدا ورودی‌های آن به اضافه یک مشخص می‌شود. مثلا در شکل ۳، ۳ وظیفه در سطح اول و در مابقی سطح‌های یک وظیفه وجود دارد. سطح گره شماره ۸ که یک یال ورودی از گره ۷ (سطح ۵) و یک یال ورودی از گره ۶ (سطح ۴) دارد با توجه به سطح این دو گره، بیشینه سطح آن‌ها به اضافه یک (۶ = ۱ + ۵) خواهد بود. مسلما وظیفه موجود در هر سطح نمی‌تواند قبل از وظیفه والد خود که در سطح بالاتر از آن قرار دارد اجرا شود.

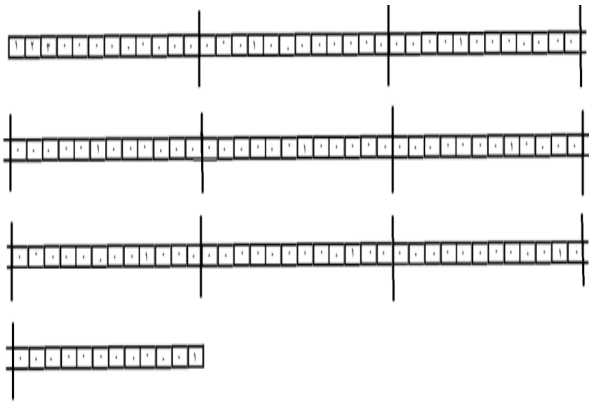


شکل ۲- فلوچارت الگوریتم ژنتیک بهبودیافته پیشنهادی



شکل ۳- سطح‌بندی گراف وظایف VOPD

برای پیاده‌سازی کروموزوم، گراف تغییر یافته را به برداری از اعداد صحیح با طول (تعداد سطح‌ها × تعداد وظایف) تبدیل می‌شود، به نحوی که بردار به تعداد سطح‌ها به زیر بردارهایی با طول تعداد وظایف تقسیم شده است. در هر سطح با توجه به اولویت وظایف، به وظایف موجود در آن، عدد طبیعی از ۱ تا تعداد وظایف آن سطح اختصاص داده می‌شود. با توجه به تعریف کروموزوم، تعداد کروموزوم‌های تولیدی برای زمان‌بندی گراف Vopd، !۳ یعنی ۶ عدد خواهد بود. بقیه کروموزوم‌های تولیدی، فقط در قسمت اول با این کروموزوم تفاوت دارند.



شکل ۴- کروموزوم ایجادی محک VOPD

در این شکل، ۱۲ عنصر اول برای سطح اول، ۱۲ عنصر بعدی برای سطح دوم و به همین ترتیب اختصاص داده شده است. شماره عناصر غیر صفر در هر سطح، شماره وظایف موجود در آن سطح را مشخص می‌کند. مثلا در سطح اول غیر صفر بودن عناصر ۱ و ۲ و ۳ حضور وظایف یک تا سه در سطح اول و عدد اختصاص داده شده به آن‌ها نشان‌دهنده اولویت اجرای وظایف آن سطح خواهد بود. مثلا در این سطح ترتیب ۱۲۳ خواهد بود. تعداد کروموزوم‌ها برای محک Consumer برابر !۲×!۴×!۳×!۲×۱ خواهد بود. یک نمونه از کروموزوم ایجادی برای این محک در شکل ۴ آورده شده است. عدد یک در خانه ۱۶ به این معنا است که در سطح دوم، وظیفه‌ی ۴ دارای اولویت اجرای یک است. به‌همین ترتیب وجود عدد ۳ در خانه ۳۱ به این معنا است که در سطح سوم وظیفه‌ی ۷ دارای اولویت اجرای سه است.

در اینجا نیز از رمزگذاری اعداد حقیقی مانند بالا برای بازنمایی در الگوریتم ژنتیک استفاده شده است. در ابتدا تعدادی کروموزوم به‌صورت تصادفی ایجاد می‌گردد. کروموزوم تولیدی برداری از اعداد صحیح با طول (تعداد سطح‌ها×تعداد وظیفه‌ها) خواهد بود، به نحوی که در ابتدا همه‌ی عناصر با صفر مقداردهی می‌شوند. سپس بردار به تعداد سطح‌ها به بردارهای با طول تعداد وظیفه‌ها تقسیم می‌شوند. هر کدام از این بردارها به‌سطح مربوطه اختصاص داده می‌شود و به‌صورت تصادفی به ژن‌های مربوط به وظیفه‌های موجود در آن سطح، عدد از ۱ تا تعداد وظیفه‌های آن سطح اختصاص می‌یابد. عملگر آمیزش به صورت تقاطع تک نقطه‌ای[۲۸] است و به این صورت پیاده‌سازی می‌شود که ابتدا یک نقطه از مضرب تعداد وظیفه‌ها برای تقاطع دو کروموزوم انتخاب و عمل تقاطع تک نقطه‌ای از آنجا انجام می‌شود. جهش نیز به این صورت پیاده‌سازی شده است که برای این عمل دو نقطه‌ی غیر صفر در یک سطح کروموزوم انتخاب و مقدار آن‌ها با یکدیگر جایگزین می‌گردد.

در آغاز هر نسل، ابتدا تابع هزینه برای هر یک از افراد جمعیت محاسبه و تابع تولید فرزند صدازده می‌شود تا از نسل فعلی به همان تعداد، یک نسل جدید تولید کند. نحوه عملکرد آن است که ابتدا افراد جمعیت جابجا می‌شوند، سپس درصدی از آن‌ها را دو به دو انتخاب کرده و عملیات آمیزش روی آن پیاده می‌شود. پس از آن درصدی از آن‌ها برای عملیات جهش انتخاب می‌گردد. در مرحله بعد میزان تابع برازش جمعیت فرزندان محاسبه می‌شود. سیاست انتخاب نسل جدید به این صورت است که ٪۲۰ والدها و ٪۸۰ فرزندان با استفاده از تابع چرخ رولت برای نسل جدید انتخاب می‌شوند. چرخ رولت براساس مقدار تابع برازش، احتمال انتخاب یک فرد را تعیین می‌کند. خروجی نیز بهترین راه‌حل از نظر تابع برازش در نسل آخر است. تابع برازش کروموزوم‌ها که باید کمینه شود، ترتیبی از وظیفه‌ها را

اخذ و وظیفه فعلی را نگاه می‌کند. اگر در ورودی‌های قبلی، وظیفه والد یا هم‌پردازنده‌ی این وظیفه بود، زمان شروع آن برابر با بیش‌ترین زمان خاتمه‌ی والدهای آن وظیفه به اضافه‌ی زمان انتقال داده از وظیفه والد به وظیفه فرزند و زمان خاتمه‌ی وظیفه‌های هم‌پردازنده آن خواهد بود. در غیر آن (یعنی فقدان وجود وظیفه والد و وظیفه هم‌سطح در میان وظیفه‌های قبلی) زمان شروع آن صفر خواهد بود.

۳-۲- بهبود الگوریتم ژنتیک

ما برای بهبود روش حل با الگوریتم ژنتیک در مرحله جهش، از الگوریتم شبیه‌سازی گداخت بهره گرفتیم تا بهبود کروموزوم بعد از مرحله جهش تضمین گردد (شکل ۲). در این بخش روش پیاده‌سازی تابع جهش عنوان شده است.

نگاشت با متد پیشنهادی: الگوریتم پیشنهادی در این قسمت در تابع جهش و عملگر انتخاب با الگوریتم ژنتیک معرفی شده برای نگاشت تفاوت دارد. در الگوریتم ژنتیک با توابع جهش و تقاطع موجود، امکان افزایش تابع هزینه کروموزوم انتخابی وجود دارد. در این الگوریتم تابع جهش برای تضمین عدم افزایش تابع هزینه با الگوریتم شبیه‌سازی گداخت پیاده‌سازی شده است.

جهش: همسایه‌های کروموزوم موردنظر را پیدا می‌کند که همسایه هر کروموزوم همان کروموزوم‌هایی هستندکه فقط در یک ژن از کروموزوم با هم متفاوت دارند. سپس با استفاده از شبیه‌سازی گداخت در یک حلقه بر روی همه همسایه‌های کروموزوم، آن‌ها را از نظر تابع برازش با وظیفه فعلی مقایسه می‌کند. اگر کروموزوم بهتری بود با آن جایگزین می‌گردد وگرنه با احتمالی جایگزین می‌شود که این احتمال رفته رفته کوچک و به مرور زمان احتمال انتخاب کروموزوم‌های بد کمتر می‌شود.

انتخاب: تعدادی از بهترین کروموزوم‌های نسل قبل، در نسل جدید کپی می‌شوند. روی درصدی از کروموزوم‌های دیگر عمل تقاطع اجرا و دو فرزند تولیدی جایگزین والدین می‌گردند. روی کروموزوم‌های باقیمانده عمل جهش اجرا و کروموزوم تولیدی جایگزین والد خود می‌شود. به این ترتیب جمعیت جدیدی با همان تعداد افراد ایجاد می‌گردد.

زمان‌بندی با متد پیشنهادی: الگوریتم پیشنهادیِ این بخش نیز در تابع جهش و عملگر انتخاب با الگوریتم ژنتیک معرفی شده برای زمان‌بندی تفاوت دارد. در الگوریتم ژنتیک با توابع جهش و تقاطع موجود، امکان افزایش تابع هزینه کروموزوم انتخابی وجود دارد. در این الگوریتم تابع جهش برای تضمین عدم افزایش تابع هزینه با الگوریتم شبیه‌سازی گداخت پیاده‌سازی شده است.

جهش: همسایه‌های کروموزوم مورد نظر را پیدا می‌کند که همسایه هر کروموزوم همان کروموزوم‌هایی هستند که فقط در یک قسمت از کروموزوم، بین ژن‌های $k \times (n+1)$ تا $n \times (k+1)$ کروموزوم، تنها مقدار موجود در دو ژن (اولویت اجرای دو وظیفه) با هم جابجا شده باشد که k یک عدد صحیح از صفر تا یکی کمتر از تعداد سطح‌ها و n تعداد وظایف است. سپس با استفاده از شبیه‌سازی گداخت در یک حلقه بر روی همسایه‌ها، آن‌ها را از نظر تابع برازش با وظیفه فعلی مقایسه می‌کند. اگر بهتر بود با آن جایگزین می‌گردد وگرنه با احتمالی جایگزین آن می‌شود که این احتمال به تدریج کوچک و به مرور زمان احتمال انتخاب کروموزوم‌هایِ بد کمتر می‌شود.

انتخاب: تعدادی از بهترین کروموزوم‌های نسل قبل، در نسل جدید کپی می‌شوند. روی درصدی از کروموزوم‌های دیگر عمل تقاطع اجرا شده و ۲ فرزند تولیدی جایگزین والدین خود می‌گردند. روی کروموزوم‌های باقیمانده عمل جهش اجرا و کروموزوم تولیدی جایگزین والد خود می‌شود. به این ترتیب جمعیت جدیدی با همان تعداد افراد ایجاد می‌گردد.

۳-۳- مدل‌سازی قطعی

مساله‌های در ابعاد و اندازه کوچک می‌تواند با روش قطعی به جواب بهینه دست یابد. روش‌های قطعی به‌طور جامع راه‌حل‌های فضای مساله را کشف می‌کنند و جواب بهتر را ارایه می‌دهند. روش شاخه و کران مثالی از روش‌های قطعی است. برنامه‌ریزی خطی یا بهینه‌سازی خطی، روشی ریاضیاست که به یافتن مقدار کمینه یا بیشینهِ یک تابع خطی روی یک چندضلعی محدب می‌پردازد. این چندضلعی محدب در حقیقت نمایش نموداری تعدادی قید از نوع نامعادله روی متغیرهای تابع است. به‌وسیله برنامه‌سازی خطی می‌توان بهترین نتیجه (مثلاً بیشترین سود یا کمترین هزینه) را در شرایط خاص و با قیود خاص به‌دست آورد [۳۵].

نگاشت با برنامه‌ریزی خطی عدد صحیح: مساله نگاشت وظایف ما بر روی معماری مش دو بعدی با روش برنامه‌ریزی خطی عدد صحیح حل شده است. معادلات مدل‌سازی خطی عدد صحیح و تابع هزینه استفاده شده در این مساله به صورت زیر خواهد بود. هدف در این روش بهینه‌سازی انرژی مصرفی برای اجرای وظایف بر روی پردازنده‌ها است. در معادلات این بخش از متغیرهای زیر استفاده شده است:

m : تعداد وظایف و n تعداد پردازنده‌ها است.
$\mathcal{M}$: عددی خیلی بزرگ در نقش بی‌نهایت است.
$\mathcal{E}_{ij}$: انرژی مصرفی توسط $task_i$ که بر روی پردازنده j اجرا می‌شود.
$\mathcal{f}_{ii'}$: حجم داده مبادله‌ای مابین $task_i$ و $task_{i'}$ است.
a_{ij} : اگر $task_i$ بر روی پردازنده‌ی j قابل اجرا باشد مقدار a_{ij} برابر یک و در غیر این صورت برابر صفر است.
$d_{jj'}$: فاصله مابین پردازنده j و j' که طبق رابطه (۱) به روش منهتن محاسبه می‌شود.

$$d_{jj'} = abs(x_{j'} - x_j) + abs(y_{j'} - y_j) \qquad (۱)$$

هدف این مدل کمینه‌سازی انرژی مصرفی است. انرژی مصرفی شامل دو بخش یعنی انرژی مصرفی اجرای $task_i$ در PE_j و انرژی مصرفی تبادل داده‌ها مابین عناصر پردازشی مختلف (در رابطه (۲)) است. بخش اول آن مجموع انرژی مصرفی هر وظیفه بر روی پردازنده انتخابی برای اجرای آن و بخش دوم انرژی مصرفی مربوط به ارتباطات است. ξ انرژی مصرفی جهت انتقال یک واحد داده در واحد مسافت را نشان می‌دهد.

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathcal{E}_{ij} \cdot \mathcal{X}_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{i'=1}^{n}\sum_{j'=1}^{m} \xi \cdot \mathcal{f}_{ii'} \cdot d_{jj'} \cdot r_{iji'j'} \qquad (۲)$$

با توجه به روابط (۳) تا (۵)، $\mathcal{X}_{ij}$ وقتی برابر یک است که پردازنده‌ی j برای $task_i$ انتخاب شده باشد.

$$\sum_{j=1}^{n} \mathcal{X}_{ij} = 1 \qquad (۳)$$

$$\mathcal{X}_{ij} \le a_{ij} \qquad \forall\, i \in \{1, \dots, m\}; \forall\, j \in \{1, \dots, n\} \qquad (۴)$$

$$\mathcal{X}_{ij} \in \{0,1\} \qquad \forall\, i \in \{1, \dots, m\}; \forall\, j \in \{1, \dots, n\} \qquad (۵)$$

$r_{iji'j'}$ یک متغیر باینری است و با توجه به وجود یال از $task_i$ به $task_{i'}$ به آن مقداردهی می‌شود. اگر $task_i$ در پردازنده j و i' در پردازنده j' اجرا شوند لذا $r_{iji'j'}$ برابر با یک است. در غیر این صورت اگر حداقل یکی از شرایط بالا صدق نکند این مقدار برابر با صفر خواهد بود.

$$r_{iji'j'} \le \mathcal{X}_{ij} \qquad (۶)$$

$$r_{iji'j'} \le x_{i'j'} \quad (۷)$$

$$r_{iji'j'} \ge x_{ij} + x_{i'j'} - 1 \quad (۸)$$

$$r_{iji'j'} \in \{0,1\} \qquad \forall\, i, i' \in \{1, \dots, m\};\ \forall\, j, j' \in \{1, \dots, n\} \quad (۹)$$

زمان‌بندی با مدل‌سازی تقلیل جایگشت‌های روش قطعی: در نگاشت قطعی، مجموعه جواب‌ها به صورت آرایه‌هایی با طول تعداد وظایف می‌باشند که خانه‌های آن‌ها با شماره پردازنده‌ی مورد استفاده برای آن وظیفه تخصیص داده شده‌اند. همه جایگشت‌های ممکن برای مساله برابر با حاصل‌ضرب تعداد پردازنده‌های مورد استفاده برای کلیه وظایف است. مثلا برای محک Jpeg که از ۸ وظیفه موجود در گراف وظایف، ۲ وظیفه بر روی ۴ پردازنده مش دوبعدی[۲۹] ۲×۲ و مابقی بر روی ۳ پردازنده قابل اجرا هستند، تعداد کل جایگشت‌هایی که باید بررسی شوند برابر با ۴×۳×۳×۳×۳×۳×۳×۳=۱۱۶۶۴ است که برای همه آن‌ها تابع هزینه محاسبه شده و جایگشتی که کم‌ترین مقدار انرژی را برای نگاشت به‌دست می‌دهد به‌عنوان جواب مساله انتخاب می‌گردد.

در حل مساله زمان‌بندی به روش قطعی مجموعه جواب‌ها به‌صورت آرایه‌هایی با طول تعداد وظایف هستند که خانه‌های آن‌ها با اولویت اجرای آن وظیفه، از یک تا تعداد وظایف تخصیص داده می‌شود. همه جایگشت‌های ممکن برای مساله برابر با فاکتوریل تعداد وظایف است. مثلا برای محک Jpeg که ۸ وظیفه در گراف وظایف آن موجود باشد، تعداد کل جایگشت‌هایی که باید بررسی شوند برابر با ۸!=۸×۷×۶×۵×۴×۳×۲×۱=۴۰۳۲۰ است که برای همه آن‌ها تابع هزینه محاسبه شده و جایگشتی که کم‌ترین مقدار زمان اجرا را در زمان‌بندی (کمترین زمان اجرای آخرین وظیفه) به دست می‌دهد به عنوان جواب مساله انتخاب می‌گردد.

در اینجا برای پیاده‌سازی زمان‌بندی قطعی جهت حذف یکسری جایگشت‌های غیرممکن مساله و با هدف کاهش جستجو، گراف مانند شکل ۳ سطح‌بندی می‌شود. از آنجا که هر وظیفه نمی‌تواند پیش از وظیفه‌ی والد خود اجرا شود، از ابتدا راه‌حل‌هایی که عدد اختصاصی به وظیفه فرزند را در زمان‌بندی کوچکتر از عدد اختصاص یافته به وظیفه والد قرار می‌دهند، حذف می‌شود. مجموعه جواب‌ها به‌صورت آرایه‌هایی با طول تعداد (سطح‌ها × تعداد وظیفه‌ها) است که به تعداد سطح‌ها به بردارهای با طول تعداد وظیفه‌ها تقسیم شده است. هر کدام از این بردارها به‌سطح مربوطه اختصاص یافته و به خانه‌های مربوط به وظیفه‌های موجود در آن سطح، عدد صحیح از ۱ تا تعداد وظیفه‌های آن سطح (اولویت اجرای وظایف موجود در سطح) اختصاص داده می‌شود.

حال برای آن که یک سری جایگشت‌های امکان‌پذیر که با این کار حذف شده‌اند را نیز در محاسبات بیاوریم، گراف را با همه سطح‌بندی‌های ممکن دیگر با در نظر گرفتن مسیر بحرانی و عدم افزایش طول آن مسیر نیز حل می‌کنیم. با حل مساله تعداد جایگشت‌ها برای محک Jpeg از ۴۰۳۲۰ به ۶=۱!×۱!×۱!×۱!×۳!×۱!×۱!×۱ خواهد رسید که این عدد با توجه به تعداد وظایف موجود در سطوح مختلف گراف سطح‌بندی شده است. مثلا در این محک تعداد وظایف موجود در سطوح اول تا پنجم آن به ترتیب برابر ۱، ۱، ۱، ۳، ۱، ۱ است که در این صورت حل همه حالت‌ها مدت زمان بسیار کمی نیاز دارد و به راحتی قابل اجرا است. این محک فقط یک سطح‌بندی خواهد داشت. همین طور محک Mpeg4 نیز دارای یک سطح‌بندی است و تعداد جایگشت‌های آن از !۱۲ به !۹×۳! کاهش می‌یابد. محک Office automation نیز دارای ۲ سطح‌بندی و بهبود تعداد جایگشت‌ها از !۵ به ۲(!۲) است. این مقادیر برای محک‌های MWD و Networking و Consumer و VOPD به ترتیب برابر با !۱۲، !۱۳، !۱۲، !۱۲و مقادیر جدید برابر ۴(!۲)+۱۰(!۲×!۲×!۲×!۲×!۲×!۱×!۳)، ۴(!۳×!۳×!۳×!۴)، ۲(!۲×!۳×!۲×!۳×!۲)+۸(!۱×!۲×!۳×!۴×!۲)+۳(!۲×!۲×!۲×!۴×!۲) و ۴۰(!۳) است که این تعداد نسبت به تعداد قبلی بسیار کمتر شده‌اند.

۴- نگاشت و زمان‌بندی وظایف و ارتباطات

در این بخش با تعریف مساله و ارایه مدل‌های جریان و فرکانس، معماری سیستم و کاربرد، به بیان مساله می‌پردازیم. برای هر قالب خاص با اطلاعات تعیین شده، گراف جریان وظیفه‌های کاربرد، و پارامترهای مدل، یک مجموعه تخصیص و زمان‌بندی تعیین می‌شود. این روش به‌طور همزمان وظایف و ارتباطات را مدل و توان مصرفی تراشه را کمینه می‌کند. فرض می‌کنیم که نتایج زمان‌بندی حاصل ما یک زمان‌بندی بلاانقطاعی[۳۰] است. در شکل ۵ ساختار مساله تخصیص و زمان‌بندی انرژی- آگاه وظیفه‌ها برای سیستم بر تراشه ارایه شده است. این مثال دارای مدل کاربرد در قالب گراف وظایف، مدل معماری، و زمان کامل شدن چند وظیفه است.

۴-۱- مدل جریان و فرکانس

تغییرات پارامترهای کلیدی ترانزیستور روی تغییرات فرکانس و توان نشتی تراشه اثر دارد. برای دما و V_{dd} ثابت، فرکانس و توان نشتی زیرآستانه هر هسته بر تراشه می‌تواند بوسیله V_{th} و L_{eff} تقریب زده شود.

مدل فرکانس: در این کار، ما از مدل فرکانس هسته ارائه شده در [۴۸] استفاده می‌کنیم. تاخیر دروازه می‌تواند با توجه به V_{th}، بوسیله رابطه (۱۰) مدل شود.

$$T_g \propto \frac{1 + V_{th}/V_{th0}}{(V - V_{th})^{\alpha}} \quad (۱۰)$$

جایی که $\alpha = 1.3$ است. مقدار V_{th} دارای توزیع نرمال است. برای مقدار قابل قبول V_{th}، رابطه (۱۰) در بازه مورد نظر تقریبا خطی است لذا T_g بوسیله توزیع نرمال تقریب‌زده می‌شود. فرکانس کار هر هسته بستگی به تاخیر مسیرهای بحرانی[۳۱] T_{cp} دارد. در پردازنده‌های جدید هزاران مسیر بحرانی وجود دارد [۳۷]. احتمال تاخیر طولانی‌ترین مسیر بحرانی تراشه ($\max T_{cp}$) در [۳۷] محاسبه شده است. لذا عکس تاخیر طولانی‌ترین مسیر می‌تواند به‌عنوان فرکانس هسته و به‌صورت رابطه (۱۱) بیان شود.

$$f = \frac{1}{delay} = \frac{1}{\max\ T_{cp}} \quad (۱۱)$$

مدل جریان نشتی: ما برای مدل کردن توان نشتی از مدل ارایه شده مقاله [۴۷] استفاده می‌کنیم. تفاوت این مدل با مدل مقاله [۳۶] آن است که از جریان نشتی تولیدی توسط اثر DIBL صرف نظر شده است. از این جریان صرف نظر شده زیرا در تکنولوژی‌های آتی، منبع اصلی جریان نشتی همان جریان نشتی زیر آستانه است. براساس این فرضیات رابطه (۱۲) را برای جریان نشتی داریم.

$$I_{leakage} \propto v_t^2 e^{\frac{(V_{off} - V_{th})}{\eta v_t}} \quad (۱۲)$$

جایی که $v_t = k\frac{T}{q}$ همان ولتاژ گرمایی (k ثابت بولتزمن، T دمای تراشه، و q شارژ الکترونی است) و $V_{th} = Tc_1 + c_2$ ولتاژ آستانه (c_1 فاکتور دمایی ولتاژ آستانه است)، η ضریب سوئینگ آستانه، و V_{off} یک پارامتر تجربی است. پارامترهای

c_1, c_2, η, V_{off} باید بوسیله برازش منحنی رابطه (۱۲) برای داده‌های استخراجی در شبیه‌ساز SPICE تعیین شود. به این منظور پارامترهای فوق با استفاده از مدل‌های کتابخانه‌ای قابل پیش‌بینی BSIM4 مربوط به روش دروازه فلزی hi-K تحت تکنولوژی ۹۵ نانومتر تعیین شده‌اند. برای تقریب اثر تغییر V_{th} روی کل توان نشتی، توزیع V_{th} برای تمامی ترانزیستورهای تراشه در رابطه (۱۲) جایگزین می‌شود لذا تابع توزیع توان مصرفی نشتی لاگ نرمال[۳۲] است.

مدل توان: براساس توزیع V_{th}، و آستانه همبسته وابسته به تکنولوژی ϕ، توزیع مورد نیاز فرکانس و توان مصرفی نشتی را استخراج می‌شود. حداقل میزان فرکانس، به‌عنوان فرکانس سیستم تعیین می‌شود. برای محاسبه توان‌نشتی از نرخ توان‌نشتی تحت تغییرات جریان نشتی بدون تغییر استفاده می‌کنیم. توان نشتی و ولتاژ آستانه V_{th} هر رخداد، از رابطه (۱۳) بدست می‌آید.

$$P_{leak}^{event} / P_{leak0}^{event} = e^{\frac{(V_{th0} - V_{th}^{event})}{\eta v_t}} \qquad (۱۳)$$

جایی که P_{leak}^{event} میزان توان نشتی برای ولتاژ آستانه V_{th}، و P_{leak}^{event} توان نشتی برای مقدار اسمی V_{th} است. جریان نشتی هر رخداد معادل میانگین مقدار توان نشتی نقطه شروع و پایان است. مثلا اگر توان نشتی نقطه شروع و پایان حالت انتخابی برابر a و b باشد لذا جریان نشتی این حالت معادل $c = \frac{(a+b)}{2}$ است.

۴-۲- مدل معماری و کاربرد

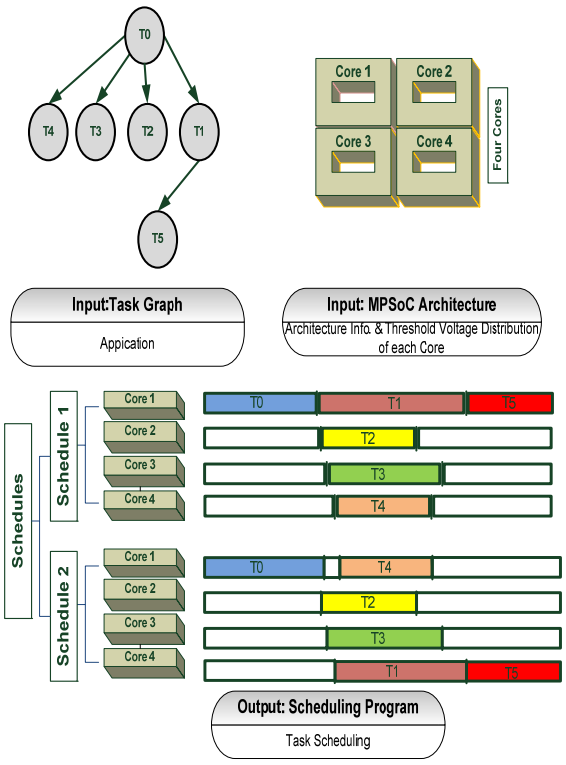
به‌طورکلی سیستم‌های نهفته دارای هسته‌های ناهمگن هستند. این سیستم‌ها از مجموعه‌ای از هسته‌ها $C=\{c_i : 1 \le i \le m\}$ تشکیل شده‌اند. توان پویا و نشتی هر هسته $core(c_i)$ وقتی که وظیفه t_j را اجرا می‌کند بوسیله P_{ij}^{dyn} و P_{ij}^{leak} ارائه می‌شود. شکل ۶ ارائه‌کننده یک مدل معماری نمونه با هم‌بندی توری[۳۳] است.

مدل کاربرد بوسیله یک گراف مستقیم غیرمدور DAG[۳۴] به‌صورت G(N,E) مدل می‌شود جایی که N مجموعه‌ای از وظیفه‌هایی است که متعلق به آن کاربرد است. همچنین E مبین مجموعه کمان‌هایی است که ارایه‌کننده وابستگی‌های داده و کنترل بین وظیفه‌های آن کاربرد هستند (شکل ۷). نماد وظیفه‌ها در گراف به صورت دایره ارایه می‌شود و هر وظیفه دارای سه نوع منبع[۳۵] (بدون کمان ورودی)، چاهک[۳۶] (بدون کمان خروجی)، و معمولی[۳۷] (دارای ورودی و خروجی) است. در این مدل، زمان رهاسازی و ضرب‌الاجل هر وظیفه t_i به‌وسیله r_i و d_i ارایه می‌شود. اگر وظیفه‌ای در محک دارای ضرب‌الاجل نباشد مقدار آن بی‌نهایت فرض می‌شود. برای هر وظیفه t_i، زمان اجرای بدترین حالت زمانی که روی هسته c_i اجرا می‌شود بوسیله et_{ij} ارایه می‌گردد. شکل ۷ یک نمونه گراف DAG و ضرب‌الاجل‌های مربوطه آن را نشان می‌دهد.

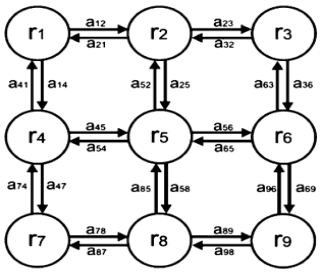
۴-۳- نگاشت و زمان‌بندی

ما در این بخش به شرح مساله تخصیص و زمان‌بندی می‌پردازیم. روال الگوریتم پیشنهادی در الگوریتم ۱ بیان شده است. این الگوریتم به پنج مرحله تقسیم شده است. در مرحله اول (سطح‌بندی گراف) که بر اساس یال‌های ورودی گراف مرجع ایجاد می‌شود. در مرحله دوم (تولید کروموزوم‌ها) که در ابتدا به‌صورت تصادفی ایجاد می‌شود و سپس بر اساس ورودی‌ها، بهبود می‌یابند. سپس در مرحله سوم (گزینش وظایف) با یک روش الویتی، n وظیفه از یک سطح وظایف برگزیده می‌شود. این روش از جهش روی کروموزوم‌های باقی‌مانده به عنوان جایگزین والد آن کروموزوم بهره می‌برد. در مرحله چهارم (زمان‌بندی)، الگوریتم نگاشت و زمان‌بندی برای کروموزوم‌های انتخابی، اجرا می‌شود. تمامی تخصیص‌ها و زمان‌بندی‌های تولیدی این مرحله برای تمامی کروموزوم‌ها اجرا می‌شود. کل اطلاعات لازم و موردنیاز تخصیص و زمان‌بندی مانند برآوردن قیود کارایی و مقدار انرژی کمینه در زمان اجرای کاربرد در مرحله پنجم (محاسبات)، محاسبه می‌شود.

ما براساس قیود نواحی انرژیِ مصرفی برای محاسبات و مبادله داده‌های ارتباطی، k زمان‌بندی را از میان n زمان‌بندی نامزد انتخاب می‌کنیم. این عمل در مرحله ششم (گزینش زمان‌بندی) انجام می‌شود. پس از فرآیند تقلیل، نگاشت فرکانس واقعی تراشه بوسیله تکنیک‌های الحاق سرعت بالغ‌شده[۳۸] فراهم می‌شود [۶]. زمان‌بند[۳۹] براساس این اطلاعات، از میان زمان‌بندی‌های انتخابی، یک زمان‌بندی مناسب را بر می‌گزینند.

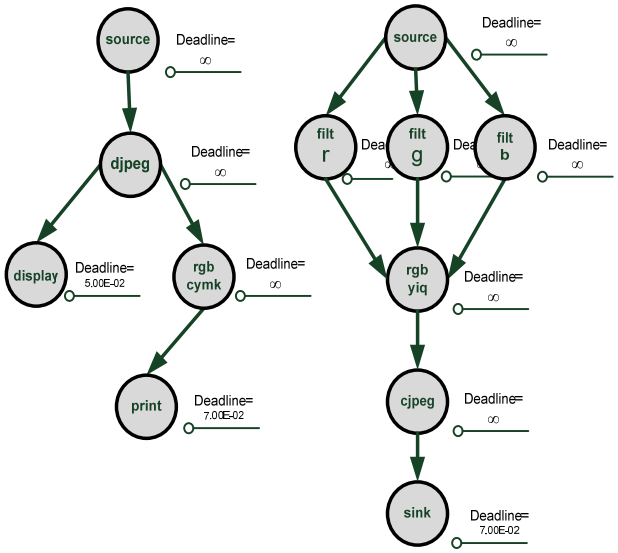


شکل ۵- تعریف مساله تخصیص و زمان‌بندی همزمان وظایف و ارتباطات



شکل ۶- مدل معماری توری شبکه پردازنده و ارتباطات نمونه ۳×۳

در این مرحله، الگوریتم تعدادی زمان‌بندی را به‌عنوان زمان‌بندی نامزد تولید می‌کند. ما برای تولید این نامزدها، یک الگوریتم تخصیص و زمان‌بندی انرژی‌آگاه را مورد استفاده قرار می‌دهیم. برای پشتیبانی از سیستم‌های دوره‌ای براساس روش تحقیقی مقاله [۴۰]، اصلاحات و تغییرات لازم را ایجاد و برای ابردوره[۴۰] وظیفه‌ها، مساله تخصیص و زمان‌بندی را حل کرده‌ایم (خط ۱۳ در الگوریتم ۱). این مساله بطور کامل در الگوریتم ۱ نشان داده شده است.



شکل ۷- نمونه گراف DAG از محک Consumer در مجموعه E3S [۳۹]

الگوریتم ما یک الگوریتم مکاشفه‌ای است که برای عملیات تخصیص و زمان‌بندی از ایده بهبود کروکوزوم استفاده می‌کند. در ایده، تقلیل زمان و تعداد جستجوی برای تولید کروموزوم و جمعیت تولیدی مساله ژنتیک یک معیار اندازه‌گیری مهم است. سطح وظیفه i برابر با اختلاف بین سطوح آن وظیفه است که تعریف می‌شود. این معیار به عنوان یک شاخص الویت وظیفه i در طول مدت زمان عملیات تخصیص و زمان‌بندی بکار می‌رود. هر چه مقدار این معیار وظیفه کمتر باشد، آن وظیفه به الویت نزدیکتر است.

در وظیفه i زودترین زمان شروع EST[۴۱] و دیرترین زمان شروع LST[۴۲] آن وظیفه است. به این دلایل در هر تکرار، وظیفه‌های آماده براساس سطح‌شان مرتب می‌شوند. یک وظیفه آماده وظیفه‌ای است که قبل از زمان تصمیم‌گیری تمامی وظیفه‌های اجدادش خاتمه یافته باشد. همچنین بدیهی است که در زمان تصمیم‌گیری، یک وظیفه آماده وارد سیستم شده است. سپس الگوریتم، یک هسته برای اجرای آن وظیفه آماده را بر می‌گزینند. هسته مورد نظر برای اجرای این وظیفه نیازمند چندین شرط است:

۱- این هسته باید توانایی اجرای آن وظیفه را داشته باشد (خط ۱۰ الگوریتم ۱).

۲- هسته موردنظر در زمان زمان‌بندی باید آزاد باشد (خط ۱۱ الگوریتم ۱).

۳- ضمناً باید برای اجرای آن وظیفه باندازه کافی سریع باشد تا ضرب‌الاجل آن را بر آورده سازد (خط ۱۳ الگوریتم ۱).

۴- اگر هسته‌ای باتوانایی انجام قیود زمان‌بندی یافت نشد، الگوریتم خاتمه و غیرممکن بودن این زمان‌بندی را گزارش می‌دهد (خطوط ۲۹ الی ۳۰ الگوریتم ۱).

در حالی که آن وظیفه در حال اجرا روی هسته مورد نظر است باید تمام قیود انرژیِ آن برآورده شود (خط ۱۶ الگوریتم ۱). اگر این قیود برآورده نشوند، الگوریتم برای رفع این مشکل از تکنیک درج تاخیر استفاده می‌کند (خط ۲۳ الگوریتم ۱). شرح درج تاخیر در مقاله [۴۱] آمده است. نهایتاً این الگوریتم یک هسته مناسب از بین هسته‌های نامزد انتخاب می‌کند که بوسیله آن تقریب زمان پایان آن وظیفه و وظیفه فرزندش کمینه می‌شود (خط ۱۹ الگوریتم ۱). وظیفه فرزند، فرزندی از آن وظیفه در DAG آن کاربرد است که سطح بیشتری از آن وظیفه را دارد (خط ۱۷ الگوریتم ۱). انتخاب هسته براساس سریعترین زمان اجرای آن وظیفه و وظیفه فرزند بحرانی، در بیشینه کردن پویایی جانشینان آن وظیفه بسیار مهم و اساسی است.

باید توجه شود که ما برای بیان زمان اجرای وظیفه فرزند در گام انتخاب هسته، در تابع درج تاخیر (DelayInsertion) تغییرات موردنیاز را ایجاد کرده و آن را بهبود داده‌ایم. اگر وظیفه فرزند قابلیت اجرا روی هسته نامزد را نداشته باشد، ما زمان $2 \times hyperperiod$ را به عنوان زمان پایان وظیفه آن فرزند استفاده می‌کنیم. سپس تمامی هسته‌هایی که قبل از نقطه زمان‌بندی بعدی بیکار می‌شوند را به‌عنوان هسته آزاد علامت‌گذاری می‌کند (خط ۳۷ الگوریتم ۱). برای کامل شدن کارها در هر تکرار، مقادیر EST وظیفه‌های زمان‌بندی شده را به‌روزرسانی و تجدید می‌کند (خط ۳۹ الگوریتم ۱).

در نهایت ما در این مرحله از نتایج تخصیص‌ها و زمان‌بندی‌های مرحله قبل استفاده کرده و تعداد k عدد تخصیص و زمان‌بندی از از بین تمامی n تخصیص و زمان‌بندی را انتخاب می‌کنیم. این انتخاب براساس میزان دمای کمینه محاسبه شده برای تخصیص‌ها و زمان‌بندی‌ها صورت می‌گیرد. این انتخاب باید با وجود تقلیل جستجو از نظر توان مصرفی کمینه باشد در حالی که در همان زمان مقدار امید ریاضی دمای تراشه را نیز کمینه می‌کند. ما $F(n,k)$ را برای فرموله کردن تعریف این معیار بکار می‌بریم. $F(n,k)$ را به‌عنوان کمینه مقدار مورد انتظار دمای k عدد تخصیص و زمان‌بندی از بین تمامی n تخصیص‌ها و زمان‌بندی‌ها قرار می‌دهیم.

الگوریتم ۱- الگوریتم تخصیص و زمان‌بندی انرژی‌آگاه پیشنهادی

```
1  compute EST(j), LST(j) and level(j)  for all tasks;
2  compute  avgE; // Average of execution time
3  CurrentTime = 0;
4  while there are Unscheduled tasks do
5        CurrentDelay = 0;
6        RT = ready tasks in non-decreasing order of level;
7       for each  j∈ RT  do
8         initial invalidCount and fastest Core info variables;
9         for each  m∈ M  do
10          ifproc m can execute task  j then
11           if core  m at CurrentTime is free then
12                calculate end time of task_j on proc m;
13                if deadline constraints of task_j is satisfied and
                                    isWrappAroundValid( ) then
14                   compute energy profile for one Hyper period;
15                   calculate peak temperature for all cores;
16                  if energy constraints is satisfied then
17                      cchild  = find child of task_j;
18                      calculate end time cchild;
19                 if endTime[j]+endTime[cchild] is minimum then
20                        update core for task_j and its child;
21                        CurrentDelay = 0;
22        else
23         CurrentDelay=DelayInsertion(G(V,E), j, m, currentTime);
24                      update core for task_j;
25                else if core m is not busy then
26                                  InvalidCount++;
27           else if core m is not busy then
28                                  InvalidCount++;
29       ifInvalidCount = | M | then
30                              return  INFEASIBLE;
31       else if core is valid then
32              assign task_j on best core ;
33              best core is busy for end time of task_j;
34       ifCurrentDelay> o then
35   break; //allow no tasks to start executing between currentTime
             // and currentTime + currentDelay
36   calculate nextSchedulePoint;
37   set idle all cores that become idle at NextSchedulePoint;
38   CurrentTime  =NextSchedulePoint;
39   update EST(j), level(j), for all unscheduled tasks;
40  return  FEASIBLE;
```

الگوریتم ۲- الگوریتم انتخاب زمان‌بندی نهایی

```
for i= 0 to nSchedule do
    for j= 0 to nSelect do
        if j= 0 then
            for l= 0 to k do
                dpVec[i][j].meet [l]=0;
                dpVec[i][j].maxTemp[l]=Lmax;
        else if i= 0 then
            for o= 0 to k do
                dpVec[i][j].meet[o]=meet[i][o];
                dpVec[i][j].maxTemp[o]=maxTSchadule[i][o];
        else
            // Calculate candidate Γ i-1 j-1
            for l= 0 to k do
                calculateΓ i-1 j-1.meet[l];
                calculateΓ i-1 j-1.maxTemp[l];
            F i-1 j = calculateexpected peak-temp dpVec[i-1][j];
            F i-1 j-1= calculateexpected peak-temp Γ i-1 j-1;
            if F i-1 j-1≤ F i-1 j then
                dpVec[i][j]=Γ i-1 j-1;
                mark ith schedule;
            else dpVec[i][j]=dpVec[i-1][j];
```

جدول ۲- محک‌های داده‌ای و ابعاد شبکه مورد آزمایش

Benchmark	Type	Task No.	Edge No.
Auto- 3×3	Homogenus	24	21
Consumer-3×3	Hetrogenus	12	12
Networking-3×3	Homogenus	13	9
Office- 2×2	Homogenus	5	5
Mpeg4- 3×3	Homogenus	12	13
MWD- 3×3	Hetrogenus	12	12
Jpeg- 2×2	Homogenus	8	9
vopd- 3×3	Hetrogenus	12	14

جدول ۳- پردازنده‌های انتخابی حل مساله پیشنهادی

Index	Core
1	AMD K6-2E 400- MHZ/ACR
2	AMD K6-2E+ 500- MHZ/ACR
3	AMD K6-IIIE+ 550- MHZ/ACR
4	IBM PowerPC 405GP- 266 MHZ
5	Motorola MPC555- 40 MHZ

این تعریف واضح است زیرا بعضی از زمان‌بندی‌ها که توانسته‌اند قیود زمانی را برآورده کنند مقدار دمای آنها معتبر نیست. برای این موارد، ما از $L_{\max}$ به جای مقدار دما استفاده می‌کنیم که این پارامتر یک عدد ثابت است که با تنظیم مقدار آن می‌توان یک کران بالا برای حداکثر دمای تراشه در زمان اجرای کاربرد اعمال کرد. بر این اساس ما برای هر برنامه‌ریزی i یک پارامتر اندازگیری دما $T_{\max}^{sel}(i)$ به صورت رابطه (۱۴) تعریف می‌کنیم.

$$T_{\max}^{sel}(i) = \min_{s \in S_{sel}} \left[\max(V_{ins} \times T_{is}^{\max} \ , \ (1 - V_{ins}) \times L_{\max}) \right] \qquad (۱۴)$$

جایی که S_{sel} مجموعه زمان‌بندی‌های انتخابی، V_{is} موید این است که زمان‌بندی s^{th} می‌تواند برنامه i^{th} قیود زمان‌بندی را برآورده کند، و $T_{is}^{\max}$ حداکثر دمای اجرای زمان‌بندی تحت شرایط برنامه i^{th} (درصورت برآوردن قیود زمانی) است. پارامترهای V_{is} و $T_{is}^{\max}$ خروجی‌های مرحله قبل هستند. براساس تعریف $T_{\max}^{sel}$، ما می‌توانیم مقدار $F(n,k)$ را بوسیله رابطه (۱۵) بیان کنیم.

$$F(n,k) = \sum_{i=1}^{n_{event}} \rho(i) \times T_{\max}^{sel}(\mathrm{i}) \qquad (۱۵)$$

جایی که $\rho(i)$ همان احتمال وقوع برنامه i است و در مرحله اول محاسبه شده است. این مساله یک نمونه از مساله کوله‌پشتی است. لذا برای حل آن الگوریتمی بر پایه روش برنامه‌ریزی پویا ارایه می‌کنیم. برای انتخاب k زمانبندی از بین n زمان‌بندی نماد $F(n,k)$ نمایش دهنده آن است که $F(n,k)$ کمینه (یعنی مبین جواب مساله) است. به‌عبارت دیگر، $F(n,k)$ می‌تواند بوسیله رابطه بازگشتی رابطه (۱۶) ارایه شود.

$$F(n,k) = \min\{F(n-1,k), union(F(n-1,k-1),n)\} \qquad (۱۶)$$

فرض کنید که هر کدام از زمان‌بندی‌ها از ۱تا n شاخص‌بندی شود. آیا عنصر n^{th} در مجموعه انتخابی می‌تواند باشد. اگر به آن مجموعه تعلق نداشته باشد، $F(n,k)$ معادل $F(n-1,k)$ است. اگر عنصر n^{th} به آن مجموعه تعلق دارد، $T_{is}^{\max}$ باید تجدید و به‌روز شود. مقادیر $T_{is}^{\max}$ یا از زمانبندی n^{th} حاصل شده یا از (k-1) زمان‌بندی دیگر که در قبلا برگزیده شده‌اند و مقادیر آن در $F(n-1,k-1)$ بیان شده است. عمل union رابطه (۱۶) عملیات به‌روزرسانی مقادیر $T_{is}^{\max}$ را انجام می‌دهد. الگوریتم ۲ با زمان چندجمله‌ای مبین شبه کد آن است. مقدار اولیه $F(i,0)$ با $L_{\max}$ تنظیم می‌شود (خط ۳ الی ۶ الگوریتم ۲) و مقدار اولیه $F(0,j)$ با حداکثر دمای تولیدی در مرحله چهارم تنظیم می‌شود (خط ۷ الی ۱۰ الگوریتم ۲). تمامی مقادیر $F(n,k)$ با استفاده از روش پایین به بالا محاسبه می‌شود. اگر زمان‌بندی i^{th} به مجموعه انتخابی متعلق باشد، زمان‌بندی i^{th} علامت زده می‌شود (خط ۲۰ الگوریتم ۲). زمان‌بندی علامت زده شده همان نتیجه نهایی متد پیشنهادی است.

۵- آزمایشات تجربی

در این بخش به بررسی کارایی روش و الگوریتم پیشنهادی می‌پردازیم و نتایج شبیه‌سازی را در سه بخش تنظیمات آزمایشات، اعتبارسنجی مدل، و نتایج آزمایشات تجربی، ارزیابی و تحلیل می‌کنیم.

۵-۱- تنظیمات آزمایشات

الگوریتم زمان‌بندی انرژی‌آگاه ارائه شده در زبان C++ پیاده‌سازی و بوسیله محک‌های مختلف E3S[۴۳] [۳۹] در تکنولوژی ۹۵ نانومتر با توجه به پیش‌بینی‌های ITRS [۱۰] و کارهای موجود مورد ارزیابی قرارگرفته است. ما مجموعه داده‌ای E3S را بر اساس فاکتور مقیاس‌بندی تکنولوژی[۴۴] تغییر داده‌ایم. به این منظور اطلاعاتی مانند توان‌مصرفی، ابعاد عناصر و غیره را براساس فاکتورهای مقیاس‌بندی گزارش‌شده در [۴۲] [۴۳] تطابق داده‌ایم. E3S دارای چندین محک است که هرکدام ارایه‌کننده یک کاربرد خاص هستند. بعضی وظیفه‌ها در هر کاربرد دارای ضرب‌الاجل بی‌درنگ سخت هستند. مشابه [۴۴]، ما از ساختار معماری 2×2 که در جدول ۲ برای ساختارهای همگن و ناهمگن استفاده کرده‌ایم. اسامی پردازنده‌های استفاده شده در جدول ۳ آمده است. البته با توجه به‌محدودیت شدید زمانی محک Auto، مقدار تمامی ورودی‌های زمانی (ضرب‌الاجل و دوره) ۱.۵ برابر و پارامترهای مدل جریان نشتی را برای تکنولوژی مورداستفاده استخراج کرده‌ایم. برای بهینه‌سازی از ابزار CPLEX [45] و برای دما از HotSpot [46] بهره

برده‌ایم. ضمناً برای مقایسه نتایج حاصل، دو روش الگوریتم ژنتیک سنتی و پیشنهادی، یک روش شبیه‌سازی گداخت و یک روش قطعی برنامه‌ریزی عدد صحیح را پیاده‌سازی کرده‌ایم.

۵-۲- اعتبارسنجی مدل پیشنهادی

مساله را در ابعاد کوچک با روش قطعی برنامه‌ریزی خطی حل و با مقایسه نتایج اعتبار سنجی کرده‌ایم که به جواب دقیق بهینه دست می‌یابد. در ابعاد بزرگ به استناد جواب‌های یافت شده موجود بعضی مقالات و پاسخ‌های شبیه‌سازی گداخت، اطمینان از جواب مطلوب نزدیک بهینه حاصل شده است.

۵-۳- نتایج آزمایش‌های تجربی

نتایج آزمایش‌ها را از نظر الگوریتم نگاشت، الگوریتم زمان‌بندی و ارزیابی متدلوژی پیشنهادی برای زمان اجرای، سرعت همگرایی، و توان مصرفی با سایر روش‌ها ارزیابی می‌کنیم. برای ارزیابی الگوریتم‌ها، آن‌ها را با ۱۰۰ تکرار و چندین نسل تولیدی، با محک‌های مختلف اجرا کرده‌ایم.

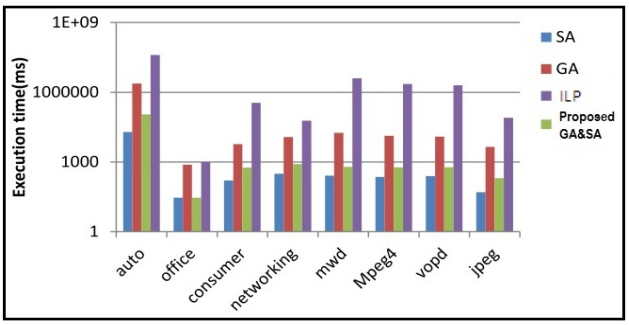
۵-۳-۱- ارزیابی الگوریتم نگاشت

زمان اجرای مرحله نگاشت الگوریتم پیشنهادی از الگوریتم ژنتیک کمتر و از الگوریتم شبیه‌سازی گداخت بیشتر است (شکل ۸). با افزایش تعداد وظایف، این اختلاف بسیار بیش‌تر می‌شود. البته احتمال‌گیر افتادن الگوریتم شبیه‌سازی گداخت در بهینه محلی زیاد و در صورت تکرار الگوریتم نیز مقادیر پاسخ تقریبی به دست آمده بهبود نخواهند یافت (جدول ۴). انرژی مصرفی مرحله نگاشت الگوریتم پیشنهادی از الگوریتم ژنتیک و شبیه‌سازی گداخت کمتر و از الگوریتم برنامه‌ریزی خطی بیشتر است (شکل ۹). در الگوریتم شبیه‌سازی گداخت احتمال گیرافتادن در بهینه محلی زیاد می‌باشد که این مورد در نمودار مشهود است.

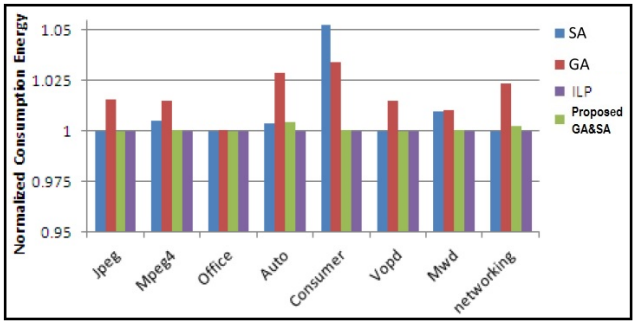
در نمودار شکل ۱۰، سرعت همگرایی الگوریتم‌ها در ۱۰۰۰ نسل متوالی برای محک‌های Jpeg و Office رسم شده است. همان‌طور که مشاهده می‌شود با استفاده از الگوریتم شبیه‌سازی گداخت و الگوریتم پیشنهادی برای این دو محک، بعد از گذشت تعداد کمی نسل، صد در صد به جواب بهینه دست می‌یابیم ولی با الگوریتم ژنتیک بعد از ۱۰۰۰ نسل به ترتیب به نرخ موفقیت ٪۶۲ و ٪۹۹ دست خواهیم یافت. البته الگوریتم ژنتیک در نهایت با تعداد تکرار بیش‌تر، پتانسیل رسیدن به جواب بهینه را دارد. الگوریتم پیشنهادی با تعداد تکرار بسیار اندک به جواب بهینه دست یافته و از سرعت همگرایی بسیار بیشتری برخوردار است.

جدول ۴- زمان پردازش مرحله نگاشت الگوریتم‌های ژنتیک، شبیه‌سازی گداخت، و پیشنهادی

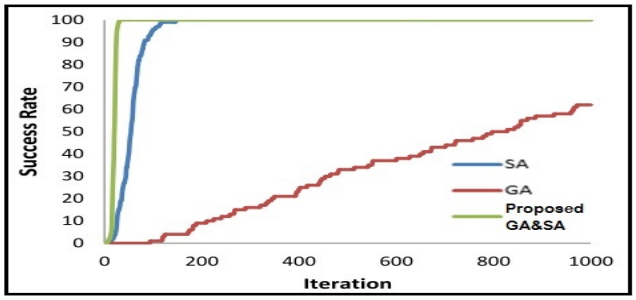
محک، الگوریتم	شبیه‌سازی گداخت	ژنتیک	پیشنهادی
auto	۱۹.۴۲	۲۳۷۱.۱۶	۱۱۳.۰۲
office	۰.۰۲۹	۰.۷۷	۰.۰۲۹
consumer	۰.۱۶	۵.۸۹	۰.۵۷
networking	۰.۳۱	۱۱.۹۳	۰.۸۲
mwd	۰.۲۶	۱۸.۱۵	۰.۶۲
Mpeg4	۰.۲۳	۱۳.۳۴	۰.۵۸
vopd	۰.۲۵	۱۲.۳۲	۰.۵۹
Jpeg	۰.۰۵	۴.۴۶	۰.۲



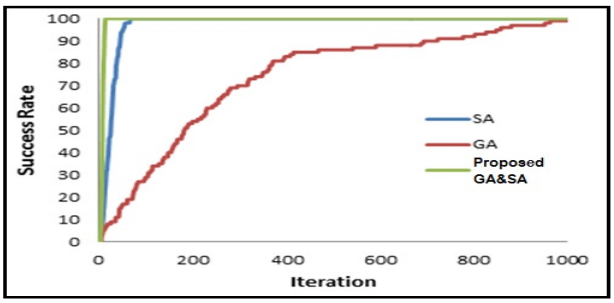
شکل ۸- سرعت اجرای نگاشت الگوریتم‌های ژنتیک، شبیه‌سازی گداخت، بهینه‌سازی و پیشنهادی



شکل ۹- انرژی مصرفی نگاشت الگوریتم‌های ژنتیک، شبیه‌سازی گداخت، بهینه‌سازی و پیشنهادی



(الف) همگرایی روش نگاشت در محک Consumer



(الف) همگرایی روش نگاشت در محک Office

شکل ۱۰- سرعت همگرایی الگوریتم‌های ژنتیک، شبیه‌سازی گداخت و پیشنهادی

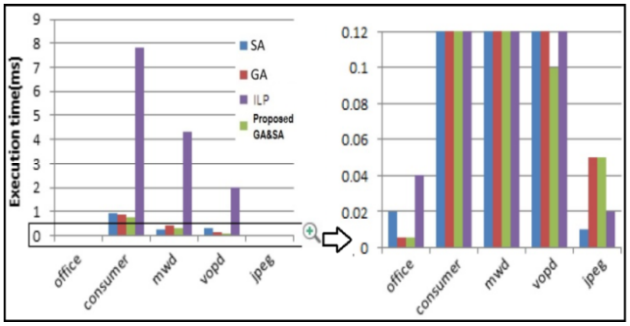
۵-۳-۲- ارزیابی الگوریتم‌های زمان‌بندی

زمان اجرای زمان‌بندی الگوریتم پیشنهادی از الگوریتم ژنتیک کمتر و از الگوریتم شبیه‌سازی گداخت بیشتر است (شکل ۱۱). با افزایش تعداد وظایف، این اختلاف

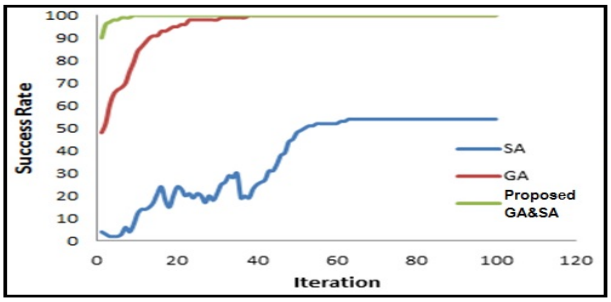
بسیار بیش‌تر می‌شود. البته احتمال‌گیر افتادن الگوریتم شبیه‌سازی گداخت در بهینه محلی زیاد و در صورت تکرار الگوریتم نیز مقادیر تقریبی به‌دست آمده بهبود نخواهند یافت (جدول ۵). الگوریتم شبیه‌سازی گداخت به صورت صد در صد به جواب نمی‌رسد و با تکرار الگوریتم نیز مقادیر تقریبی به دست آمده بهبود پیدا نخواهند کرد ولی الگوریتم ژنتیک بهبودیافته با دقت صد درصد به جواب می‌رسد. و زمان آن در برابر روش‌های قطعی بسیار ناچیز است (جدول ۵). انرژی مصرفی الگوریتم پیشنهادی نیز مانند نگاشت، از الگوریتم ژنتیک و شبیه‌سازی گداخت کمتر ولی از الگوریتم برنامه‌ریزی خطی بیشتر است.

جدول ۵- زمان اجرای زمان‌بندی الگوریتم‌های ژنتیک، شبیه‌سازی گداخت، قطعی و پیشنهادی

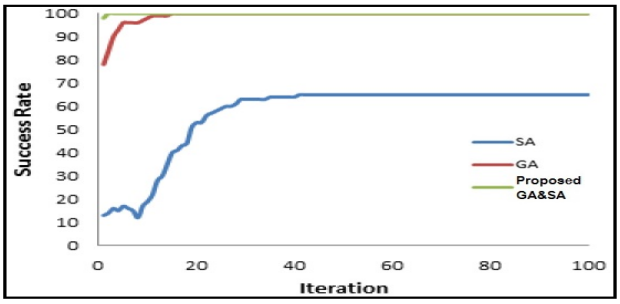
محک، الگوریتم	شبیه‌سازی گداخت	ژنتیک	پیشنهادی	قطعی
office	۰.۰۲	۰.۰۰۶	۰.۰۰۶	۰.۰٤
consumer	۰.۹۱	۰.۸۸	۰.۷۸	۷.۸۳
mwd	۰.۲۵	۰.٤	۰.۳۱	٤.۳٤
vopd	۰.۳	۰.۱۲	۰.۱	۲
jpeg	۰.۰۱	۰.۰۵	۰.۰۵	۰.۰۲



شکل۱۱- سرعت اجرای زمان‌بندی الگوریتم‌های ژنتیک، شبیه‌سازی گداخت، بهینه‌سازی و پیشنهادی



(الف) همگرایی روش زمان‌بندی در محک Consumer



(الف) همگرایی روش زمان‌بندی در محک MWD

شکل ۱۲- سرعت همگرایی الگوریتم‌های ژنتیک، شبیه‌سازی گداخت و پیشنهادی

در نمودار شکل ۱۲ سرعت همگرایی الگوریتم‌های پیاده‌سازی شده برای رسیدن به مقدار بهینه در ۱۰۰ نسل متوالی برای محک‌های Consumer و MWD ترسیم شده است. با مشاهده آنها بعد از گذشت تعدادی نسل به صورت ۱۰۰٪ به جواب بهینه دست می‌یابیم که دارای شیب خوب همگرایی هستند. همگرایی روش پیشنهادی در مقابل الگوریتم ژنتیک با افزایش نسل‌ها بیش‌تر استولی با الگوریتم شبیه‌سازی گداخت در ۱۰۰ بار اجرا به ترتیب به نرخ موفقیت ۵۴٪ و ۶۷٪ دست خواهیم یافت. به‌عبارتی در محک اول از ۱۰۰ آزمایش در تکرار ۶۳ تنها ۵۴ مورد به جواب بهینه دست یافته‌اند و مابقی در ۴۶ مورد به‌جواب رسیده‌اند. حتی با تکرار نیز به جواب بهینه نمی‌رسند و در تله بهینه‌های محلی گرفتار می‌شوند.

۵-۳-۳- ارزیابی متدلوژی پیشنهادی

با توجه به نتایج شبیه‌سازی و مقایسه زمان اجرا و انرژی مصرفی می‌توان بیان کرد، در مواردی که تعداد هسته‌های معماری و وظایف کاربرد کم باشد استفاده از روش‌های قطعی ممکن و تضمین‌کننده جواب بهینه است. به‌دلیل افزایش نمایی زمان محاسبات با افزایش تعداد وظایف کاربرد، این روش‌ها برای کاربردهای بزرگتر در مدت زمان خیلی زیاد جواب می‌دهند. وقتی که این تعداد از حد مشخصی بیشتر می‌شود مساله به دلیل افزایش نمایی فضای حافظه کامپیوتر غیر قابل حل خواهد بود لذا استفاده از روش‌های مکاشفه‌ای اجتناب ناپذیراست و جواب نزدیک به بهینه برای مساله را در مدت زمان کوتاهی ارایه می‌دهند. اگر پارامترهای اولیه خوبی برای روش شبیه‌سازی گداخت انتخاب گردد احتمال به دام افتادن آن در بهینه محلی کم و جواب خوبی ارایه می‌دهد.

روش الگوریتم ژنتیک تضمین‌کننده جواب خوب نخواهد بود زیرا ممکن است مقادیر نزدیک‌بهینه که در یک نسل به‌وجود آمده‌اند در نسل‌های بعدی با عملیات جهش و تقاطع بر روی آن‌ها حذف گردند و مجموعه جواب حاصل در نسل‌های بعدی یک نسل، نسبت به آن نسل دارای میانگین تابع ارزیاب بدتر باشد. ارایه روش ما تضمین‌کننده بهبود پس از مرحله‌جهش در الگوریتم پیشنهادی، این مشکل الگوریتم ژنتیک را برطرف می‌کند. استفاده از این روش موجب بهبود جواب در هر نسل می‌گردد که ضمن برخورداری از سرعت همگرایی بالا، در همزمانی نگاشت و زمان‌بندی وظایف و ارتباطات از سرعت اجرای خوبی نیز برخوردار است.

۶- نتیجه‌گیری

ما یک متدلوژی نگاشت و زمان‌بندی بی‌درنگ انرژی‌آگاه برای برنامه‌ریزی همزمان وظایف و ارتباطات با هدف حل سریع با جواب نزدیک بهینه در تراشه‌های چند هسته‌ای ارایه داده‌ایم که همزمان با کمینه‌سازی انرژی مصرفی و کاهش زمان اجرا، کار تولید جواب دقیق مساله را با کاهش تعداد جستجوها و با فرار از تله پاسخ‌های بهینه محلی انجام می‌دهد. متدلوژی پیشنهادی با برخورداری از ساختار نوین کروموزوم در الگوریتم ژنتیک و برخورداری از تابع جهش شبیه‌سازی گداخت، دارای قابلیت جلوگیری از تولید راه‌حل‌های غیرممکن جهت کاهش زمان تولید جواب نزدیک بهینه است. تحلیل نتایج آزمایشات نشان می‌دهد که در نگاشت و زمان‌بندی همزمان نسبت به روش سنتی ژنتیک از سرعت همگرایی و تولید جواب بسیار خوب همراه با تولید جواب نزدیک بهینه برخورداری است. همچنین قابل کاربرد برای طراحی سیستم‌های بزرگ و بی‌درنگ همگن و ناهمگن است. ما در برنامه آینده تحقیقاتی خود در حال افزودن قابلیت اطمینان به متدلوژی پیشنهادی و توسعه آن برای پشتیبانی از ساختارهای چندولتاژی هستیم.

مراجع

[1] A. Kumar, M. Shafique, A. Kumar, and J. Henke, "Mapping on Multi/Many-core Systems: Survey of Current and Emerging Trends," *Proc. DAC*, pp. 338-342, 2013.

[2] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE Micro*, vol. 25, pp. 10-16, 2005.

[3] O. Eduard, and et. al., "Energy Efficiency and Renewable Energy Iintegration in Data Centres," Strategies and modelling review, Renewable and Sustainable Energy Reviews, no.42, pp. 429-445, 2015.

[4] S. Mittal, "A Survey of Techniques for Improving Energy Efficiency in Embedded Computing Systems," *International Journal of Computing Aided Engineering and Technology*, vol. 6, no. 4, pp. 450-459, 2014.

[5] P. Nathaniel, D. Blaauw, and D. Sylvester, "Low-Power Near-Threshold Design: Techniques to Improve Energy Efficiency Energy-Efficient Near-Threshold Design Has Been Proposed to Increase Energy Efficiency Across a Wid," IEEE Solid-State Circuits Magazine, vol. 7, no.2, pp. 49-57, 2015.

[6] J. Sartori, A. Pant, R. Kumar, and P. Gupta, "Variation-aware Speed Binning of Multi-core Processors," *11th ACM/IEEE International Symposium on Quality Electronic Design, ISQED*, San Jose, 2010.

[7] R. Viswanath, V. Wakharkar, A. Watew, and V. Lbonheur, "Thermal Performance Challenges from Silicon to Systems," *Intel Technology Journal*, vol.4, no.3, pp. 1-16, 2000.[Online].Available:http://www.intel.com/technology/itj / q32000/articles/art_4.htm.

[8] L. Y. Lin, and et. al., "Communication-driven Task Binding for Multiprocessor with Latency Insensitive Network-on-chip," *ASP-DAC*, pp. 39–44, 2005.

[9] G. Ascia, V. Catania, and M. Palesi, "Multi-objective Mapping for Mesh-based NoCArchitectures," *CODES+ISSS*, pp. 182–187, 2004.

[10] International technology roadmap for semiconductors, 2010, http://www.itrs.net/Links/2010ITRS/Home2010.htm.

[11] A. Mahabadi, SM. Zahedi, and A. Khonsari, "Reliable Energy-aware Application Mapping and Voltage–frequency IslandPartitioning for GALS-based NoC," *Journal of Computer and System Sciences*, vol. 79, no.4, pp.57–74, 2013.

[12] B. Khodabandeloo, A. Khonsari, F. Gholamian, M. H. Hajiesmaili, A. Mahabadi, and H. Noori, "Scenario-based Quasi-static Task Mapping and Scheduling for Temperature-efficient MPSoCDesign under Process Variation," *Microprocessors and Microsystems*, vol. 38, pp. 399–414, 2014.

[13] J. Castrillon, A. Tretter, R. Leupers, and G. Ascheid, "Communication-aware Mapping of KPN Applications onto Heterogeneous MPSoCs," *DAC*, pp. 1266–1271, 2012.

[14] W. Che, and K. S. Chatha, "Unrolling and Retiming of Stream Applications onto Embedded Multicore Processors," *DAC*, pp. 1272–1277, 2012.

[15] S. Gupta, G. Agarwal, and V. Kumar, "Task Scheduling in Multiprocessor System Using Genetic Algorithm," *Machine Learning and Computing (ICMLC), Second International Conference*, pp. 267-271, 2010.

[16] S. Sivanandam, and P. Visalakshi, "Dynamic Task Scheduling with Load Balancing using Parallel Orthogonal Particle Swarm Optimisation," *International Journal of Bio-Inspired Computation*, vol. 1, pp. 276-286, 2009.

[17] S. Ninomiya, K. Sakanushi, Y. Takeuchi, and M. Imai, "Task Allocation and Scheduling for Voltage-Frequency Islands Applied NoC-based MPSoC Considering Network Congestion," *Embedded Multicore Socs (MCSoC), IEEE 6th International Symposium*, pp. 107-112, 2012.

[18] G. Chen, F. Li, S. and Son, M. Kandemir, "Application Mapping for Chip Multiprocessors," *DAC*, pp. 620–625, 2008.

[19] S. Banerjee, and N. Dutt, "Efficient Search Space Exploration for HW-SW Partitioning," *International Conference on Hardware/Software Codesign and System Synthesis*, pp. 122-127, 2004.

[20] A. Hartman, D. Thomas, and B. Meyer, "A Case for Lifetime-aware Task Mapping in Embedded Chip Multiprocessors," *CODES+ISSS*, pp. 145 –154, 2010.

[21] E. Seo, Y. Koo, and J. Lee, "Dynamic Repartitioning of Real-time Schedule on a Multicore Processor for Energy Efficiency," *Proc. Int. Conf. Embedded and Ubiquitous Computing*, pp. 69–78, Aug. 2006.

[22] J. Hu, and R. Marculescu, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE Trans. Comp.-Aided Des. Integ. Cir. Sys.*, no. 4, pp. 551–562, 2005.

[23] C. Marcon, A. Borin, A. Susin, L. Carro, and F. Wagner, "Time and energy efficient mapping of embedded applications onto NoCs," *ASP-DAC*, pp. 33–38, 2005.

[24] C. Marcon, E. Moreno, N. Calazans, and F. Moraes, "Comparison of network-on-chip mapping algorithms targeting low energy consumption. Computers Digital Techniques," *IET*, pp. 471–482, 2008.

[25] B. H. Meyer, A. S. Hartman, and D. E. Thomas, "Cost-effective Slack Allocation for Lifetime Improvement in NoC-based MPSoCs," *DATE*, pp. 1596–1601, 2010.

[26] H. Orsila, and et. al., "Automated Memory-aware Application Distribution for Multi-processor System-on-Chips," *J. Syst. Archit.*, no. 11, pp.795–815, 2007.

[41] T. Chantem, X.S. Hu, and R.P. Dick, "Temperature-aware Scheduling and Assignment for Hard Real-time Applications on MPSoCs," *IEEE Trans. VLSI System*, pp. 1884-1897, 2011.

[42] G. Link, and N. Vijaykrishnan, "Thermal Trends in Emerging Technologies," In Proc. Int. Symp. Quality of Electronic Design, pp. 625–632, 2006.

[43] W. Huang, K. Rajamani, M. R. Stan, and K. Skadron, "Scaling with Design Constraints Predicting the Future of Big Chips," *IEEE Micro special issue on Big Chips*, 2011.

[44] M. Momtazpour, E. Sanaei, and M.Goudarzi, "Power-yield Optimization in MPSoCTask Scheduling under Process Variation," ISQED, pp. 747-754, 2010.

[45] CPLEX 11.1ILOG: http://www.ilog.com/product/cplex/, 2013.

[46] HotSpot: http://lava.cs.virginia.edu/HotSpot/, 2013.

[47] R. Teodorescu, B. Greskamp, J. Nakano, S. Sarangi, A. Tiwari, and J. Torrellas, "VARIUS: A Model of Parameter Variation and Resulting Timing Errors for Microarchitects," 2nd Workshop *on Architectural Support for Gigascale Integration*, San Diego, USA, 2007.

[48] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects," *IEEE Trans. on Semiconductor Manufacturing*, vol. 21, no. 1, 2008.

[49] A. Bonfietti, L. Benini, M. Lombardi, and M. Milano, "An Efficient and Complete Approach for Throughput-maximal SDFAllocation and Scheduling on Multi-core Platforms," *DATE*, pp. 897–902, 2010.

[50] N. Satish, K. Ravindran, and K. Keutzer, "ADecomposition-based Constraint Optimization Approach forStatically Scheduling Task Graphs with Communication Delaysto Multiprocessors," *DATE*, pp. 57–62, 2007.

[51] L. Thiele, L. Schor, H. Yang, and I. Bacivarov, "Thermal-awareSystem Analysis and Software Synthesis for Embedded Multi-processors," *DAC*, pp. 268–273, 2011.

[52] D. Wu, B. M. Al-Hashimi, and P. Eles,"Scheduling and Mapping of Conditional Task Graphs for the Synthesis of LowPower Embedded Systems," *DATE*, pp. 10090, 2003.

[53] Y. Markovskiy, E. Caspi, R. Huang, J. Yeh, M. Chu, J. Wawrzynek, and A. DeHon, Analysis of Quasi-Static Scheduling Techniques in a Virtualized Reconfigurable Machine. Proceedings of ACM/SIGDA Tenth International Symposium on Field-Programmable Gate Arrays, pp. 196-205, 2002.

[27] X. Wu, and et. al., "Genetic Algorithms for Integrating Cell Formation with Machine Layout and Scheduling," *Computers & Industrial Engineering*, vol. 5, no. 2, pp. 277-289, 2007.

[28] J. Choi, H. Oh, S. Kim, and S. Ha, "Executing Synchronous Dataflow Graphs on a SPM-based Multicore Architecture," *DAC*, pp. 664–671, 2012.

[29] S. Manolache, P. Eles, and Z. Peng, "Task Mapping and Priority Assignment for Soft Real-time Applications under Deadline Miss Ratio Constraints," *ACM Trans. Embed. Comput. Syst.,* vol. 19, pp.13–19, 2008.

[30] H. Javaid, and S. Parameswaran, "A Design Flow for Application Specific Heterogeneous Pipelined Multiprocessor Systems," *DAC*, pp. 250–253, 2009.

[31] M. Ruggiero, and et. al., "Communication-aware Allocation and Scheduling Framework for Stream-oriented Multi-processor Systems-on-chip," *DATE*, pp. 3–8, 2006.

[32] L. Thiele, I. Bacivarov, W. Haid, and K. Huang, "Mapping Applications to Tiled Multiprocessor Embedded Systems," *ACSD*, pp. 29–40, 2007.

[33] S. Murali, M. Coenen, A. Radulescu, K. Goossens, and G. De Micheli, "A methodology for Mapping Multiple Use-cases onto Networks on Chips," *DATE*, pp. 118–123, 2006.

[34] C.-E. Rhee, H.-Y. Jeong, and S. Ha, "Many-to-Many Core-Switch Mapping in 2-D Mesh NoC Architectures," *ICCD*, pp. 438–443, 2004.

[35] C. M. Chen, and C. T. King, "Using Integer Linear Programming for Instruction Scheduling and Register Allocation in Multi-issue Processors," *Multi-Issue Processors. Computers and Mathematics with Applications*, 1997.

[36] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotLeakage: A temperature-aware Model of Sub Threshold and Gate Leakage for Architects," *Tech. Rep. CS-2003-05*, University of Virginia, 2003.

[37] K. Bowman, S. Duvall, and J. Meindl, "Impact of Die-to-die and within Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Giga Scale Integration," *IEEE J. Solid State Circuits*, vol. 37, no. 2, pp. 183-190, 2002.

[38] W. Zhang, G. Sun, and S. Bin, "A Novel Task Communication and Scheduling Algorithm for NoC-based MPSoC," *International Journal of Smart Home*, vol. 9, no. 10, pp. 179-188, 2015.

[39] R. Dick. "Embedded Systems Synthesis Benchmarks Suite (e3s)," http://www.ece.northwestern.edu/ dickrp/e3s/.

[40] E. L. Lawler, and C. U. Martel, "Scheduling Periodically Occurring Tasks on Multiple Processors," *Information Processing Ltrs.*, vol. 7, no. 1, pp. 9-12,1981.

امین‌اله مه‌آبادی تحصیلات خود را در رشته مهندسی برق سخت‌افزار و معماری کامپیوتر به انجام رسانده و اکنون استادیار گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه شاهد است. تحقیقات مورد علاقه نامبرده طراحی مدارات مجتمع، مدیریت منابع بر تراشه، زمان‌بندی شبکه بر تراشه و طراحی ابزارهای شبیه‌سازی هوشمند است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

mahabadi@shahed.ac.ir

فاطمه عسگری بیدهندی لیسانس خود را در رشته مهندسی کامپیوتر نرم‌افزار از دانشگاه الزهرا و فوق‌لیسانس خود را در رشته معماری کامپیوتر از دانشگاه شاهد اخذ نموده است. تحقیقات مورد علاقه نامبرده معماری کامپیوتر، زمان‌بندی و مدیریت منابع است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

fateme.asgari@gmail.com

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۸/۲۸

تاریخ اصلاح: ۱۳۹۴/۰۹/۲۹

تاریخ قبول شدن: ۱۳۹۴/۱۰/۱۵

نویسنده مرتبط: دکتر امین‌اله مه‌آبادی، دانشکده فنی و مهندسی، دانشگاه شاهد، تهران، ایران.

[1]Performance
[2]Semiconductor
[3]Multiprocessor System on Chip (MPSoC)
[4]Power Density
[5]Leakage Power
[6]Deep Submicron
[7]Embedded System
[8]Application
[9]Constraint
[10]Hardware Task
[11]Software Task
[12]Digital Signal Processing(DSP)
[13]Accelerator
[14] Integer Linear Programming (ILP)
[15] Condition Linear Programming (CLP)
[16]Non Linear Programming (NLP)
[17] Mixed Integer Linear Programming (MILP)
[18]Homogenous
[19]NP-Hard
[20] Constructive
[21] Transformative
[22] Branch-and-Bound (BB)
[23]Heuristic
[24] Genetic Algorithm (GA)
[25] Simulated Annealing (SA)
[26] Network on Chip (NoC)
[27] Pareto
[28]One-point Crossover
[29] 2D Mesh
[30]Non-Preemptive Scheduling
[31]Critical Path
[32]Lognormal
[33] Mesh
[34] Directed Acyclic Graph (DAG)
[35] Source
[36]Sink
[37]General
[38]Speed Binding
[39]Scheduler
[40]Hyper Periodic
[41]Earliest Start Time (EST)
[42]Latest Start Time (LST)
[43]Benchmarks
[44]Technology Scaling

علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۴۷-۵۹
مقاله عادی

نظرکاوی بین‌زبانی با استفاده از ویژگی‌های معنایی

شیما اسمعیلی تفت آزاده شاکری

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

چکیده

نظرکاوی یکی از زیربخش‌های متن‌کاوی است. در این حوزه به بررسی متن‌های نظرمند پرداخته می‌شود و هدف تشخیص مثبت و یا منفی بودن مفهوم این متن‌ها است. روش‌ها و راه‌حل‌های پیشنهادی در این حوزه به دو دسته باناظر و بدون ناظر دسته‌بندی می‌شود. از آن‌جا که روش‌های باناظر کارایی و دقت بالاتری نسبت به روش‌های بدون ناظر دارد، سعی می‌شود تا آن‌جایی که امکان دارد شرایط برای استفاده از روش‌های باناظر فراهم شود. اصلی‌ترین نیازمندی این روش‌ها، داده‌های برچسب‌خورده، به عنوان داده آموزش، در دامنه و زبان داده‌های آزمون است. وجود چنین داده‌هایی در تمام جفت دامنه و زبان‌ها محدودیتی برای استفاده از این گونه روش‌ها محسوب می‌شود. با توجه به زمان‌بر و پرهزینه بودن تولید داده‌های برچسب‌خورده به عنوان داده‌های آموزش، معمولاً ایجاد چنین مجموعه داده‌ای به عنوان بهترین راه‌حل در نظر گرفته نمی‌شود. همچنین به دلیل بیان متفاوتی که در دامنه‌ها و زبان‌های متفاوت وجود دارد، استفاده از داده‌های آموزش موجود در دامنه و یا زبان متفاوت به طور مستقیم موجب کاهش کارایی روش‌ها می‌شود. اما وجود داده‌های آموزش در اکثر دامنه‌ها در زبان‌های با منابع غنی انگیزه‌ای برای استفاده غیرمستقیم از این داده‌ها برای نظرکاوی داده‌های آزمون در زبان‌های دیگر ایجاد می‌کند. از این رو روش‌هایی به عنوان روش‌های بین‌زبانی ارائه شد که در آن‌ها از داده‌های آموزش موجود در زبان متفاوت با داده‌های آزمون، برای استخراج اطلاعات و در نهایت انتقال اطلاعات به زبان مورد نظر، بهره می‌برد. در این مقاله روشی برای نظرکاوی بین‌زبانی ارائه می‌شود که این استخراج و انتقال اطلاعات با کیفیت بالایی انجام می‌شود و همچنین برای اکثر زبان‌ها، حتی زبان‌های منابع‌محدود نیز قابل استفاده می‌باشد و به منابع موجود در زبان مورد نظر وابستگی کمی دارد. این روش با استفاده از داده‌های بدون برچسب در هر دو زبان مبدأ و مقصد، یک گراف دوبخشی بین دو دسته از ویژگی‌های محوری و غیرمحوری می‌سازد و ویژگی‌های معنایی را از آن استخراج می‌کند. تنها منبع مورد نیاز برای این روش، یک لغت‌نامه است که به دلیل استفاده از تعداد محدودی از ترجمه‌های آن، میزان وابستگی بالایی به این منبع ندارد.

کلمات کلیدی: نظرکاوی، تحلیل نظرات، ویژگی محوری، ویژگی معنایی، بین‌زبانی، ویژگی مستقل از دامنه، ویژگی وابسته به دامنه، گراف دوبخشی، رده‌بندی.

۱- مقدمه

امروزه با پیشرفت تکنولوژی و گسترش میزان استفاده از اینترنت و فضای مجازی، تولید محتوا توسط همه کاربران امکان‌پذیر شده است. نوشتن وبلاگ، به‌روزرسانی صفحه شخصی در شبکه‌های اجتماعی و وارد کردن نظرات و تجربیات شخصی در مورد یک مطلب از جمله این محتواها است.

نظرات و تجربیات شخصی می‌تواند درباره یک محصول و یا یک خدمت باشد، کاربری که از یک محصول یا خدمت استفاده کرده، بازخوردی در رابطه با آن بیان می‌کند. این نظرات که می‌توانند برای افراد و کاربران دیگر که قصد استفاده از این محصول و یا خدمت را دارند، مفید واقع شود، لحظه به لحظه در حال افزایش و حجیم شدن است. حوزه تحلیل نظرات[۱] و یا نظرکاوی[۲] سعی در تشخیص خودکار مثبت و یا منفی بودن و یا به عبارت دیگر تشخیص جهت احساسی[۳] این نظرات دارد تا بتوان خلاصه‌ای از نظرات وارد شده در مورد یک محصول و یا خدمت به کاربران جدید ارائه داد و آن‌ها را برای تصمیم‌گیری درست هدایت کرد. علاوه بر کاربرد ذکر شده، جمع‌آوری خلاصه‌ای از نظرات داده شده، می‌تواند برای تولیدکنندگان و صاحبان مشاغل نیز بسیار مفید واقع شود، زیرا می‌توانند به ارزیابی مناسبی از محصولات و خدمات ارائه شده خود دست پیدا کنند. این حوزه که زیربخشی از حوزه متن‌کاوی[۴] است، تنها داده‌های نظرمند[۵] را بررسی می‌کند و از داده‌های بدون نظر[۶] صرف نظر می‌کند.

حوزه تحلیل نظرات بیش از یک دهه است که مورد توجه محققین و پژوهش‌گران قرار گرفته است [۱، ۲]. تا به حال، روش‌های باناظر [۲، ۳] و بدون

ناظر [۱، ۴] متعددی برای بهبود کارایی این پیش‌بینی‌ها، پیشنهاد شده است. دسته اول روش‌های باناظر است که با استفاده از داده‌های برچسب‌خورده و استخراج ویژگی‌ها به تحلیل نظرات می‌پردازد. این روش‌ها عموماً دارای کارایی بالا است و وجود داده‌های برچسب‌خورده به عنوان داده آموزش در دامنه و زبان داده‌های آزمون از پیش‌نیازهای آن‌ها به شمار می‌رود. اما در برخی دامنه‌ها و یا زبان‌ها داده‌های برچسب‌خورده در حجم و کیفیت مناسبی در دسترس نیست. دسته دوم روش‌های بدون ناظر است. مزیتی که روش‌های بدون ناظر دارد، عدم نیاز آن‌ها به داده‌های برچسب‌خورده است. این روش‌ها به استخراج کلمات و عباراتی می‌پردازد که دارای جهت احساسی باشد و سپس با استفاده از آن‌ها برچسب نظرات را پیش‌بینی می‌کند. این روش‌ها معمولاً در مقایسه با روش‌های دسته اول دارای کارایی پایین‌تری است.

استفاده از اینترنت و تولید محتوا محدود به افراد و کاربران خاصی نیست و هر فرد با هر زبانی امکان تولید محتوا به زبان خود را دارد. بدیهی است که زبان‌های مختلف، ویژگی‌ها و خصلت‌های متفاوتی دارد و این اختلاف تنها به استفاده از کلمات متفاوت خلاصه نمی‌شود. همچنین برخی زبان‌ها نسبت به سایر زبان‌ها رایج‌تر بوده و در نتیجه منابع و داده‌های بیش‌تری از آن‌ها در اختیار است که در چنین زبان‌هایی کار تحلیل نظرات ساده‌تر انجام می‌شود. اما در زبان‌هایی که با کمبود داده و یا منابع مواجه است، کار دشوارتر می‌شود. از آن‌جا که تولید مجموعه داده‌ای برچسب‌خورده کاری زمان‌بر و پرهزینه است و ممکن است چنین داده‌هایی در حجم مناسب موجود نباشد، یکی از راه‌حل‌ها، استفاده از داده‌های موجود در زبان‌های دیگر است. اما داده‌های دو زبان متفاوت دارای ویژگی‌های کاملاً متفاوتی است که برای به‌کارگیری این داده‌ها نیاز به روش‌هایی برای برطرف کردن مانع است. این روش‌ها که روش‌های بین‌زبانی خوانده می‌شود، تلاش می‌کند از نظرات برچسب‌خورده موجود در زبان دیگر (زبان مبدأ) برای پیش‌بینی برچسب نظرات در زبان مورد نظر (زبان مقصد) استفاده کند.

بدیهی است که دو زبان مختلف، تفاوت‌هایی با هم دارد و به عبارتی دارای توزیع متفاوتی است. روش‌های بین‌زبانی تلاش می‌کند با به‌کارگیری راه‌حلی امکان استفاده از داده‌های موجود در یک زبان برای پیش‌بینی برچسب نظرات در زبان دیگر را فراهم کند تا به کیفیت روش‌های تک‌زبانه نزدیک‌تر شود. ترجمه یکی از داده‌ها به زبان دیگر و هم‌زبان شدن داده‌های آموزش و آزمون از ساده‌ترین راه‌حل‌های موجود به شمار می‌رود [۵، ۶]. اما برای بسیاری از زوج‌زبان‌ها منابع مناسبی برای ترجمه داده‌ها با کیفیت بالا از یک زبان به زبان دیگر در اختیار نیست و در نتیجه لازم است روش‌های دیگری که وابستگی کمتری به این منابع دارد ارائه شود و این محدودیت‌ها را برای این‌گونه از زبان‌ها کاهش دهد.

این مقاله با استفاده از ایده خوشه‌بندی گراف، به ساخت گرافی دوبخشی و دوزبانه می‌پردازد. این گراف بین ویژگی‌های محوری و غیرمحوری استخراج شده از تعداد زیادی نظرات بدون برچسب در هر دو زبان مبدأ و مقصد ساخته می‌شود. ویژگی‌های محوری جفت‌کلمه‌هایی است که هر کلمه، ترجمه کلمه دیگر در زبان دیگری است. با استفاده از یک منبع ترجمه، ویژگی‌های محوری که مستقل از زبان و یا به عبارت دیگری بدون ابهام است، انتخاب می‌شود. اما در مقابل ویژگی‌های غیرمحوری تک‌کلمه‌هایی است که احتمال وجود ابهام در ترجمه آن‌ها بیش‌تر تخمین زده شده است و در نتیجه این ویژگی‌ها ترجمه نمی‌شود و به صورت تک‌کلمه مورد استفاده قرار می‌گیرد. با استفاده از گراف ساخته شده و یکی از الگوریتم‌های پیشنهاد شده برای خوشه‌بندی گراف [۷]، ویژگی‌ها با احتمالی به هر خوشه تعلق می‌گیرد. با انتخاب تعدادی از بهترین خوشه‌ها و استفاده از داده‌های آموزش، یک رده‌بند ایجاد می‌شود که با توجه به دوزبانه بودن خوشه‌ها، برای داده‌های آزمون نیز قابل استفاده است.

روش پیشنهادی توانسته نسبت به سایر روش‌های بین‌زبانی وابستگی خود را به منابع ترجمه کاهش دهد. در این روش نیازی به ترجمه کل داده‌های یک زبان به زبان دیگر نیست و تنها از ترجمه کلمات استفاده می‌شود. همچنین در این روش تمام کلمات ترجمه نمی‌شود و می‌توان با ترجمه کردن تعداد محدودی از کلمات، به کارایی بالایی دست یافت. علاوه بر این، رده‌بند ایجاد شده برای رده‌بندی نظرات در هر دو زبان قابل استفاده می‌باشد و تنها به یکی از زبان‌ها اختصاص ندارد.

نتایج ارزیابی روش پیشنهادی نشان می‌دهد که در مقایسه با دو روش پایه انتخابی دارای اختلاف معنادار از لحاظ آماری است و این نکته بیان‌گر این است که توانسته نمایش بهتری از داده‌ها در دو زبان ارائه کند و از ارتباط بین دو دسته از ویژگی‌ها اطلاعات مفیدی استخراج کند. این روش همین‌طور که ذکر شد، از وابستگی کمی به منبع ترجمه برخوردار است و همچنین حساسیت کمی به تعداد خوشه‌های انتخابی از خود نشان داده است. علاوه بر این‌ها تعداد ویژگی‌های غیرمحوری به عنوان عامل تأثیرگذاری بر کارایی روش پیشنهادی تشخیص داده نشد.

در ادامه ابتدا در بخش ۲ مروری بر کارهای پیشین صورت گرفته در حوزه نظرکاوی صورت می‌گیرد و سپس در بخش ۳ روش پیشنهادی مطرح می‌شود و در بخش ۴ ارزیابی‌های انجام شده برای این روش ارائه می‌شود.

۲- کارهای پیشین

در این بخش به بررسی و مرور کارهایی که در گذشته در این حوزه صورت گرفته‌اند، پرداخته می‌شود. همان‌طور که قبلا ذکر شد، پژوهش‌های انجام‌گرفته در این حوزه را می‌توان به بخش‌ها و زیربخش‌هایی تقسیم کرد. پژوهش‌های بسیاری به تحلیل نظرات به صورت پایه‌ای می‌پردازند [۴-۱]. در [۱] با استفاده از اختلاف اطلاعات متقابل نظرات و یک کلمه مرجع[۷] مثبت و اطلاعات متقابل نظرات و یک کلمه مرجع منفی، به صورت بدون ناظر، به رده‌بندی نظرات پرداخته می‌شود.

روش دیگری در [۲] ارائه شده است که با استفاده از ویژگی‌های ۱-گرام، ۲-گرام و برچسب مقوله نحوی[۸]، سه روش یادگیری ماشین را با هم مقایسه می‌کند. در [۸] این ایده استفاده از روش‌های یادگیری ماشین بهبود می‌یابد. در این روش جدید ابتدا جملات بدون نظر از متن نظرات حذف می‌شود تا متن نظرات خلاصه‌تر و شفاف‌تر شود. سپس الگوریتم تحلیل نظرات پیشنهاد شده بر روی متن‌های جدید اجرا می‌شوند.

در [۹] تنها با استفاده از صفت‌ها، قیدها و فعل‌های موجود در نظرات، به پیش‌بینی جهت احساسی نظرات پرداخته می‌شود و در [۱۰] با استفاده از محاسبه اختلاف مدل زبانی[۹] نظر با مدل زبانی نظرات مثبت و نظرات منفی، شبیه‌ترین مدل زبانی به نظر شناسایی می‌شود. با توجه به بیان متفاوت احساسات در متن‌های رسمی و غیررسمی، در [۱۱] به بررسی متن‌های غیررسمی پرداخته می‌شود. در سال‌های اخیر کاربرد نمایش طیفی کلمات[۱۰] [۱۲] در تحلیل نظرات با در نظر گرفتن جهت احساسی متون در [۱۵-۱۳] بررسی می‌شود.

یک دسته از پژوهش‌ها در حوزه نظرکاوی، به بررسی نظرات بیان شده به شکل توئیت، در شبکه اجتماعی توئیتر[۱۱] می‌پردازد [۱۶، ۱۷]. توئیت‌ها کوتاه، غیررسمی و حاوی نمادهای خاص و اصطلاحات عامیانه هستند. این نظرات به دلیل تفاوت‌هایی که با نظرات بیان‌شده در تارنماهای مبتنی بر نظر دارند، لازم است روش‌هایی متناسب با خصوصیت‌هایشان ارائه و پیشنهاد شوند.

در ادامه، ابتدا در بخش ۱-۲ کارهای گذشته در حوزه بین‌زبانی مطرح می‌شوند، رویکردهای متفاوت روش‌های گوناگون بیان می‌شود و بعد از آن در بخش ۲-۲ حوزه بین‌دامنه‌ای بررسی می‌شود و خلاصه‌ای از روش‌های پیشنهاد شده در این حوزه معرفی می‌شود.

۲-۱- حوزه بین‌زبانی

کارهای انجام شده در حوزه بین‌زبانی نیز رویکردهای متفاوتی به مسأله دارد. برای مثال، در [۱۸] از یک لغت‌نامه احتمالاتی در سطح کلمه برای ترجمه نظرات در زبان چک به زبان انگلیسی استفاده می‌شود و مسأله بین‌زبانی به مسأله‌ای در حوزه تک‌زبانه تبدیل می‌شود. در [۱۹] با استفاده از سه نوع لغت‌نامه متفاوت و میزان گرایش معنایی موجود برای کلمات در زبان انگلیسی، گرایش معنایی برای کلمات اسپانیایی نیز به دست آورده می‌شود. در [۵] از ایده استفاده کردن از نظرات بدون برچسب استفاده می‌شود و به جای تنها ترجمه نظرات از یک زبان به زبان دیگر، نظرات بدون برچسب چینی به انگلیسی و همچنین نظرات برچسب‌دار انگلیسی به چینی توسط ماشین ترجمه، ترجمه می‌شود. از این رو نظرات بدون برچسب و با برچسب در هر دو زبان تولید می‌شود. درنتیجه رده‌بندی در هر دو زبان به دست می‌آید که با استفاده از ترکیب این دو رده‌بند، نظرات چینی و ترجمه شده آن‌ها به انگلیسی، برچسبی به نظرات زده می‌شود. در [۶] نیز این ایده گسترش می‌یابد که به بهبود کارایی روش منجر می‌شود.

همان‌طور که گفته شد، مشکل اساسی موجود در حوزه بین‌زبانی، اختلاف بین توزیع ویژگی‌ها در زبان‌های مختلف است و با ترجمه کردن داده‌های موجود در یک زبان دیگر به زبان مورد نظر، همچنان این اختلاف وجود دارد. روشی که در [۲۰] ارائه می‌شود، این اختلاف توزیع محاسبه می‌شود و تا حد امکان کاهش می‌یابد. برای این کار ویژگی‌هایی که بین دو توزیع احتمالاتی در دو زبان بیشترین اختلاف را دارد برای استنباط بهتر اختلاف دو توزیع، مفیدتر شناخته می‌شود. ایده این روش وزن‌دهی ویژگی‌ها و نمونه‌های موجود است. که سعی می‌شود به وسیله آن‌ها توزیع دو زبان هر چه بیش‌تر به هم شبیه‌تر شود. استفاده از نیروی انسانی راه دیگری برای افزایش کارایی روش‌های بین‌زبانی است که در [۲۱] مطرح می‌شود. در این روش نظرات بدون برچسب در زبان مورد نظر با استفاده از ماشین ترجمه به زبان مبدأ ترجمه می‌شود. این نظرات ترجمه شده با استفاده از رده‌بند حاصل از داده‌های برچسب‌خورده در زبان مبدأ رده‌بندی و سپس بهترین نمونه‌ها انتخاب می‌شود. این نمونه‌ها توسط یک خبره برچسب زده می‌شود و به داده‌های برچسب‌خورده در زبان مبدأ برای ایجاد یک رده‌بند بهتر اضافه می‌شود. این روند تا رسیدن به کارایی مطلوب تکرار می‌شود.

یکی از روش‌های تحلیل نظرات استفاده از واژه‌نامه[۱۲] است. در حوزه بین‌زبانی سعی می‌شود از یک واژه‌نامه در یک زبان برای برچسب‌گذاری نظرات در زبان دیگر استفاده شود. در [۲۲] یک نمونه از این روش‌ها ارائه می‌شود که از ارتباط درون‌زبانی و میان‌زبانی ویژگی‌ها برای ایجاد واژه‌نامه در زبان مقصد استفاده می‌شود.

در [۲۳] از پیکره موازی برای انتقال اطلاعات از یک زبان به زبان دیگر استفاده می‌شود. در این روش با استفاده از هم‌ترازی در سطح کلمات، حاشیه‌نویسی‌ها از زبان مبدأ به زبان مقصد منتقل می‌شود. رویکرد دیگری که در این حوزه مورد استفاده قرار می‌گیرد، استخراج اطلاعاتی است که به طور واضح و آشکار در داده‌ها قابل دریافت نمی‌باشد. تخصیص دیریشله نهفته[۱۳] از جمله روش‌هایی است که سعی در استخراج این‌گونه اطلاعات دارد. در [۲۴] با استفاده از این روش به همراه ماشین ترجمه، ارتباط میان ویژگی‌های دو زبان استخراج می‌شود و تحلیل نظرات بر روی جنبه‌های نظرات انجام می‌گیرد. در [۲۵] با استفاده از پیکره موازی هم‌ترازشده در سطح جمله، نمایش طیفی عبارت‌ها استخراج می‌شود و از این اطلاعات برای پیش‌بینی جهت معنایی نظرات بهره برده می‌شود.

روش ارائه شده در [۲۶، ۲۷]، حالت بین‌زبانی روشی است که ابتدا در حوزه بین‌دامنه‌ای مطرح شده بود [۲۸]. با ایجاد تغییرات، تطبیق‌ها و پیشنهاداتی که ارائه شد چالش‌های استفاده در حوزه بین‌زبانی برطرف شدند. مهم‌ترین تغییری که انجام گرفت، به استخراج معادل ویژگی‌های زبان مبدأ در زبان مقصد می‌توان اشاره کرد.

۲-۲- حوزه بین‌دامنه‌ای

در روش‌های ارائه شده در حوزه بین‌دامنه‌ای برای تشخیص و کاهش اختلاف موجود بین دو دامنه متفاوت تلاش می‌شود. روش پیشنهاد شده در [۲۸] علاوه بر نظرات برچسب‌خورده در زبان مبدأ از نظرات بدون برچسب در هر دو زبان نیز بهره می‌برد. این‌گونه نظرات را که می‌توان در مقیاس بزرگ‌تری نسبت به نظرات برچسب‌خورده جمع‌آوری کرد، علی‌رغم نداشتن برچسب، حاوی اطلاعات مفیدی است. در این مقاله از این نظرات برای استخراج کوواریانس دو دسته از ویژگی‌ها استفاده می‌شود. یک دسته از ویژگی‌ها که ویژگی‌های محوری نامیده می‌شود، ویژگی‌هایی جهت‌داری است که به مقدار خوبی نشان‌دهنده برچسب نظرات است. این ویژگی‌ها با استفاده از اطلاعات متقابل بین ویژگی و برچسب، استخراج می‌شود و سایر ویژگی‌ها ویژگی‌های غیرمحوری نامیده می‌شود. با محاسبه کوواریانس بین این دو دسته از ویژگی‌ها و کاهش ابعاد آن رده‌بندی نظرات انجام می‌شود.

در [۲۹] نیز از ایده به‌کارگیری نظرات بدون برچسب استفاده می‌شود. مشابه روش پیشنهاد شده در [۲۸] ویژگی‌ها به دو دسته تقسیم می‌شود، اما از تعریف متفاوتی استفاده شده و سعی شده ویژگی‌های وابسته به دامنه را از ویژگی‌های مستقل از دامنه جدا شود. ایده دیگری که مطرح می‌شود ایجاد یک گراف بین این دو دسته از ویژگی است. سپس با استفاده از یکی از روش‌های خوشه‌بندی گراف، ویژگی‌ها خوشه‌بندی می‌شود و با استفاده از نتیجه این خوشه‌بندی و نظرات برچسب‌خورده موجود در دامنه دیگر، برچسب‌گذاری نظرات در دامنه مورد نظر انجام می‌شود.

در [۳۰] نیز رویکرد مشابهی در پیش گرفته می‌شود. با استفاده از الگوریتم بهینه‌سازی پیشنهادی به استخراج کلمات مشترک و کلمات خاص هر دامنه پرداخته می‌شود.

روش پیشنهاد شده در این مقاله در حوزه کارهای بین‌زبانی قرار می‌گیرد و با استفاده از نظرات بدون برچسب در دو زبان مبدأ و مقصد، هم‌زمان به استخراج اطلاعات از زبان مبدأ و انتقال آن‌ها به زبان مقصد می‌پردازد. این روش با استفاده از یک لغت‌نامه و ترجمه تعداد کمی از کلمات می‌تواند این استخراج و انتقال را با کیفیت خوبی انجام دهد. مشابه ایده پیشنهاد شده در [۲۹] گرافی بین دو دسته از ویژگی‌ها ایجاد می‌شود و ویژگی‌های مشترکی برای کلمات کاملاً متفاوت دو زبان استخراج می‌شود. با ارائه ایده‌هایی که در ادامه به آن‌ها پرداخته می‌شود، برای چالش‌ها و محدودیت‌های موجود در این پژوهش راه‌حلی پیشنهاد شده است.

۳- نظرکاوی بین‌زبانی با استفاده از ویژگی‌های معنایی

نظرات موجود در سایت‌ها و صفحات شخصی، هم در دامنه‌های متفاوتی قرار می‌گیرد و هم دارای زبان‌های متفاوتی است. برای تحلیل نظرات به صورت باناظر، بهترین حالت استفاده از نظرات برچسب‌خورده در دامنه و زبان مشابه همان نظرات است. اما به دلائل گوناگونی که قبلاً نیز ذکر شد، امکان دارد این نظرات برچسب‌خورده در دامنه و زبان مورد نظر موجود نباشد، درحالی‌که در دامنه متفاوت و یا زبان متفاوتی، چنین نظراتی یافت می‌شود. از طرفی تولید نظرات برچسب‌خورده پرهزینه و زمان‌گیر است و از طرف دیگر، استفاده مستقیم از نظرات در دامنه و یا زبان دیگر، کارایی مطلوبی ندارد. بنابراین روش‌هایی برای قابل

استفاده شدن نظرات در دامنه و یا زبان دیگر، ارائه می‌شود تا چنین محدودیتی برای نظرات وجود نداشته باشد. روش‌های بین‌دامنه‌ای از نظرات برچسب‌خورده در دامنه متفاوت و زبان یکسان و روش‌های بین‌زبانی از نظرات برچسب‌خورده در دامنه یکسان و زبان متفاوت استفاده می‌کند.

روش‌های بین‌دامنه‌ای و بین‌زبانی علاوه بر تفاوت‌ها و رویکرد متفاوتی که با هم دارند، دارای اشتراکاتی نیز هستند. هر دو دسته از این روش‌ها از داده‌هایی استفاده می‌کنند که نسبت به داده‌های مورد نظرشان از توزیع متفاوتی برخوردار است و سعی می‌کنند مسأله تفاوت در توزیع دو دسته از داده را به گونه‌ای برطرف سازند. این نکته، این امکان را به ما می‌دهد که از روش‌های بین‌دامنه‌ای بتوانیم در حوزه بین‌زبانی نیز استفاده کنیم. اما روش‌های بین‌دامنه‌ای مزیتی نسبت به روش‌های بین‌زبانی دارد و آن یکسان بودن کلمات استفاده شده در دو توزیع است. درحالی‌که در روش‌های بین‌زبانی دو توزیع متفاوت از کلمات کاملا متفاوت استفاده می‌کند که برای به کار بردن روش‌های بین‌دامنه‌ای در حوزه بین‌زبانی، باید این مسأله در نظر گرفته و راه‌حلی برای رفع آن ارائه شود.

با توجه به نکات بیان‌شده، روش‌های بین‌دامنه‌ای به صورت مستقیم در حوزه بین‌زبانی قابل استفاده نیست. در این مقاله سعی شده روش‌هایی برای برطرف ساختن چالش‌های پیش‌رو برای کمک گرفتن از روش‌های بین‌دامنه‌ای پیشنهاد شود. روش پیشنهادی این مقاله، از یک روش بین‌دامنه‌ای موجود [۲۹] ایده گرفته است. این روش بین‌دامنه‌ای که هم‌ترازی ویژگی‌های طیفی[۱۴] نام دارد، راه‌حلی برای استفاده از نظرات برچسب‌خورده در یک دامنه مبدأ برای برچسب زدن نظرات در دامنه مقصد، به کمک نظرات بدون برچسب در هر دو دامنه، ارائه داده است. چگونگی تشخیص تمایز بین ویژگی‌ها در هر دو زبان و ایجاد گراف دوبخشی که از بخش‌های اصلی این روش به شمار می‌روند، چالش‌هایی بود که در این پژوهش بررسی و روشی برای برطرف ساختن آن‌ها پیشنهاد شد.

در ادامه ابتدا ایده روش هم‌ترازی ویژگی‌های طیفی به صورت مختصر توضیح داده می‌شود و بعد از آن ایده مطرح شده در این مقاله به صورت دقیق‌تر بررسی می‌شود.

۳-۱- روش هم‌ترازی ویژگی‌های طیفی

روش هم‌ترازی ویژگی‌های طیفی برای مسأله تحلیل نظرات بین‌دامنه‌ای مطرح شد. حوزه تحلیل نظرات بین‌دامنه‌ای از نظرات دارای برچسب در دامنه متفاوت و زبان یکسان نسبت به نظرات مورد نظر استفاده می‌کند و هدف کاهش فاصله بین دو دامنه مبدأ و مقصد است که این فاصله، توزیع متفاوت داده‌ها در دو دامنه مختلف است. برای مثال، در دامنه کتاب، کلماتی مثل طولانی، هیجان‌انگیز، یکنواخت، ... برای توصیف یک کتاب استفاده می‌شود، درحالی‌که در دامنه دیگری مانند موبایل این کلمات استفاده چندانی ندارد و در عوض کلماتی مثل طراحی، کیفیت، دوربین، ... برای بیان نظرات بیش‌تر به کار می‌رود که این کلمات هم در دامنه کتاب بسیار کم‌تر مشاهده می‌شود. این روش علاوه بر بهره‌بردن از نظرات دارای برچسب، از نظرات بدون برچسب نیز استفاده می‌کند. این نظرات را به دلیل عدم نیاز به برچسب‌گذاری می‌توان در مقیاس بزرگ‌تری جمع‌آوری کرد. در این روش سعی شده با استفاده از این نظرات، اطلاعات بیش‌تری از دامنه‌ها استخراج شود و از این اطلاعات برای شناسایی برچسب نظرات بهره برده شود.

در این روش ابتدا از بین ویژگی‌های موجود، ویژگی‌های مستقل از دامنه[۱۵] استخراج می‌شود (بخش ۳-۱-۱) و سپس بین این ویژگی‌ها و سایر ویژگی‌ها که ویژگی‌های وابسته به دامنه[۱۶] نامیده می‌شود، گراف دوبخشی تشکیل داده می‌شود (بخش ۳-۱-۲). سپس این گراف دوبخشی با استفاده از روشی که در بخش ۳-۱-۳ به طور کامل توضیح داده می‌شود، به صورت نرم خوشه‌بندی می‌شود که ویژگی‌ها با وزنی به هر خوشه اختصاص داده می‌شود. وزن هر ویژگی در هر خوشه، ویژگی طیفی نامیده می‌شود. با استخراج ویژگی‌های طیفی برای هر نظر و استفاده از یک رده‌بند، با توجه به دوزبانه بودن خوشه‌ها، برچسب نظرات در هر دو زبان را می‌توان پیش‌بینی کرد. در ادامه، هر بخش از این روش به طور دقیق‌تری توضیح داده می‌شود.

۳-۱-۱- انتخاب ویژگی‌های مستقل از دامنه

انتخاب ویژگی‌های مستقل از دامنه اصلی‌ترین و مهم‌ترین قسمت این روش است. این روش وابستگی بالایی به ویژگی‌های مستقل از دامنه انتخاب شده دارد، هر چه روش انتخابی بهتر باشد و ویژگی‌های مناسب‌تری انتخاب شود، فاصله میان دو دامنه به میزان بیش‌تری کاهش می‌یابد. این ویژگی‌ها، ویژگی‌هایی است که در هر دو دامنه مورد نظر پرتکرار و دارای رفتار مشابه است.

با توجه به خصوصیات ذکر شده برای ویژگی‌های مستقل از دامنه، ایده‌ای که در این روش مطرح شد، استفاده از اطلاعات متقابل[۱۷] بین ویژگی‌ها و دامنه‌ها برای استخراج ویژگی‌هایی با رفتار یکسان در دو دامنه است. هر چه این مقدار کم‌تر باشد، نشان‌دهنده استقلال ویژگی از دامنه است. در نتیجه پس از محاسبه اطلاعات متقابل برای تمام ویژگی‌ها، ویژگی‌هایی که کم‌ترین مقدار را دارد، به عنوان ویژگی‌های کاندیدا انتخاب می‌شود.

$$I(X;D)=\sum_{\substack{d\in D\\ D=\{s,t\}}}\sum_{\substack{x\in X\\ X=\{0,1\}}} p(x,d)\log_2\frac{p(x,d)}{p(x)p(d)} \qquad (۱)$$

در این‌جا، X دربردارنده وجود و عدم وجود ویژگی مورد نظر و D دربردارنده دامنه مبدأ و دامنه مقصد است. در نتیجه عبارت بالا برای چهار حالت مختلف محاسبه می‌شود.

پس از مشخص شدن ویژگی‌های کاندیدا، ویژگی‌هایی به عنوان ویژگی‌های مستقل از دامنه انتخاب می‌شود که تعداد رخدادشان از مقدار کمینه تعیین شده بیش‌تر باشد. در این صورت ویژگی‌های مستقل از دامنه، کلمات پراستفاده‌ای است که در هر دو دامنه رفتار مشابهی دارد. در نتیجه این ویژگی‌ها هم‌رخدادی زیادی با سایر ویژگی‌ها دارد و استخراج ویژگی‌های طیفی با دقت خوبی انجام می‌گیرد.

۳-۱-۲- ساخت گراف دوبخشی

در این مرحله، گراف دوبخشی ساخته می‌شود. پس از انتخاب ویژگی‌های مستقل از دامنه، سایر ویژگی‌ها به عنوان ویژگی‌های وابسته به دامنه انتخاب می‌شود. برای تحلیل نظرات بین‌دامنه‌ای و استفاده از برچسب نظرات در یک دامنه برای برچسب‌گذاری نظرات در دامنه دیگر نیاز به وجود نظرات در هر دو دامنه و خوشه‌بندی تمام ویژگی‌ها است. در نتیجه با استفاده از نظرات بدون برچسب در هر دو دامنه، یک گراف دوبخشی ایجاد می‌شود. یال‌های این گراف بین ویژگی‌های مستقل از دامنه و ویژگی‌های وابسته به دامنه است و دارای وزنی به نسبت تعداد هم‌رخدادی دو ویژگی مذکور در نظرات بدون برچسب است.

به عبارت دیگر، این گراف دوبخشی با استفاده از نظرات موجود در هر دو دامنه مبدأ و مقصد ساخته می‌شود. در یک بخش از این گراف ویژگی‌های مستقل از دامنه f_{DI} و در بخش دیگر آن ویژگی‌های وابسته به دامنه f_{DS} قرار دارد. با توجه به تعریف مسأله بین‌دامنه‌ای، تمام این ویژگی‌ها در یک زبان مشترک است و ویژگی‌های وابسته به دامنه در دو دامنه با توزیع متفاوتی دیده شده است. وزن یال بین این دو دسته از ویژگی‌ها به صورت زیر محاسبه می‌شود:

$$w_{ij} = c(S_s, f_{DI_i} \cup f_{DS_j}) + c(S_t, f_{DI_i} \cup f_{DS_j}) \quad (۲)$$

که در این‌جا، $c(S_s, f_{DI_i} \cup f_{DS_j})$ تعداد نظراتی در دامنه مبدأ S_s است که هم iامین ویژگی مستقل از دامنه f_{DI_i} و هم jامین ویژگی وابسته به دامنه f_{DS_j} در آن مشاهده شده باشد. و $c(S_t, f_{DI_i} \cup f_{DS_j})$ تعداد نظرات در دامنه مقصد S_t است که هر دو ویژگی f_{DI_i} و f_{DS_j} در آن رخ داده باشد.

۳-۱-۳- استخراج ویژگی‌های طیفی

همان‌طور که در بخش‌های قبلی توضیح داده شد، با تعریف دو محدودیت برای ویژگی‌های مستقل از دامنه، کلماتی که محدودیت‌ها را برآورده می‌کند کاندیدا برای این دسته از ویژگی‌ها است. اگر فرض کنیم تعداد ویژگی‌های مستقل از دامنه p باشد، p ویژگی پرتکرار با کمترین مقدار اطلاعات متقابل بین ویژگی و دامنه، به عنوان ویژگی‌های مستقل از دامنه انتخاب می‌شود و پس از آن گراف دوبخشی بین ویژگی‌های مستقل از دامنه و ویژگی‌های وابسته به دامنه ساخته می‌شود. از آن‌جایی که در این مراحل از برچسب نظرات استفاده‌ای نمی‌شود، انتخاب ویژگی‌های مستقل از دامنه و ساخت گراف دوبخشی با استفاده از نظرات بدون برچسب صورت می‌گیرد.

اکنون به بررسی روند استخراج ویژگی‌های طیفی پرداخته می‌شود. پس از ساخت گراف دوبخشی بین ویژگی‌های مستقل از دامنه و ویژگی‌های وابسته به دامنه، به سادگی می‌توان ماتریس یال‌های گراف حاصل را تشکیل داد. اگر تعداد کل ویژگی‌ها n و تعداد ویژگی‌های مستقل از دامنه p باشد، ماتریس یال‌های گراف، که M خوانده می‌شود، دارای ابعاد $p \times (n-p)$ خواهد بود. هم‌چنین ماتریس مجاورت این گراف که یک ماتریس مربعی است، ماتریسی به ابعاد $n \times n$ خواهد بود که:

$$A = \begin{bmatrix} 0 & M \\ M^T & 0 \end{bmatrix} \quad (۳)$$

این ماتریس مجاورت A، حاوی وزن خام یال‌ها است. ایده مطرح شده در این الگوریتم، استخراج ویژگی‌های طیفی از ماتریس لاپلاس گراف، به جای ماتریس مجاورت گراف است. فرمول (۴) روش محاسبه ماتریس لاپلاس L را نشان می‌دهد. با محاسبه لاپلاس ماتریس مجاورت، وزن‌های خام به وزن‌های نرمال شده تبدیل می‌شود که این وزن‌های نرمال شده به صورت بهتری نشان‌دهنده میزان اهمیت یال‌ها است.

$$L = D^{-\frac{1}{2}} \times A \times D^{-\frac{1}{2}} \quad (۴)$$

که D در فرمول بالا، ماتریس درجه گراف و یک ماتریس قطری است. درایه‌های قطری ماتریس، درجه رأس متناظر آن درایه است و درایه‌های غیرقطری آن برابر با صفر است.

$$d_{ii} = \sum_j a_{ij} \quad (۵)$$

بعد از محاسبه ماتریس لاپلاس گراف دوبخشی، ویژگی‌های طیفی استخراج می‌شوند که برای استخراج آن‌ها از تجزیه مقدار منفرد[۱۸] استفاده شده است که یکی از روش‌های خوشه‌بندی نرم[۱۹] محسوب می‌شود.

$$U \Sigma V^T = SVD(L) \quad (۶)$$

تجزیه مقدار منفرد با ایجاد تعداد زیادی خوشه، وزن هر ویژگی در هر خوشه را در ماتریس U ذخیره می‌کند و همچنین ماتریس Σ یک ماتریس قطری است که دربردارنده میزان قوت هر خوشه است.

اگر ویژگی‌های مستقل از دامنه به درستی انتخاب شده باشد و خصلت استقلال از دامنه را داشته باشد، ویژگی‌های وابسته به دامنه نیز به درستی توسط تجزیه مقدار منفرد، به صورت وزن‌دار، خوشه‌بندی می‌شود. در نتیجه برای رده‌بندی می‌توان تنها از ویژگی‌های طیفی استخراج شده ویژگی‌های وابسته به دامنه استفاده کرد. به عبارتی دیگر ویژگی‌های طیفی همان وزن ویژگی‌ها در خوشه‌های ایجاد شده است. با توجه به میزان قوت خوشه‌ها، تنها k تا از بهترین خوشه‌ها برای تشخیص وزن هر نظر در هر خوشه انتخاب می‌شود. این خوشه‌ها که میزان درستی ویژگی‌های طیفی آن‌ها بالاتر است، برای تحلیل نظرات کافی است.

$$\theta = U_{[p:n,1:k]} \in \mathbb{R}^{(n-p)\times k} \quad (۷)$$

با استفاده از θ که حاوی ویژگی‌های طیفی است و نسبت به ویژگی‌های مستقل از دامنه و وابسته به دامنه تعداد بسیار کم‌تری دارد، می‌توان فضای نظرات را به فضای طیفی منتقل کرد و نظرات را در فضای جدید رده‌بندی کرد.

اگر s یک نظر در فضای ویژگی‌های وابسته به دامنه باشد که دارای ابعاد $1 \times (n-p)$ است، s' همان نظر در فضای ویژگی‌های طیفی است و ابعاد آن $1 \times k$ خواهد بود که با توجه به ویژگی‌های استفاده‌شده وزنی در هر خوشه دارد.

$$s' = s \times \theta \quad (۸)$$

در نتیجه، به جای مدل کردن کلمات نظرات، ویژگی‌های طیفی نظرات را استخراج کرده و با استفاده از آن‌ها رده‌بندی صورت می‌گیرد. همان‌طور که قبلاً بیان شد، تعداد ویژگی‌های طیفی نسبت به ویژگی‌ها مستقل از دامنه و وابسته به دامنه، بسیار کم‌تر است و در نتیجه از میزان تنک[۲۰] بودن فضای مسأله بسیار کاسته می‌شود و می‌توان انتظار داشت که رده‌بندی با دقت بالاتری حاصل شود.

۳-۲- روش پیشنهادی

مسأله مورد بررسی در این مقاله مسأله تحلیل نظرات بین‌زبانی است. این حوزه به بررسی روش‌هایی می‌پردازد که می‌خواهد از داده‌های موجود در زبان دیگری برای رده‌بندی داده‌ها در زبان مورد نظرشان استفاده کند. این داده‌ها که معمولاً در یک دامنه است، ویژگی‌های کاملاً متفاوتی با یکدیگر دارد. به عبارت دیگر دو زبان مختلف به علت استفاده از کلمات متفاوت، دارای ویژگی‌های مختلفی است. اما با استفاده از منابع ترجمه، مانند لغت‌نامه، ماشین ترجمه و یا پیکره‌های دوزبانه می‌توان ویژگی‌های دو زبان را به هم نگاشت کرد. در عین حال باید به این نکته توجه داشت که منابع ترجمه حاوی ترجمه‌های مبهم و گاهی نادرست است.

در روش پیشنهادی از روش هم‌ترازی ویژگی‌های طیفی [۲۹] برای حل مسأله تحلیل نظرات بین‌زبانی کمک گرفته شده است. اما به دلیل عدم اشتراک بین ویژگی‌های دو زبان، تعریف و روش محاسبه‌ای که برای ویژگی‌های مستقل از دامنه ارائه شده در حوزه تحلیل نظرات بین‌زبانی قابل استفاده نیست. بنابراین برای محاسبه و شناسایی یک دسته از ویژگی‌ها که نقشی مشابه نقش ویژگی‌های مستقل از دامنه داشته باشد، به معرفی تعریف و روش جدیدی نیاز است. در این مقاله این ویژگی‌ها، ویژگی‌های محوری[۲۱] نامیده شد. ویژگی‌های محوری در این‌جا

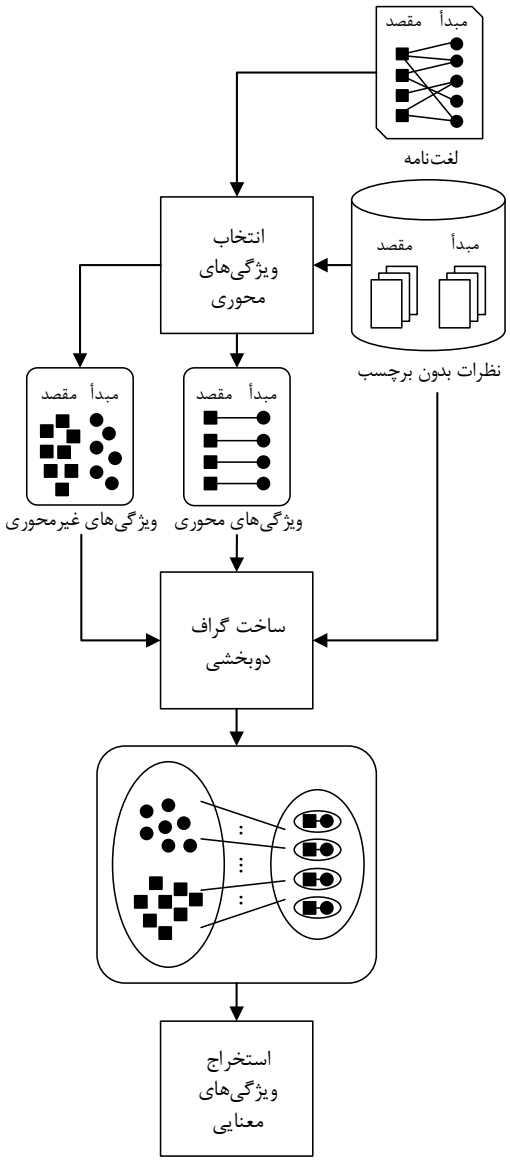
جفت‌کلمه‌هایی است که به عنوان ویژگی‌های مستقل از زبان تعریف شده است و استقلال از زبان، عدم وجود ابهام در ترجمه ویژگی در نظر گرفته شده است. برای شناسایی چنین ویژگی‌هایی احتمال استقلال از زبان برای تمام جفت‌کلمه‌های موجود در منبع ترجمه محاسبه می‌شود. جفت‌کلمه‌هایی که ترجمه‌شان مبهم در نظر گرفته می‌شود جزء دسته دوم یعنی ویژگی‌های غیرمحوری قرار می‌گیرد. از آن‌جایی که وجود ابهام در ترجمه نشان‌دهنده نامناسب بودن این هم‌ترازی‌ها است، راهکاری برای این مسأله نیز ارائه شد. با حذف هم‌ترازی‌های ویژگی‌های غیرمحوری و به عبارتی ترجمه نکردن این ویژگی‌ها، ویژگی‌های غیرمحوری تبدیل به تک‌کلمه‌هایی شد که شامل ویژگی‌هایی از زبان مبدأ و مقصد می‌باشد که در ویژگی‌های محوری جای نگرفته است.

به طور خلاصه، در روش پیشنهادی تعدادی از جفت‌کلمه‌های موجود در منبع ترجمه به عنوان ویژگی‌های محوری انتخاب می‌شود و سایر ویژگی‌های دو زبان به صورت تک‌کلمه در دسته ویژگی‌های غیرمحوری قرار می‌گیرد. ارتباط میان این دو دسته از ویژگی‌ها به کمک ایجاد یک گراف دوبخشی جمع‌آوری می‌شود. سپس از این ارتباطات به دست آمده، ویژگی‌های معنایی استخراج می‌شود و در نهایت نظرات رده‌بندی می‌شود. در این روش علاوه بر نظرات برچسب‌خورده در زبان مبدأ، از نظرات بدون برچسب در هر دو زبان نیز استفاده شده است. این نظرات که به صرف هزینه و زمان برای تعیین برچسب نیاز ندارد، برای جمع‌آوری به وقت و هزینه کم‌تری نسبت به داده‌های آموزش و آزمون احتیاج دارد و در عین حال به دلیل موجود بودن در هر دو زبان به کسب اطلاعات لازم برای تحلیل نظرات بین‌زبانی کمک می‌کند. در شکل ۱ فرآیند استخراج ویژگی‌های معنایی در روش پیشنهادی قابل مشاهده است.

۳-۲-۱- انتخاب ویژگی‌های محوری

ویژگی‌های محوری نقش مشابهی با ویژگی‌های مستقل از دامنه در روش هم‌ترازی ویژگی‌های طیفی دارد. همان‌طور که ویژگی‌های مستقل از دامنه برای انتقال اطلاعات از یک دامنه به دامنه دیگر مورد استفاده قرار می‌گرفت، ویژگی‌های محوری نیز وظیفه انتقال اطلاعات از یک زبان به زبان دیگری را دارد. ویژگی‌های محوری تنها ویژگی‌هایی است که با استفاده از منبع ترجمه، به زبان دیگر ترجمه و به عبارتی با ویژگی‌ای در آن زبان هم‌تراز می‌شود. بنابراین می‌توان ادعا کرد که از این ویژگی‌ها به عنوان پل ارتباطی بین دو زبان استفاده می‌شود. اگر کلمات مناسبی به عنوان ویژگی‌های محوری انتخاب نشود، این ارتباط میان دو زبان به درستی ساخته نمی‌شود و در نتیجه اطلاعات موجود در یک زبان به زبان دیگر به خوبی منتقل نمی‌شود.

برای شناسایی ویژگی‌های محوری، لازم به تعریف محدودیت‌هایی است که نشان‌دهنده خصلت استقلال از زبان باشد و کلماتی که این محدودیت‌ها را برآورده می‌کند، پل ارتباطی خوبی بین دو زبان برقرار کند. به همین منظور پرتکرار و رایج بودن ویژگی به عنوان یک محدودیت برای ویژگی‌های محوری تعریف شد. هر چه یک ویژگی در نظرات بیش‌تری استفاده شده باشد، می‌توان فرض کرد که با تعداد ویژگی‌های متمایز بیش‌تری هم‌رخدادی و ارتباط دارد. دومین محدودیت، مستقل بودن ویژگی از زبان است که از تعاریف اصلی ویژگی‌های محوری به شمار می‌رود. ایده‌ای که در این‌جا برای شناسایی چنین ویژگی‌هایی مطرح می‌شود، ایجاد ارتباط میان ویژگی‌های دو زبان است. وجود چنین ارتباطی میان دو ویژگی از دو زبان متفاوت، نشان‌گر هم‌تراز بودن این ویژگی‌ها در دو زبان است. از منابعی که حاوی ارتباطاتی میان ویژگی‌های دو زبان متفاوت باشد، می‌توان به منابع ترجمه اشاره کرد. در منابع ترجمه، این ارتباطات به صورت ترجمه یک ویژگی در زبان مبدأ به یک یا چند ویژگی در زبان مقصد تعریف می‌شود.



شکل ۱- فرآیند استخراج ویژگی‌های معنایی

منابع ترجمه متفاوتی در زبان‌های مختلف وجود دارد. لغت‌نامه ساده‌ترین و ماشین ترجمه جزء پیچیده‌ترین منابع ترجمه به شمار می‌رود. روش پیشنهادی که مستقل از نوع منبع ترجمه مورد استفاده است، تنها به یک نگاشت کلمات در زبان مبدأ به کلمات در زبان مقصد احتیاج دارد. در نتیجه لغت‌نامه که در اکثر جفت‌زبان‌ها، حتی زبان‌های با منابع محدود موجود است، مورد استفاده قرار گرفته است.

با استفاده از منابع ترجمه، جفت‌کلمه‌هایی مانند (x_s, x_t) به دست می‌آید. اولی کلمه‌ای در زبان مبدأ و دومی کلمه‌ای در زبان مقصد است که یک هم‌ترازی بین این دو کلمه ایجاد شده است. ویژگی‌های محوری مجموعه‌ای از همین جفت‌کلمه‌ها است. این نکته حائز اهمیت است که در منابع ترجمه برای یک کلمه ممکن است چند ترجمه وجود داشته باشد و یا یک کلمه ترجمه چند کلمه باشد. همچنین این ترجمه‌ها ممکن است دارای خطا باشند و یا برخی از هم‌ترازی‌ها تنها در موارد خاصی برقرار باشد. در روش پیشنهاد شده برای انتخاب ویژگی‌های محوری، سعی شده این مسائل در نظر گرفته شود و بهترین و بدون‌ابهام‌ترین ترجمه‌ها انتخاب شود.

برای انتخاب ویژگی‌های محوری، محدودیت‌های تعریف شده بر روی جفت‌کلمه‌های موجود اعمال می‌شود. جفت‌کلمه‌هایی محدودیت اول را برآورده

می‌کند که هر دو کلمه در زبان خود پرتکرار باشد در نتیجه کلماتی که از حد آستانه تعریف شده، کم‌تر ظاهر شده باشد، به همراه زوجشان، از لیست کاندیداهای ویژگی‌های محوری حذف می‌شود. محدودیت دوم استقلال ویژگی از زبان است. استقلال ویژگی از زبان تشابه رفتار جفت‌کلمه یعنی رخ دادن و رخ ندادن دو کلمه در نظرات متناظرشان تعریف شده است. ویژگی‌هایی که دو کلمه آن در زبان متناظرشان، رفتار مشابهی با یکدیگر داشته باشد این محدودیت را برآورده می‌کند. این رفتار مشابه و عدم وابستگی بین ویژگی و زبان را می‌توان با استفاده از فرمول اطلاعات متقابل محاسبه کرد. این فرمول در ذیل نشان داده شده است [۳۱].

$$\begin{aligned} I(X;L) &= \sum_{e_x \in \{0,1\}} \sum_{e_l \in \{s,t\}} p(e_x, e_l) \log \frac{p(e_x, e_l)}{p(e_x)p(e_l)} \\ &= \frac{N_{11}}{N} \log \frac{N N_{11}}{N_{1.} N_{.1}} + \frac{N_{10}}{N} \log \frac{N N_{10}}{N_{1.} N_{.0}} \\ &+ \frac{N_{01}}{N} \log \frac{N N_{01}}{N_{0.} N_{.1}} + \frac{N_{00}}{N} \log \frac{N N_{00}}{N_{0.} N_{.0}} \end{aligned} \quad (۹)$$

در فرمول بالا، X ویژگی و یا همان جفت‌کلمه مورد نظر و L زبان است که اگر e_x برابر با یک باشد، احتمال وجود و اگر برابر با صفر باشد، احتمال عدم وجود ویژگی مد نظر است که برای هر زبان، کلمه متناظرش در جفت‌کلمه، برای محاسبه احتمال در نظر گرفته می‌شود. به صورت دقیق‌تر، N_{11} تعداد نظرات حاوی x_s در زبان مبدأ، N_{10} تعداد نظرات حاوی x_t در زبان مقصد، N_{01} تعداد نظرات بدون x_s در زبان مبدأ و N_{00} تعداد نظرات بدون x_t در زبان مقصد است. $N_{1.}$ مجموع N_{11} و N_{10}، $N_{0.}$ مجموع N_{01} و N_{00}، $N_{.1}$ مجموع N_{11} و N_{01} و $N_{.0}$ مجموع N_{10} و N_{00} است. N نیز تعداد کل نظرات در زبان مبدأ و زبان مقصد است.

این فرمول میزان وابستگی دو متغیر ویژگی و زبان را محاسبه می‌کند، در نتیجه هر چه مقدار این احتمال کم‌تر باشد دو متغیر وابستگی کم‌تر و استقلال بیش‌تری نسبت به هم دارند. بنابراین کلماتی به عنوان ویژگی‌های محوری انتخاب می‌شود که کم‌ترین مقادیر اطلاعات متقابل را داشته باشد. در این صورت می‌توان فرض کرد این کلمات علاوه بر استقلال از زبان، دارای کیفیت بالای ترجمه و بدون ابهام در دامنه مورد نظر نیز است.

در صورتی که یک کلمه در زبان مبدأ دارای چند ترجمه مختلف در زبان مقصد باشد، با هر کدام از این ترجمه‌ها، یک جفت‌کلمه تشکیل می‌دهد که هر کدام از این جفت‌کلمه‌ها مقدار اطلاعات متقابل متفاوتی خواهند داشت. جفت‌کلمه‌ای دارای کم‌ترین مقدار است که رفتار آن ترجمه در زبان مقصد به رفتار کلمه در زبان مبدأ نزدیک‌تر باشد. به عبارت دیگر آن ترجمه در دامنه و زبان مورد بررسی، بهترین معادل برای کلمه مورد نظر می‌باشد. در این صورت جفت‌کلمه‌هایی که به عنوان ویژگی‌های محوری انتخاب می‌شوند را می‌توان یک مفهوم غیرمبهم در هر دو زبان در نظر گرفت.

همان‌طور که مشاهده می‌شود، در فرمول (۹) برای محاسبه اطلاعات متقابل، از برچسب نظرات استفاده‌ای نمی‌شود، در نتیجه می‌توان برای انتخاب ویژگی‌های محوری از نظرات بدون برچسب که تعداد بیش‌تری از آن‌ها در اختیار است، استفاده کرد.

۳-۲-۲- ساخت گراف دوبخشی

اکنون ویژگی‌های محوری انتخاب شده است. این ویژگی‌ها رأس‌های یک بخش از گراف دوبخشی را تشکیل می‌دهد و بخش دیگر این گراف از ویژگی‌های غیرمحوری تشکیل می‌شود. همان‌طور که گفته شد، ویژگی‌های محوری به صورت جفت‌کلمه (x_s, x_t) است و به عبارت دیگر یک ویژگی محوری از یک کلمه در زبان مبدأ و یک کلمه در زبان مقصد (که ترجمه کلمه اول است) تشکیل شده است. اما در مقابل ویژگی‌های غیرمحوری به صورت تک‌تک و جدا است، یعنی یک ویژگی غیرمحوری یا یک کلمه در زبان مبدأ y_s و یا یک کلمه در زبان مقصد y_t است که این ویژگی‌ها به ترجمه و هم‌ترازی با ویژگی دیگر در زبان دیگر نیازی ندارد. این خصوصیت موجب بالا رفتن کیفیت روش نسبت به سایر روش‌هایی است که نیاز به ترجمه تک‌تک کلمات دارد. در نتیجه منبع ترجمه‌ای که بتواند کلمه‌ها را ترجمه کند برای این روش کافی است و نیازی به ترجمه جمله در این‌جا وجود ندارد.

پس از مشخص شدن رأس‌های گراف دوبخشی، یال‌های گراف ایجاد می‌شود. گراف دوبخشی، یک گراف وزن‌دار غیرمنفی است که وزن یال‌های آن متناسب با تعداد هم‌رخدادی دو رأس آن در نظرات زبان مبدأ و زبان مقصد است. هم‌رخدادی در این‌جا، وقوع یک کلمه در همسایگی کلمه دیگر است که این همسایگی می‌تواند به صورت یک پنجره اطراف یک کلمه تعریف شود. اندازه این پنجره می‌تواند از ۱ (دو کلمه‌ای هم‌رخدادی دارند که در مجاورت هم در متن ظاهر شده باشند) تا طول سند (هر دو کلمه‌ای که در یک سند ظاهر شده‌اند، هم‌رخدادی دارند) متغیر باشد. در این‌جا اندازه پنجره، طول سند در نظر گرفته شده است. نظری که حاوی یک ویژگی محوری است، وزن یال متناظرش با ویژگی‌های غیرمحوری موجود در همان نظر را یک واحد افزایش می دهد.

بدیهی است که میان ویژگی‌های محوری و همچنین میان ویژگی‌های غیرمحوری یالی وجود ندارد و یال‌ها تنها میان این دو دسته از ویژگی‌ها برقرار می‌شود. طبق توضیحات داده شد، iامین ویژگی محوری را می‌توان مطابق فرمول (۱۰) نشان داد. وزن یال میان iامین ویژگی محوری و jامین ویژگی غیرمحوری، بسته به این‌که ویژگی غیرمحوری مورد نظر متعلق به کدام زبان باشد، به کمک یکی از دو فرمول (۱۱) و (۱۲) محاسبه می‌شود. اگر ویژگی غیرمحوری متعلق به زبان مبدأ باشد، از فرمول (۱۱) و در غیر این صورت از فرمول (۱۲) استفاده می‌شود.

$$f_{p_i} = (x_{s_i}, x_{t_i}) \quad (۱۰)$$

$$w_{ij} = c(x_{s_i} \cup f_{n_j}, S_s) \quad (۱۱)$$

$$w_{ij} = c(x_{t_i} \cup f_{n_j}, S_t) \quad (۱۲)$$

که $c(x_{s_i} \cup f_{n_j}, S_s)$ تعداد نظراتی در زبان مبدأ است که دارای هر دو کلمه x_{s_i} و f_{n_j} باشد. x_{s_i} کلمه متعلق به زبان مبدأ در ویژگی محوری iام و f_{n_j} ویژگی محوری jام است و $c(x_{t_i} \cup f_{n_j}, S_t)$ تعداد نظرات در زبان مقصد، دارای دو کلمه x_{t_i} و f_{n_j} است.

در این مرحله نیز به علت عدم به‌کارگیری برچسب نظرات، می‌توان از نظرات بدون برچسب استفاده کرد. نظرات بدون برچسب در زبان مبدأ، یال‌های بین ویژگی‌های محوری (x_s, x_t) و ویژگی‌های غیرمحوری در زبان مبدأ y_s و نظرات بدون برچسب در زبان مقصد، یال‌های بین (x_s, x_t) و y_t را می‌سازد. به طور دقیق‌تر، وزن یال بین (x_s, x_t) و y_s، میزان هم‌رخدادی x_s و y_s است و وزن یال بین (x_s, x_t) و y_t، میزان هم‌رخدادی x_t و y_t. در این صورت یک ویژگی محوری که متشکل از کلمات هر دو زبان است با ویژگی‌های غیرمحوری در هر دو زبان یال مشترک دارد. از این رو، در گراف حاصل نقش ویژگی‌های غیرمحوری زبان مبدأ از ویژگی‌های غیرمحوری زبان مقصد، متمایز نیست.

۳-۲-۳- استخراج ویژگی‌های معنایی

با به‌کارگیری دو محدودیت، ویژگی‌های محوری انتخاب و گراف ارتباطشان با ویژگی‌های غیرمحوری ساخته شد. به دلیل دوزبانه بودن گراف و انجام خوشه‌بندی در فضای معنایی، ویژگی‌های استخراجی از این ماتریس ویژگی‌های معنایی نامیده شد. این ویژگی‌ها از نگاه دیگر حاوی مفاهیم کلی‌تر و سطح بالاتری است که به عبارتی معنای خاصی را دربردارد. این ویژگی‌ها بیان می‌کند هر کلمه چه میزان تعلقی به هر خوشه دارد. در ادامه روند استخراج ویژگی‌های معنایی برای مسأله بین‌زبانی توضیح داده می‌شود.

پس از ساخت ماتریس مجاورت طبق فرمول (۳)، ماتریس لاپلاس گراف با استفاده از فرمول (۴) محاسبه می‌شود. همان‌طور که گفته شد، این مرحله وزن‌های خام گراف را به وزن‌های نرمال شده تبدیل می‌کند و در نتیجه این وزن‌های جدید معنی‌دار است. برای مثال اگر در ماتریس اولیه A وزن یال بین f_{p_m} و f_{n_i} با وزن یال بین f_{p_m} و f_{n_j} مساوی و برابر w باشد و f_{n_i} ویژگی پرکاربردتری نسبت به f_{n_j} باشد، در ماتریس L وزن این دو یال برابر نیست و وزن یال f_{p_m} با f_{n_i} کم‌تر از وزن یالش با f_{n_j} است. زیرا مقدار w برای دو ویژگی f_{n_i} و f_{n_j} مفاهیم متفاوتی دارد و برای f_{n_j} حاوی اطلاعات بیش‌تری است.

اکنون با استفاده از فرمول (۶) تمام ویژگی‌های محوری و غیرمحوری، خوشه‌بندی می‌شود که خوشه‌های حاصل شامل جفت‌کلمه‌ها و تک‌کلمه‌های زبان مبدأ و زبان مقصد است. خصلت مهم این خوشه‌ها دوزبانه بودن آن‌ها است. اگر خوشه‌ها را فضای معنایی مستقل از زبان تعریف کنیم، وزنی که هر ویژگی محوری و یا غیرمحوری در هر خوشه دارد ویژگی معنایی آن محسوب می‌شود. با استفاده از فرمول (۸) می‌توان نظرات را از فضای ویژگی‌های محوری و غیرمحوری به فضای معنایی منتقل کرد که با توجه به ویژگی‌های معنایی، نظرات در دو زبان مبدأ و مقصد، ویژگی‌های مشترکی خواهد داشت.

پس از محاسبه میزان اختصاص هر نظر به هر خوشه، با استفاده از یک رده‌بند، تأثیر هر خوشه بر برچسب نظرات آموخته می‌شود. این مرحله بر روی نظرات برچسب‌خورده در زبان مبدأ که داده‌های آموزش به حساب می‌آید، صورت می‌گیرد. از اطلاعات آموخته شده، برای برچسب‌گذاری نظرات در زبان مقصد که داده‌های آزمون محسوب می‌شود، استفاده می‌شود. لازم به ذکر است که به دلیل انتقال نظرات به فضای معنایی و تعریف ویژگی‌ها مشترک برای دو زبان، رده‌بندی به صورت مستقیم و مشابه حالت تک‌زبانه انجام می‌گیرد.

۴- ارزیابی

در این بخش، آزمایش‌ها انجام شده برای ارزیابی روش پیشنهادی ارائه می‌شود. ابتدا مجموعه داده‌ای مورد استفاده و سپس روش پایه‌ای که برای مقایسه میزان کارایی روش پیشنهادی به کار گرفته شده معرفی می‌شود و در نهایت جزئیات آزمایش‌ها صورت گرفته بیان می‌شود.

۴-۱- مجموعه داده‌ها

مجموعه داده‌ای مورد استفاده، بخشی از داده‌های جمع‌آوری شده توسط پریتنهوفر[۲۲] [۲۷] است. این مجموعه داده‌ای به صورت تقریبی حاوی ۸۰۰ هزار نظر، از نظرات محصولات آمازون برای سه دسته از محصولات کتاب، موسیقی و دی‌وی‌دی است. همچنین این نظرات در چهار زبان انگلیسی، آلمانی، فرانسوی و ژاپنی موجود می‌باشد.

در آزمایش‌ها انجام شده زبان انگلیسی به عنوان زبان مبدأ و زبان آلمانی به عنوان زبان مقصد در نظر گرفته شد. همچنین از نظرات موجود در دامنه کتاب برای آزمایش‌ها استفاده شد. آمار این مجموعه داده‌ای در جدول ۱ قابل مشاهده است.

در داده‌های جمع‌آوری‌شده، هر کاربر علاوه بر ارائه نظر خود برای محصول مورد نظر، یک امتیاز از ۱ تا ۵ نیز به محصول داده است که برچسب نظرات با استفاده از همین امتیاز مشخص می‌شود. به نظرات دارای امتیاز ۴ تا ۵، برچسب مثبت و به نظرات دارای امتیاز ۱ تا ۲، برچسب منفی داده شده است. نظرات با امتیاز ۳ نیز برچسب خنثی تعلق می‌گیرد که از مجموعه داده‌ای حذف شده است. آزمایش‌ها بر روی نظرات مثبت و منفی صورت گرفته است. در نتیجه در این مجموعه داده‌ای اسناد موجود نظرمند هستند.

جدول ۱- آمار مجموعه داده‌ها در دامنه کتاب

	بدون برچسب	نظرات مثبت	نظرات منفی
انگلیسی	۵۰,۰۰۰	۲,۰۰۰	۲,۰۰۰
آلمانی	۱۶۵,۴۵۷	۲,۰۰۰	۲,۰۰۰

از نظرات دارای برچسب انگلیسی برای ساخت مدل رده‌بند استفاده می‌شود و برای تست این مدل، نظرات دارای برچسب آلمانی مورد استفاده قرار گرفته است.

برای منبع ترجمه لغت‌نامه گوگل انگلیسی به آلمانی[۲۳] (بدون استفاده از ماشین ترجمه) انتخاب شده است که احتمالات ترجمه در این آزمایش‌ها در نظر گرفته نشده است. در این لغت‌نامه هر کلمه انگلیسی، یک یا چند ترجمه آلمانی دارد که برای هر کلمه محتمل‌ترین ترجمه از مجموعه ترجمه‌ها، انتخاب شده است. لغت‌نامه منتخب حاوی ۲۶,۱۲۸ کلمه انگلیسی و ۱۷,۳۵۰ کلمه آلمانی است.

۴-۲- روش پایه

برای ارزیابی روش پیشنهادی دو روش پایه برای مقایسه انتخاب شد تا میزان کارایی و بهبود آن بهتر نمایش داده شود. یکی از روش‌های پایه انتخابی، روش یادگیری تناظرات ساختاری بین‌زبانی[۲۴] (CL-SCL) [۲۷] است. این روش که ابتدا برای حوزه بین‌دامنه‌ای مطرح شده بود [۲۸] و بعداً در حوزه بین‌زبانی مورد استفاده قرار گرفت، می‌تواند معیار خوبی برای ارزیابی روش پیشنهادی این مقاله باشد. در این روش نیز تعدادی کلمه به عنوان ویژگی‌های محوری انتخاب می‌شود و سپس تناظرات این ویژگی‌ها با سایر ویژگی‌ها محاسبه می‌شود که از تناظرات محاسبه‌شده برای پیش‌بینی رخداد ویژگی‌های محوری استفاده شده است. به عبارتی دیگر، برای هر جفت ویژگی محوری و غیرمحوری، وزنی استخراج می‌شود که این مقدار با همبستگی دو ویژگی نسبت مستقیم دارد. برای اجرای این روش از کد حاصل از مقاله پریتنهوفر [۲۷] که به صورت آزاد در اختیار قرار گرفته بود، استفاده شد.

روش پایه دیگری که مورد بررسی قرار گرفت، روش مبتنی بر مدل زبانی بین‌زبانی است که کارایی آن برای حوزه تک‌زبانه بررسی شده است [۱۰]. در این روش، با استفاده از مدل زبانی ۱-گرام‌ها، امتیاز مثبت و منفی برای اسناد محاسبه می‌شود. در مدل زبانی استفاده‌شده، با در نظر گرفتن مجموعه‌ای از اسناد، احتمال مشاهده یک سند با استفاده از بیشینه راست‌نمایی[۲۵] و هموارسازی لاپلاس[۲۶] محاسبه می‌شود.

در این روش، مدل زبانی نظرات با برچسب مثبت و مدل زبانی نظرات با برچسب منفی به صورت جداگانه ساخته می‌شود و برای هر نظر امتیاز جداگانه‌ای برای هر برچسب محاسبه می‌شود. این امتیاز، میزان شباهت مدل زبانی نظر مورد بررسی با دو مدل زبانی به‌دست‌آمده، است. برچسب انتخابی برای این نظر، برچسب مدل زبانی با مقدار شباهت بیش‌تر و یا مقدار اختلاف کم‌تر است. این اختلاف دو مدل زبانی با استفاده از روش KL-divergence [۳۲] محاسبه می‌شود (فرمول (۱۳)).

$$D(\theta_d \,\|\, \theta^{+/-}) = \sum_{w \in d} p(w \mid \theta_d) \log \frac{p(w \mid \theta_d)}{p(w \mid \theta^{+/-})} \qquad (۱۳)$$

در فرمول بالا θ_d مدل زبانی نظر مورد بررسی و $\theta^{+/-}$ مدل زبانی نظرات مثبت و یا نظرات منفی است. $D(\theta_d \,\|\, \theta^{+/-})$ نیز میزان اختلاف مدل زبانی نظر با مدل زبانی نظرات مثبت و یا نظرات منفی است.

برای استفاده از این روش در حوزه بین‌زبانی، با استفاده از لغت‌نامه گوگل، نظرات در زبان مبدأ به زبان مقصد ترجمه می‌شوند و مسأله به یک مسأله تک‌زبانه در زبان مقصد تبدیل می‌شود که در این صورت می‌توان از روش پیشنهاد شده [۱۰] استفاده کرد.

۴-۳- آزمایش‌ها

در این بخش به بررسی آزمایش‌ها انجام شده می‌پردازیم. برای رده‌بندی نظرات از رده‌بند SVM[۲۷] استفاده شده است و برای ارزیابی روش، معیار صحت[۲۸] در نظر گرفته شده است. این مقدار برابر با درصد نسبت تعداد نظرات با برچسب پیش‌بینی‌شده درست، به تعداد کل نظرات است (فرمول (۱۴)).

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \qquad (۱۴)$$

در این‌جا، TP و TN به ترتیب تعداد نظرات مثبت درست پیش‌بینی شده و تعداد نظرات منفی درست پیش‌بینی شده است و FP و FN نیز به ترتیب تعداد نظرات منفی که مثبت پیش‌بینی شده و تعداد نظرات مثبت که منفی پیش‌بینی شده است.

انتخاب ویژگی‌های محوری

اولین ارزیابی بر روی روش انتخاب ویژگی‌های محوری انجام شد. در آزمایش اول، ابتدا ویژگی‌های پرکاربرد در هر دو زبان (که پایه‌ای‌ترین روش محسوب می‌شود) به عنوان ویژگی‌های محوری انتخاب شد. این کلمات که تنها از پرکاربردترین کلمات در هر دو زبان محسوب می‌شود، خصلت خاص دیگری ندارد. در آزمایش بعدی ویژگی‌های محوری براساس اطلاعات متقابل بین برچسب نظرات و کلمات به کار رفته در آن‌ها انتخاب شد که لازم بود این روند محاسبه بر روی نظرات برچسب‌خورده صورت گیرد و کلماتی که بیشترین مقدار اطلاعات متقابل را دارد به عنوان ویژگی محوری انتخاب شود. از آن‌جایی که فرض شده این نظرات در زبان مقصد موجود نیست، تنها از نظرات برچسب‌خورده در زبان مبدأ استفاده شد و کلمات انتخاب شده با استفاده از منبع ترجمه، ترجمه و به جفت‌کلمه تبدیل شد. این کلمات به جز پرکاربرد بودن، گرایشی به یکی از برچسب‌های مثبت و یا منفی دارند و دربردارنده یک گرایش احساسی است. در آزمایش بعدی، انتخاب ویژگی‌های محوری طبق روندی که در زیر بخش ۴-۱ توضیح داده شد، صورت گرفت. در این روش که از اطلاعات متقابل بین ویژگی و زبان استفاده می‌کند شهود متفاوتی با دو روش قبلی دارد. در این روش سعی می‌شود ویژگی‌های محوری مستقل از زبان باشد که اطلاعاتی که از گراف دوبخشی حاصل می‌شود برای مسأله بین‌زبانی مفید باشد و در مرحله بعدی ارتباط این اطلاعات با برچسب استخراج شود. نتایج این آزمایش‌ها در جدول ارائه شده است.

ویژگی‌های انتخابی توسط این سه روش از لحاظ خصلت بسیار با هم متفاوت است. ویژگی‌های دسته اول ویژگی‌هایی است که هر دو کلمه آن به دفعات متعددی در نظرات به کار گرفته شده باشد. ویژگی‌های انتخابی دسته دوم، ویژگی‌های پرکاربرد، وابسته به برچسب و جهت‌دار است، اما ویژگی‌های دسته سوم، در هر دو زبان بسیار پرکاربرد و دو کلمه آن دارای رفتار مشابهی است. همان‌طور که مشاهده می‌شود روش سوم نسبت به هر دو روش اول و دوم از لحاظ آماری بهتر عمل کرده است. دلیل آن را می‌توان استفاده از اطلاعات هر دو زبان و خصلت استقلال از زبان این ویژگی‌ها برشمرد.

همچنین روش اول نسبت به روش دوم عملکرد بهتری داشته است که بیان‌گر این نکته است که در روش پیشنهادی ویژگی‌های محوری تنها کافی است نماینده خوبی برای زبان خود باشند و جهت‌دار بودن شرط لازم برای آن‌ها نیست. به عبارتی می‌توان گفت که با استفاده از اطلاعات متقابل بین کلمه و برچسب، امکان انتخاب ویژگی‌های محوری مناسب وجود دارد. اما تضمینی وجود ندارد که با استفاده از این روش، ویژگی‌های مستقل از زبان انتخاب شود. همچنین این ویژگی‌ها فقط با استفاده از داده‌های زبان مبدأ انتخاب می‌شود و ممکن است برای زبان مقصد ویژگی مناسبی نباشد.

جدول ۲- مقایسه روش‌های انتخاب ویژگی‌های محوری

معیار انتخاب	صحت
پرکاربرد بودن	۷۸/۱۴۸
پرکاربرد بودن + اطلاعات متقابل بین کلمه و برچسب	۷۸/۲۷۴
پرکاربرد بودن + اطلاعات متقابل بین ویژگی و زبان	**۸۱/۷۵۲**[۲۹]

هم‌ترازی کلمات

در روش پیشنهادی برای جفت کردن ویژگی‌های محوری ایده استفاده از یک لغت‌نامه مطرح شد. همان‌طور که در بخش ۲ گفته شد، در سال‌های اخیر روش‌های مبتنی بر نمایش طیفی کلمات در حوزه‌های مختلفی مورد استفاده قرار گرفته است. این دسته از روش‌ها در حوزه بین‌زبانی نیز وارد شده است. در [۳۳] با استفاده از ایده‌ای که در [۳۴] مطرح شده بود، با بهره‌گیری از یک پیکره موازی[۲۹] و داده‌های تک‌زبانه برای دو زبان مورد نظر، بردارهایی برای کلمات هر دو زبان استخراج می‌شود و می‌توان با استفاده از توابع محاسبه شباهت دو بردار، بردارهای نزدیک به هم را شناسایی کرد. در نتیجه برای کلمه از زبان مبدأ، کلماتی از زبان مقصد که دارای بردارهای نزدیک به بردار کلمه مورد نظر است، به دست می‌آید.

در این بخش جفت‌کلمه‌های کاندیدا برای ویژگی‌های محوری را با استفاده از این روش به دست می‌آوریم و با روش پیشنهادی که استفاده از یک لغت‌نامه است، مقایسه می‌کنیم.

در آزمایش اول، با استفاده از یک ابزار آماده[۳۰]، پیکره موازی آلمانی- انگلیسی و نظرات بدون برچسب، بردارهای کلمات استخراج شد و با استفاده از میزان شباهت کسینوسی، برای هر کلمه از زبان مبدأ، ۵ کلمه از زبان مقصد با نزدیک‌ترین بردار به بردار کلمه زبان مبدأ، انتخاب شد. کلماتی که از حد آستانه

کم‌تر ظاهر شده‌اند نیز از کاندیداهای ویژگی‌های محوری حذف شد. در نهایت مانند روش پیشنهادی با محاسبه مقدار اطلاعات متقابل میان ویژگی و زبان، جفت‌کلمه‌های مستقل از زبان به عنوان ویژگی‌های محوری انتخاب شد. در آزمایش دوم روند انجام شده طبق روندی است که در بخش ۳-۲ گفته شد. نتایج این دو آزمایش در جدول ۳ قابل مشاهده می‌باشد.

جدول ۳- مقایسه روش‌های متفاوت برای هم‌ترازی کلمات

روش هم‌ترازی	صحت
نمایش طیفی کلمات	۵۵/۰۲۸
لغت‌نامه	**۸۱/۷۵۲**

همان‌طور که مشاهده می‌شود، با استفاده از نمایش طیفی کلمات معادل خوبی برای کلمات به دست نیامده است. در نتیجه با استفاده از این ویژگی‌های محوری چگونگی ارتباط میان دو زبان به خوبی تشخیص داده نشده که منجر به کاهش بسیار زیاد کارایی شده است.

انتخاب ویژگی‌های غیرمحوری

آزمایش بعدی بر روی روش انتخابی ویژگی‌های غیرمحوری انجام شد. در ابتدا سعی شد برای این دسته از ویژگی‌ها از یک واژه‌نامه برای هر زبان استفاده شود. این واژه‌نامه‌ها حاوی کلمات مثبت، منفی و خنثی بود. ابتدا تنها کلمات جهت‌دار (مثبت و منفی) به عنوان ویژگی‌های غیرمحوری انتخاب شد. در آزمایش بعدی از کلمات خنثی نیز استفاده شد تا تاثیر وجودشان بررسی شود. در مرحله بعدی به جای استفاده از واژه‌نامه، از کلمات استفاده شده در نظرات به عنوان ویژگی‌های غیرمحوری استفاده شد. در یک آزمایش کلماتی انتخاب شدند که تعداد رخدادشان از یک مقدار کمینه بیش‌تر باشد و در آزمایش بعدی مقدار کمینه بسیار کم‌تری انتخاب شد تا تعداد کلمات بیش‌تری (بیش از ۴ برابر) برای ویژگی‌های غیرمحوری انتخاب شود و عملاً تنها از کلماتی که تنک بودند، استفاده نشد. نتایج این آزمایش‌ها در جدول ۴ آمده است.

جدول ۴- مقایسه روش‌های انتخابی ویژگی‌های غیرمحوری

ویژگی‌های غیرمحوری	صحت
کلمات جهت‌دار واژه‌نامه	۷۲/۵
واژه‌نامه	۷۲/۴۵
کلمات پرکاربرد	۷۹/۲
کلمات رایج	**۸۱/۴۵**

این آزمایش‌ها نشان می‌دهد وجود کلمات خنثی در ویژگی‌های غیرمحوری تأثیر منفی قابل توجهی بر کارایی روش نمی‌گذارد. همچنین هر چه تعداد ویژگی‌های غیرمحوری بیش‌تر باشد و در نتیجه ویژگی‌های معنایی برای کلمات بیش‌تری استخراج شود، کارایی روش پیشنهادی افزایش می‌یابد.

بررسی تأثیر ایست‌واژه‌ها[۳۱]

در آزمایش بعدی وجود و یا عدم وجود ایست‌واژه‌ها بررسی شده است. یک بار ایست‌واژه‌ها در متن نظرات حفظ شد و بار دیگر در مرحله پیش‌پردازش این کلمات از متن نظرات حذف شد. نتایج این بررسی در جدول ۵ نشان داده شده است.

جدول ۵- ارزیابی تأثیر ایست‌واژه‌ها در کارایی

	صحت
با وجود ایست‌واژه‌ها	۷۲/۵
بدون وجود ایست‌واژه‌ها	**۸۱/۷۱۲**

همان‌طور که مشاهده می‌شود وجود ایست‌واژه‌ها در این روش تأثیر منفی دارد و وجود این کلمات در این روش مؤثر واقع نمی‌شود و بهتر است حذف ایست‌واژه‌ها به عنوان یک مرحله از پیش‌پردازش بر روی نظرات انجام شود. علی‌رغم این‌که می‌توان ایست‌واژه‌ها را ویژگی‌های مستقل از زبان و همچنین غیرمبهم فرض کرد (در عمل نیز با در نظر گرفتن ایست‌واژه‌ها، تعداد زیادی از آن‌ها در مجموعه ویژگی‌های محوری قرار گرفت)، فاقد اطلاعات معنایی بودنشان را می‌توان از دلایل بروز این رفتار برشمرد که این خصوصیت ایست‌واژه‌ها سبب می‌شود گراف دوبخشی، نسبت به حالت حذف ایست‌واژه‌ها، حاوی اطلاعات مفید کم‌تری باشد و در نتیجه خوشه‌بندی با دقت کم‌تری صورت گیرد.

مقایسه با روش‌های پایه

در این قسمت نتایج مقایسه روش پیشنهادی با روش‌های پایه ارائه می‌شود. با توجه به نتایج آزمایش‌ها قبلی، روش پیشنهادی (CLSFD[۳۲]) بر روی مجموعه داده‌ای انجام گرفت. ابتدا ایست‌واژه‌ها از مجموعه داده‌ای حذف شد. سپس ۲۵۰ ویژگی محوری با استفاده از اطلاعات متقابل بین ویژگی و زبان انتخاب شد و کلمات کاربردی نظرات به عنوان ویژگی‌های غیرمحوری انتخاب شد و در نهایت ۱۰۰ ویژگی معنایی از گراف دوبخشی حاصل استخراج شد. در ضمن تأثیر تعداد ویژگی‌های محوری و تعداد ویژگی‌های معنایی در ادامه بررسی خواهد شد. برای روش CL-SCL نیز تعداد ویژگی‌های محوری ۲۵۰ انتخاب شد. به دلیل مفید بودن ایست‌واژه‌ها برای این دو روش پایه، در آزمایش‌ها روش CL-SCL و مدل زبانی، ایست‌واژه‌ها در مرحله پیش‌پردازش حذف نشد.

این آزمایش‌ها به صورت اعتبارسنجی متقابل ۵ بخشی انجام شد. در هر آزمایش ۳۲۰۰ نظر (۱۶۰۰ نظر مثبت و ۱۶۰۰ نظر منفی) در زبان مبدأ به عنوان داده آموزش و ۸۰۰ نظر (۴۰۰ نظر مثبت و ۴۰۰ نظر منفی) در زبان مقصد به عنوان داده آزمون انتخاب شد. میانگین صحت این ۵ آزمون در جدول ۶ گزارش شده است. برای مقایسه آماری این روش‌ها و بررسی معنادار بودن اختلاف کارایی روش‌ها از لحاظ آماری، آزمون زوج‌شده t بر روی نتایج حاصله انجام گرفت. نتایج این آزمایش‌ها نیز در جدول ۶ گزارش شده است. همان‌طور که مشاهده می‌شود، روش پیشنهادی نسبت به دو روش پایه بهتر عمل می‌کند و اختلاف کارایی آن نسبت به دو روش پایه از نظر آماری معنادار است.

جدول ۶- مقایسه روش پیشنهادی با روش‌های پایه

	صحت
مدل زبانی	۷۲/۷۷۵
CL-SCL	۷۹/۳۲۲
CLSFD	**۸۱/۷۱۲**[۳۳]

تحلیل دیگری که می‌توان برای مقایسه این روش‌ها انجام داد، دلیل تأثیر متفاوت ایست‌واژه‌ها در این سه روش است. همان‌طور که ذکر شد، دو روش پایه در حالت وجود ایست‌واژه‌ها عملکرد بهتری دارد و همچنین از منبع ترجمه برای ترجمه تمامی ویژگی‌ها استفاده می‌کند، اما روش پیشنهادی در حالت عدم وجود این کلمات بهتر عمل می‌کند و از منبع ترجمه تنها برای ترجمه تعداد محدودی

ویژگی بهره می‌برد. از آن‌جا که ترجمه اکثر ایست‌واژه‌ها ترجمه غیرمبهم و با دقت خوبی صحیح است، وجود این کلمات در حالتی که تمام ویژگی‌ها ترجمه می‌شود می‌تواند کیفیت ترجمه را به مقدار زیادی افزایش دهد. در نتیجه این رفتار برای این روش‌ها قابل پیش‌بینی و توجیه‌پذیر است.

روش پیشنهادی بین‌زبانی، علاوه بر کارایی بهتر از لحاظ آماری نسبت به دو روش دیگر، وابستگی کم‌تری به منبع ترجمه دارد که علت آن ترجمه شدن تنها ویژگی‌های محوری در این روش است. اما در دو روش دیگر لازم است تمام کلمات ترجمه شود. اگر کلمه‌ای در منبع ترجمه موجود نباشد و یا به خوبی ترجمه نشود، امکان کاهش کارایی در این روش‌ها وجود دارد. در روش پیشنهادی نبود ترجمه کلمات در منبع ترجمه تنها باعث حذف آن از کاندیداهای ویژگی محوری می‌شود به طوری که حضور این کلمات در ویژگی‌های غیرمحوری باعث استخراج ویژگی‌های معنایی برای چنین کلمات می‌شود. در نتیجه روش پیشنهادی علاوه بر وابسته نبودن به منبع ترجمه، از کلماتی که ترجمه‌ای برای آن‌ها موجود نیست هم بهره می‌برد. در صورتی که منبع ترجمه بهتری در دسترس باشد احتمال افزایش کارایی هر سه روش وجود دارد و با توجه به وابستگی بیش‌تر روش‌های پایه به منبع ترجمه، می‌توان پیش‌بینی کرد این افزایش کارایی برای روش‌های پایه نسبت به روش پیشنهادی بیش‌تر باشد.

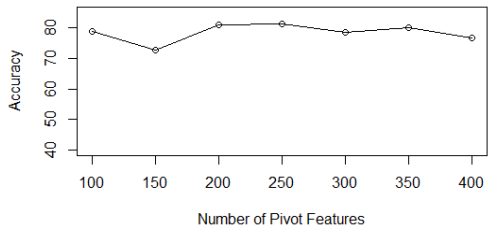
همان‌طور که در بخش قبل توضیح داده شد، نظرات در دو زبان مبدأ و مقصد با انتقال به فضای معنایی، دارای ویژگی‌های مشترکی می‌شود. در نتیجه مدل رده‌بندی که از روی نظرات زبان مبدأ ساخته می‌شود هم برای رده‌بندی نظرات زبان مبدأ و هم برای رده‌بندی نظرات زبان مقصد، قابل استفاده است. به همین دلیل با استفاده از ویژگی‌های معنایی استخراج‌شده، علاوه بر آزمایش‌ها بین‌زبانی، آزمایش‌ها تک‌زبانه را نیز می‌توان انجام داد. نتایج این آزمایش‌ها در هر دو زبان انگلیسی و آلمانی در جدول ۷ قابل مشاهده است. همان‌طور که مشاهده می‌شود کارایی روش در زبان آلمانی به میزان قابل توجهی از کارایی در زبان انگلیسی بهتر است. به نظر می‌رسد این تفاوت در کارایی به دلیل اختلاف تعداد نظرات بدون برچسب در دو زبان انگلیسی و آلمانی باشد. بنابراین می‌توان نتیجه گرفت این نظرات تأثیر بالایی در کارایی روش دارند و تعداد بیش‌تر این نظرات به استخراج بهتر ویژگی‌های معنایی کمک می‌کند.

جدول ۷- نتایج آزمایش‌ها تک‌زبانه

	صحت
انگلیسی	۷۷/۳۷۸
آلمانی	۸۴/۰۲۶

بررسی حساسیت به تعداد ویژگی‌های محوری

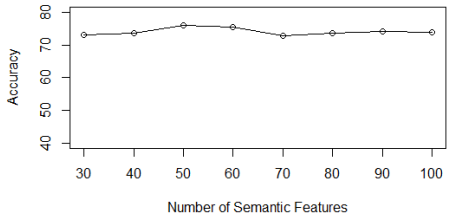
با تغییر تعداد ویژگی‌های محوری انتخابی و ثابت ماندن دیگر تنظیمات، تأثیر این پارامتر در روش پیشنهادی بررسی شد. نتایج این آزمایش‌ها در شکل ۲ نشان داده شده است. همان‌طور که ملاحظه می‌شود روش پیشنهادی در یک بازه بزرگ از ۲۰۰ تا ۳۵۰ به تعداد ویژگی‌های محوری حساسیت زیادی ندارد.



شکل ۲- حساسیت روش به تعداد ویژگی‌های محوری

بررسی حساسیت به تعداد ویژگی‌های معنایی

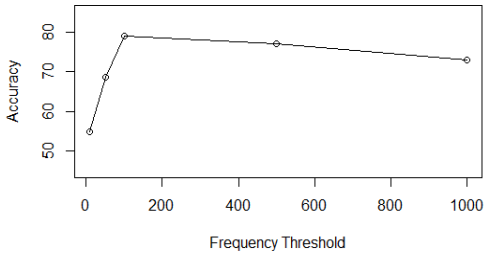
علاوه بر آزمایش‌ها گفته‌شده، حساسیت روش پیشنهادی نسبت به تعداد ویژگی‌های معنایی نیز بررسی شد. در این بررسی تعداد متفاوتی از ویژگی‌های معنایی با سایر تنظیمات مشابه، استخراج شد که روند تغییر کارایی روش در شکل ۳ نشان داده شده است. همان‌طور که ملاحظه می‌شود، تعداد ویژگی‌های معنایی استخراج‌شده تأثیر قابل ملاحظه‌ای بر کارایی روش نداشته و کارایی روش با تغییر مقدار این پارامتر تغییر زیادی نمی‌کند.



شکل ۳- حساسیت روش به تعداد ویژگی‌های معنایی

بررسی تأثیر میزان حدآستانه تکرار ویژگی‌های محوری

همان‌طور که گفته شد، یکی از محدودیت‌های تعریف‌شده برای انتخاب ویژگی‌های محوری، پرتکرار بودن هر دو ویژگی یک جفت ویژگی در زبان‌های متناظرشان است. خصلت پرتکرار بودن با انتخاب یک حد آستانه برای تعداد تکرار ویژگی‌ها در داده‌های بدون برچسب بررسی می‌شود. آزمایش‌های متعددی برای بررسی تأثیر مقدار این حد آستانه انجام گرفت. نتایج این آزمایش‌ها در شکل ۴ نشان داده شده است.



شکل ۴- تأثیر مقدار حد آستانه ویژگی‌های محوری بر روش پیشنهادی

همان‌طور که ملاحظه می‌شود، در صورتی که مقدار حد آستانه انتخابی بسیار کوچک باشد، ویژگی‌های خوبی به عنوان ویژگی‌های محوری انتخاب نمی‌شود. همچنین با انتخاب مقدار بسیار بالا برای حد آستانه، تعداد زیادی از ویژگی‌های مناسب، از کاندیداهای ویژگی‌های محوری حذف می‌شوند. در نتیجه مقداری برای حد آستانه مناسب است که هم از انتخاب ویژگی‌های کم‌تکرار و نامناسب جلوگیری کند و هم باعث حذف ویژگی‌های خوب نشود.

۵- نتیجه‌گیری

در این مقاله مسأله بین‌زبانی در حوزه نظرکاوی بررسی شد. روش پیشنهادی این مقاله، ابتدا با انتخاب دو دسته ویژگی، ویژگی‌های محوری و ویژگی‌های غیرمحوری، در هر دو زبان مبدأ و مقصد، ساخت یک گراف دوبخشی و در نهایت

استخراج ویژگی‌های معنایی برای ویژگی‌های غیرمحوری، توانست به کارایی خوبی دست پیدا کند. در این روش که از یک لغت‌نامه دوزبانه برای ترجمه ویژگی‌های محوری استفاده شد، برای ساخت بهتر گراف و استفاده از داده‌های بیشتر، از نظرات بدون برچسب در هر دو زبان بهره برد. از مزیت‌های این روش نسبت به روش‌های پایه بررسی‌شده، می‌توان به وابستگی کم این روش به حجم و کیفیت منبع ترجمه اشاره کرد. با توجه به تنظیمات و منابع مورد نیاز روش پیشنهادی، این روش می‌تواند برای زبان‌هایی که ماشین ترجمه مناسبی ندارد نیز مفید واقع شود. وجود نظرات بدون برچسب از ضروریات این روش است و هر چه تعداد این نظرات بیش‌تر باشد، دقت روش نیز افزایش پیدا می‌کند.

با توجه به خاصیت دوزبانه بودن گراف دوبخشی، ویژگی‌های معنایی برای ویژگی‌های غیرمحوری هر دو زبان به دست آمد که سبب شد امکان استفاده از مدل به‌دست‌آمده برای رده‌بندی نظرات در هر دو زبان فراهم شود. در آینده می‌توان حالت گسترش‌یافته این گراف را برای بیش از دو زبان بررسی کرد و از مدل حاصل برای رده‌بندی نظرات در چند زبان متفاوت استفاده کرد. همچنین با ایجاد منابع ترجمه با کیفیت‌های متفاوت می‌توان تأثیر میزان کیفیت منبع ترجمه بر کارایی روش را بررسی کرد. افزایش تعداد نظرات بدون برچسب در زبان انگلیسی و بررسی میزان تغییر در کارایی روش نیز از جمله کارهایی است که در آینده می‌توان به آن پرداخت.

با توجه به وابسته نبودن روش پیشنهادی به یک زوج زبان خاص، می‌توان با ایجاد مجموعه داده‌ای مناسب برای زبان فارسی، کارایی این روش را برای زوج زبان فارسی- انگلیسی نیز بررسی کرد.

مراجع

[1] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," *Proc. Annu. Meet. ACL*, 2002.

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up?: Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. EMNLP*, 2002.

[3] D. Tang, and et. al., "Sentiment Embeddings with Applications to Sentiment Analysis," *IEEE Trans. Knowledge and Data Engineering*, vol. 28, pp. 496-509, 2016.

[4] T. Zagibalov, and J. Carroll, "Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text," *Proc. Int'l Conf. COLING*, 2008.

[5] X. Wan, "Co-Training for Cross-Lingual Sentiment Classification," *Proc. ACL/AFNLP Int'l Joint Conf.*, 2009.

[6] X. Wan, "Bilingual Co-Training for Sentiment Classification of Chinese Product Reviews," *Computational Linguistics*, vol. 37, pp. 587-616, 2011.

[7] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in NIPS*, 2001.

[8] B. Pang, and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," *Proc. Annu. Meet. ACL*, 2004.

[9] Y. Dang, Y. Zhang, and H. Chen, "A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews," *IEEE Intelligent Systems*, vol. 25, pp. 46-53, 2010.

[10] Y. Hu, and et. al., "A Language Modeling Approach to Sentiment Analysis," *Proc. Int'l Conf. ICCS*, 2007.

[11] M. Thelwall, and et. al., "Sentiment in Short Strength Detection Informal Text," *JASIST*, vol. 61, pp. 2544-2558, 2010.

[12] T. Mikolov, and et. al., "Distributed Representations of Words and Phrases and their Compositionality," *Advances in NIPS*, 2013.

[13] A. L. Maas, and et. al., "Learning Word Vectors for Sentiment Analysis," *Proc. Annu. Meet. ACL*, 2011.

[14] I. Labutov, and H. Lipson, "Re-Embedding Words," *Proc. Annu. Meet. ACL*, 2013.

[15] D. Tang, and et. al., "Sentiment Embeddings with Applications to Sentiment Analysis," *IEEE Trans. Knowledge and Data Engineering*, vol. 28, pp. 496-509, 2016.

[16] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys," *Proc. Int'l Conf. COLING*, 2010.

[17] C. Tan, and et. al., "User-Level Sentiment Analysis Incorporating Social Networks," *Proc. ACM Int'l Conf. SIGKDD*, 2011.

[18] J. S. Olsson, D. W. Oard, and J. Hajic, "Cross-Language Text Classification," *Proc. Annu. Int'l ACM Conf. SIGIR*, 2005

[19] J. Brooke, M. Tofiloski, and M. Taboada, "Cross-Linguistic Sentiment Analysis: From English to Spanish," *Recent Advances in NIPS*, 2009.

[20] C. Wan, R. Pan, and J. Li, "Bi-Weighting Domain Adaptation for Cross-Language Text Classification," *Proc. Int'l Joint Conf. IJCAI*, 2011.

[21] M. S. Hajmohammadi, R. Ibrahim, A. Selamat, and H. Fujita, "Combination of Active Learning and Self-Training for Cross-Lingual Sentiment Classification with Density Analysis of Unlabelled Samples," *Information Sciences*, vol. 317, pp. 67-77, 2015.

[22] D. Gao, and et. al., "Cross-Lingual Sentiment Lexicon Learning with Bilingual Word Graph Label Propagation," *Computational Linguistics*, vol. 41, pp. 21-40, 2015.

[23] M. S. C. Almeida, and et. al., "Aligning Opinions: Cross-Lingual Opinion Mining with Dependencies," *Proc. ACL/AFNLP Joint Conf.*, 2015.

[24] H. Guo, and et. al., "OpinionIt: A Text Mining System for Cross-Lingual Opinion Analysis," *Proc. ACM CIKM*, 2010.

[25] S. Jain, and S. Batra, "Cross Lingual Sentiment Analysis using Modified BRAE," *In Proc. Conf EMNLP*, 2015.

[26] P. Prettenhofer, and B. Stein, "Cross-Language Text Classification Using Structural Correspondence Learning," *Proc. Annu. Meet. ACL*, 2010.

[27] P. Prettenhofer, and B. Stein, "Cross-Lingual Adaptation Using Structural Correspondence Learning," *ACM TIST*, vol. 3, p. 13, 2011.

[28] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification," *Proc. Annu. Meet. ACL*, 2007.

[29] S. J. Pan, and et. al., "Cross-Domain Sentiment Classification via Spectral Feature Alignment," *Proc. Int'l Conf. WWW*, 2010.

[30] G. Zhou, and et. al., "Cross-Domain Sentiment Classification via Topical Correspondence Transfer," *Neurocomputing*, vol. 159, pp. 298-305, 2015.

[31] K. W. Church, and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, vol. 16, pp. 22-29, 1990.

[32] T. M. Cover, and J. A. Thomas, "Elements of Information Theory," *Wiley-Interscience*, 1991.

[33] S. Gouws, Y. Bengio, and G. Corrado, "BilBOWA: Fast Bilingual Distributed Representations without Word Alignments," *Proc. Int'l Conf. Machine Learning*, 2015.

[34] Mikolov, Tomas, Quoc V. Le, and I. Sutskever, "Exploiting Similarities among Languages for Machine Translation," *CoRR*, abs/1309.4168, 2013.

شیما اسمعیلی تفت مدرک کارشناسی خود را در رشته مهندسی فناوری اطلاعات از دانشگاه تهران در سال ۹۲ دریافت کرد. در همان سال نیز با استفاده از سهمیه استعدادهای درخشان در مقطع کارشناسی‌ارشد مشغول به تحصیل شد. سپس در سال ۹۵ موفق به اخذ مدرک کارشناسی‌ارشد در رشته مهندسی فناوری اطلاعات از دانشگاه تهران گردید. علایق پژوهشی او متن‌کاوی، بازیابی اطلاعات و داده کاوی می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

shima.esmaeili@ut.ac.ir

آزاده شاکری استادیار دانشکده مهندسی برق و کامپیوتر پردیس دانشکده‌های فنی دانشگاه تهران است. او مدرک دکترای خود را در سال ۱۳۸۷ از دانشگاه ایلینویز اوربانا- شمپین در آمریکا دریافت کرد. زمینه‌های پژوهشی مورد علاقه وی مدیریت اطلاعات متنی، بازیابی اطلاعات، متن کاوی، و داده کاوی می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

shakery@ut.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۸/۱۶

تاریخ اصلاح: ۱۳۹۴/۱۰/۱۳

تاریخ قبول شدن: ۱۳۹۴/۱۰/۲۳

نویسنده مرتبط: دکتر آزاده شاکری، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران.

[1]Sentiment Analysis
[2]Opinion Mining
[3]Sentiment Orientation
[4]Text Mining
[5]Subjective
[6]Objective
[7]Seed
[8]Part of Speech Tag
[9]Language Model
[10]Word Embedding
[11]Twitter
[12]Lexicon
[13]Latent Dirichlet Allocation
[14]Spectral Feature Alignment
[15]Domain-Independent Features
[16]Domain-Specific Features
[17]Mutual Information
[18]Singular Value Decomposition
[19]Soft Clustering
[20]Sparse
[21]Pivot Features
[22]Prettenhofer
[23]https://translate.google.com/#en/de/
[24]Cross-Lingual Structural Correspondence Learning
[25]Maximum Likelihood
[26]Laplace Smoothing
[27]http://svmlight.joachims.org/
[28]Accuracy
[29]Parallel Corpus
[30]https://github.com/gouwsmeister/bilbowa
[31]Stopwords
[32]Cross-Lingual Semantic Feature Derivation

علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۶۰-۶۹
مقاله عادی

ارائه یک شبکه روی تراشه با کارآیی بالا و توان مصرفی کم برای شبکه‌های عصبی

نسرین اکبری　　بیتا دبیری　　مهدی مدرسی

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

چکیده

پیاده‌سازی سخت‌افزاری شبکه‌های عصبی به دلیل سفارشی‌سازی ساختار سخت‌افزار و حذف سربار نرم‌افزار سهم به‌سزایی در بهینه‌سازی توان و تاخیر انجام محاسبات عصبی دارد. نظر به اهمیت ارتباطات بین نورون‌ها در کارایی کلی شبکه‌های عصبی، در این مقاله یک هم‌بندی نوین شبکه روی تراشه جهت مدیریت ترافیک شبکه‌های عصبی ارائه شده است. این هم‌بندی، که براساس هم‌بندی معروف dragonfly ساخته شده است، برای انجام ترافیک چندپخشی و کاهش اتصالات بهینه گشته است. این هم‌بندی یک نمونه از هم‌بندی‌های سلسله مراتبی است و گره‌ها ابتدا در قالب گروه‌هایی تقسیم شده و در داخل هر گروه، از یک گذرگاه مشترک برای ارتباط آن‌ها استفاده می‌شود. سپس یک هم‌بندی سطح بالاتر گروه‌ها را به یکدیگر متصل می‌سازد. مشخصه اصلی هم‌بندی ارائه شده قطر کم و توانایی مناسب در انجام همه‌پخشی است. در این شبکه با انجام زمان‌بندی ارتباطات در زمان طراحی، از پیچیدگی مسیریاب‌ها کم شده که این امر زمینه‌ساز کاهش بیشتر توان و تاخیر شبکه می‌شود. این مقاله هم‌بندی پیشنهادی را با چند هم‌بندی پیشین مقایسه می‌کند که نتایج، نشان‌دهنده‌ی کاهش چشم‌گیر توان مصرفی و زمان تأخیر ارسال بسته‌ها و نیز افزایش گذردهی کلی شبکه تحت ترافیک چندپخشیِ شبکه‌های عصبی است.

کلمات کلیدی: شبکه روی تراشه، شبکه عصبی، راهگزینی مدار، کم‌توان، هم‌بندیdragonfly.

۱- مقدمه

پیاده‌سازی سخت‌افزاری شبکه‌های عصبی[1] یکی از موضوعاتی است که در سال‌های اخیر مورد توجه فعالان صنعتی و دانشگاهی مهندسی کامپیوتر قرار گرفته است. یک دلیل این امر نیاز روزافزون به عملکرد هوشمند در طیف وسیعی از سیستم‌های کامپیوتری، از حسگرها و سیستم‌های نهفته کوچک گرفته تا سرویس‌دهنده‌های بزرگ داده می‌باشد که شبکه‌های عصبی مصنوعی بهترین راه برای پیاده‌سازی الگوریتم‌های هوشمند شناخته می‌شوند.

اجرای یک شبکه عصبی بزرگ به‌صورت بی‌درنگ و با نرخ ورودی زیاد مستلزم انجام حجم زیادی از محاسبات سنگین اعشاری بوده که نیازمند به کارگیری یک ریزپردازنده و یا پردازنده‌ی گرافیکی قدرتمند است. از طرف دیگر، استفاده از پردازنده‌های قوی توان مصرفی و هزینه‌ی سیستم را به شدت بالا می‌برد که یک سامانه نهفته[2] مانند بینایی ربات جهت تشخیص چهره، چه از لحاظ هزینه‌ی تمام شده و چه از لحاظ توان مصرفی نمی‌تواند چنین هزینه‌ای را متحمل شود؛ بنابراین، استفاده از شتاب‌دهنده‌های سخت‌افزاری[3] که با پیاده‌سازی موازی شبکه عصبی بر روی سخت‌افزار (مثلا بر روی یک FPGA) و حذف سربار اجرای نرم‌افزار مدت زمان اجرا و توان مصرفی را کاهش می‌دهند، یکی از راه‌های مفید برای اجرای مناسب شبکه‌های عصبی است.

برای پاسخ به این نیاز، در سمت صنعت، شرکت‌های بزرگ طراحی و ساخت تراشه و سیستم‌های کامپیوتری مانند IBM، Intel، ARM، Xilinx، nVidia، Google، و بسیار شرکت‌های بزرگ و کوچک دیگر، اقدام به ساخت تراشه‌های خاص شبکه‌های عصبی کرده‌اند و یا هسته‌های پردازشی بر مبنای شبکه‌های عصبی را در محصولات خود مجتمع ساخته‌اند [۱][۲][۳][۴].

در سمت دانشگاه نیز گروه‌های معماری کامپیوتری بسیار در حال کار بر روی زمینه پیاده‌سازی سخت‌افزاری شبکه‌های عصبی هستند که این جریان در تعداد روز افزون مقالات در این زمینه در کنفرانس‌ها و ژورنال‌های معتبر رشته کامپیوتر منعکس می‌گردد [۵].

علاوه بر کاربرد در سیستم‌های هوشمند، در برخی پژوهش‌های قبلی نشان داده شده است که می‌توان از شبکه‌های عصبی برای پیاده‌سازی سریع توابع با بار

پردازشی سنگین استفاده کرد و یک تابع پیچیده محاسباتی را با مجموعه‌ای از عملیات ضرب و جمع پیاده‌سازی نمود. از این رو، استفاده از یک سخت‌افزار خاص منظوره برای محاسبات عصبی در پردازنده‌های همه منظوره و پردازنده‌های گرافیکی [۶] پیشنهاد شده است.

بیشتر سخت‌افزاره‌ای شتاب‌دهنده شبکه‌های عصبی به صورت یک تراشه بسیار هسته‌ای[۴] ساخته می‌شوند تا بتوانند از موازات ذاتی این مدل محاسباتی برای افزایش سرعت و برون‌دهی[۵] بهره ببرند. از آنجا که تعداد اتصالات بین نورون‌ها بسیار زیاد است (نورون‌های هر لایه، به صورت کامل به نورون‌های لایه‌ی بعد داده ارسال می‌نمایند) شبکه روی تراشه[۶] که بین هسته‌های پردازشی ایجاد می‌گردد، از تاثیر زیادی بر کارایی کل سیستم برخوردار است [۷].

به بیان دیگر، همان‌گونه که در بخش بعد گفته خواهد شد، هر دو عملیات ساده ضرب و جمع اعشاری که در یک واحد پردازشی شبکه عصبی انجام می‌شود نیازمند دریافت یک داده از واحدهای دیگر است که این امر نشان‌دهنده نسبت بالای ارتباطات به محاسبات در این مدل پردازشی است. این حجم بالای ارتباطات نسبت توان مصرفی و تاخیر شبکه روی تراشه را به کل توان و تاخیر سیستم افزایش می‌دهد و لذا بهینه‌سازی شبکه روی تراشه در این سیستم‌ها از اهمیت حیاتی برخوردار است.

در بسیاری از پژوهش‌های پیشین، از شبکه‌های روی تراشه راه‌گزینی بسته[۷] با هم‌بندی‌های[۸] معمولی مانند درخت[۹][۸]، توری[۱۰] [۹]، و Clos [۱۰]، به عنوان زیرساخت ارتباطی نورون‌ها استفاده شده است.

اما بسیاری از شبکه‌های عصبی که در سیستم‌های نهفته استفاده می‌شوند که این سیستم‌ها با یک نرخ ثابت ورودی‌ها را به شبکه عصبی اعمال می‌کنند. برای مثال، یک بخش هوشمند تشخیص چهره در یک سامانه نهفته، ورودی دوربین را با یک نرخ ثابت دریافت کرده و جهت تشخیص پارامترهای مورد نظر به بخش شبکه عصبی ارسال می‌دارد [۱۱]. از آنجا که در شبکه عصبی هم تمام نورون‌های که عملیات پردازشی یکسان (اما با عملوندهای متفاوت) را بر روی داده‌ها انجام می‌دهند، به شرط دریافت همزمان ورودی، خروجی خود را در یک زمان تولید می‌کنند.

این نظم در زمان‌بندی، در کنار نظم ذاتی در الگوی توزیع مکانی ترافیک در شبکه‌های عصبی (که نورون‌های هر لایه خروجی خود را برای تمام نورون‌های لایه بعد ارسال می‌نمایند) می‌تواند برای ساده‌سازی شبکه روی تراشه و در نتیجه کاهش مساحت، تاخیر، و توان مصرفی آن استفاده گردد.

در این مقاله، ما از این ویژگی نظم و پیش‌بینی‌پذیری در ترافیک تولیدی شبکه‌های عصبی استفاده کرده و یک شبکه روی تراشه ساده با زمانبندی ایستا[۱۱] جهت ایجاد ارتباط بین نورون‌ها ارائه می‌دهیم.

این شبکه با انجام یک زمان‌بندی ایستا برای ارسال داده‌های هر نورون به سایر نورون‌ها، نوعی راه‌گزینی مدار[۱۲] را برای ارتباطات بین نورون‌ها فراهم می‌آورد. با انجام زمانبندی در زمان طراحی دیگر نیاز به وجود مسیریاب‌های[۱۳] هوشمند در شبکه روی تراشه نیست، زیرا زمان ارسال داده توسط هر نورون و نیز مسیری که برای تحویل دادن داده به تمام نورون‌هایی که به آن داده نیاز دارند از قبل تعیین شده و در جدول زمانبندی در تمام مسیریاب‌ها قرار گرفته است.

با حذف هوشمندی از مسیریاب‌ها، توان و مساحت آنها به اندازه قابل توجهی کاهش می‌یابد زیرا دیگر نیازی به مسیریابی، داوری، تخصیص کانال مجازی، کنترل جریان، و ذخیره‌سازی بسته‌ها در هر مسیریاب نمی‌باشد. حذف این فعالیت‌ها از مسیریاب‌های شبکه روی تراشه افزون بر کاهش توان و مساحت به کاهش تاخیر ارتباطات هم کمک می‌کند، زیرا انجام این مراحل مستلزم صرف زمان است و در شبکه‌های روی تراشه، هر یک از مراحل فوق در یک مدت کلاک انجام می‌شود [۱۲]. در معماری ارائه شده در این مقاله، مانند شبکه‌های راه‌گزینی مدار، هر داده پس از رسیدن به هر مسیریاب، در کلاک بعدی به صورت بی‌درنگ طبق جدول زمانبندی داخل مسیریاب به گره بعدی ارسال می‌شود و از این رو تاخیرکمی به بسته‌ها اعمال می‌کند.

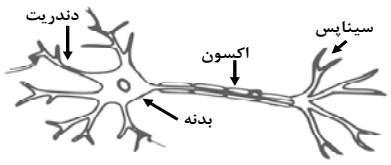
در این مقاله، شبکه روی تراشه پیشنهادی از هم‌بندیdragonfly استفاده می‌کند [۱۳]. Dragonfly یک هم‌بندی مدرن و با قطر[۱۴] کم است که در یکی از پیشرفته‌ترین نسل‌های ابر رایانه‌های شرکت Cray به کار رفته است [۱۴]. دلیل این انتخاب راحتی پیاده‌سازی آن در سطح مدار، قطر کم، و توانایی بالا در انجام ارتباطات چندپخشی[۱۵] است که آن را برای ترافیک شبکه‌های عصبی مناسب می‌کند.

در ادامه این مقاله، در بخش دوم به مروری بر مفاهیم پایه در شبکه‌های عصبی و ساختار آن‌ها خواهیم پرداخت. در بخش سوم، کارهای پیشین و مربوط به پیاده‌سازی سخت‌افزاری شبکه‌های عصبی مطرح شده است. بخش چهارم به معرفی و بررسی معماری شبکه روی تراشه پیشنهادی اختصاص دارد. در فصل بخش پنجم، نتایج حاصل از مقایسه طرح پیشنهادی با چند کار مطرح در این زمینه آمده است. در نهایت، فصل ششم به نتیجه‌گیری و جمع‌بندی در مورد معماری ارائه شده اختصاص دارد.

۲- پیش‌زمینه پژوهش

۲-۱- شبکه‌های عصبی زیستی

شبکه عصبی یک مدل محاسباتی بر مبنای یادگیری است که از سیستم عصبی موجودات زنده الهام گرفته شده است. نورون‌ها (یا سلول‌های عصبی) بخش اصلی مغز انسان را تشکیل داده و دارای یک شبکه‌ی پیچیده ارتباطی برای تبادل اطلاعات بین یکدیگر می‌باشند [۱۵]. شکل ۱ ساختار ساده شده یک نورون در سیستم عصبی انسان را نشان می‌دهد.



شکل ۱- ساختار کلی یک نورون در سیستم عصبی

هر نورون شامل سلول بدنه و دو نوع شاخه یعنی اکسون و دندریت برای دسترسی به خارج می‌باشد. هر نورون، سیگنال‌های نورون‌های دیگر را به شکل پالس الکتریکی از طریق دندریت (گیرنده) دریافت کرده و سیگنال‌های تولیدی خود را از طریق اکسون (فرستنده) به نورون‌های دیگر منتقل می‌کند. در این ساختار، در محلی به نام سیناپس، اکسون‌های هر نورون به دندریت‌های نورون دیگر متصل می‌شوند. در سیستم‌های زیستی، یادگیری با تنظیمات اتصالات سیناپسی که بین نورون‌ها قرار دارد به وجود می‌آید. به بیان دیگر، یادگیری با تغییر الگوی ارتباط بین نورون‌ها و شدت اثر آن‌ها بر یکدیگر ایجاد می‌گردد.

نکته جالب در مورد شبکه عصبی مغز انسان آن است که ارتباط بین نورون‌ها از طریق پالس‌هایی با فرکانس در حد چند صد هرتز انجام می‌شود که بارها از فرکانس کاری یک رایانه معمولی کمتر است. اما، با این حال، تصمیمات پیچیده‌ای مانند تشخیص چهره در مغز بسیار سریع و در حد چند میلی ثانیه انجام می‌گردد. دلیل این سرعت و کارآیی، پردازش موازی و توزیع‌شده در مقیاس بسیار بزرگ در مغز است: در مغز انسان بیش از ۱۰۰ میلیارد نورون وجود دارد که هریک از آن‌ها به طور متوسط با ۷ هزار نورون دیگر در ارتباط بوده و یک فرآیند پردازش را در کنار هم و با موازات فراوان به انجام می‌رسانند [۱۵].

۲-۲- شبکه‌های عصبی مصنوعی

انواع زیادی از مدل‌های محاسباتی تحت عنوان کلی شبکه‌های عصبی معرفی شده‌اند که هر کدام از بخشی از قابلیت‌های سیستم عصبی الهام گرفته و برای دسته‌ای از کاربردها قابل استفاده هستند. پر کاربردترین این مدل‌ها برای سیستم‌های نهفته هوشمند (مثلا در یک تشخیص‌دهنده تصویر) شبکه‌های عصبی پرسپترونی چند لایه (MLP)[۱۶] است که از یک لایه ورودی، n-1 لایه میانی (و یا پنهان) و یک لایه خروجی تشکیل شده است. بیشترین حالت پیاده‌سازی MLPها در حل مسائل کوچک دارای یک لایه ورودی، یک‌لایه میانی و یک لایه خروجی است. MLPها یکی از مهمترین رده‌های خانواده شبکه‌های عصبی پیشران[۱۷] هستند. در معماری پیشران، جریان داده فقط در یک جهت و از لایه ورودی به سمت لایه‌های پنهان و از آنجا به لایه خروجی حرکت می‌نماید.

به بیان دیگر، در این ساختار، ورودی نورون‌های لایه i خروجی نورون‌های لایه i-1 است (به‌غیر از اولین لایه پنهانی که به ورودی‌های شبکه عصبی متصل است). هر نورون در لایه‌های میانی و خروجی که در شکل ۲ نشان داده شده است، خروجی تمام نورون‌های لایه قبل را گرفته و مجموع حاصل ضرب ورودی‌ها در وزن مخصوص به هر ورودی را به صورت

$$y = \sum_{j=1}^{n} w_j x_j$$

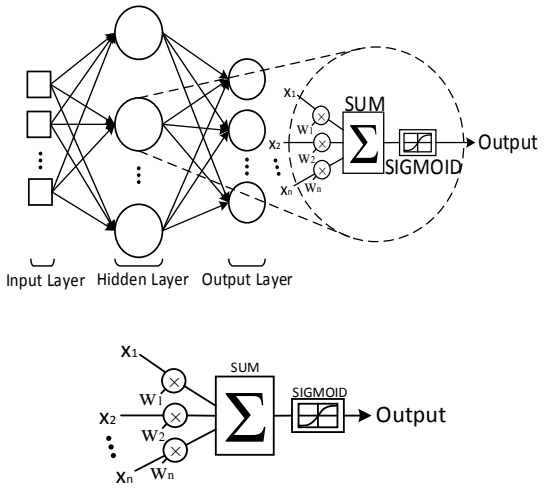
محاسبه می‌کند (ضرب نقطه‌ای آرایه وزن W و ورودی X در یکدیگر). در نهایت هم یک تابع فعال‌سازی[۱۸] (AF) جهت تولید نتیجه خروجی بر روی حاصل جمع نهایی (y) اعمال می‌شود. سپس جواب حاصل به تمام نورون‌های لایه بعد ارسال می‌گردد.

تابع سیگموئید[۱۹] یکی از پر استفاده‌ترین توابع فعال‌سازی در شبکه‌های عصبی مصنوعی بوده که تابعی هموار و اکیدا صعودی می‌باشد و به شکل زیر تعریف می‌شود (β پارامتر شیب است):

$$g(y)=1/(1+\exp(-\beta y))$$

در فرمول بالا، y ورودی تابع بوده که در واقع همان مجموع حاصل‌ضرب تمام ورودی‌ها در وزن‌های متناظرِ خود در یک نورون است [۱۶].

چون پیاده‌سازی این تابع مشکل است، معمولا در پیاده‌سازی سخت‌افزاری شبکه‌های عصبی، خروجی تابع سیگموید با یک جدول جستجو[۲۰] تقریب زده می‌شود.



شکل ۲- یک شبکه عصبی مصنوعی سه لایه (MLP) به همراه ساختار داخلی نورون

این بدان معنا است که به جای پیاده‌سازی مستقیم تابع، برخی مقادیر ورودی و خروجی تابع در یک جدول ذخیره شده و در زمان اجرا نزدیک‌ترین مقدار موجود در جدول به خروجی‌های مورد نظر یافته شده و جواب متناظر با آن به عنوان خروجی نورون در نظر گرفته می‌شود. به دلیل ماهیت تقریبی[۲۱] شبکه‌های عصبی، کاهش دقت پیاده‌سازی جدولی، به شرط وجود مقادیر کافی در جدول، در بسیار از مواقع باعث کاهش دقت محسوسی نخواهد شد.

در شبکه‌های عصبی مصنوعی وزن‌های هر نورون در فرآیندی به نام یادگیری یا آموزش[۲۲] تعیین می‌شود. توانایی یادگیری از اصلی‌ترین ویژگی‌های شبکه عصبی است. فرآیند یادگیری در حوزه شبکه‌های عصبی مصنوعی را می‌توان به شکل تنظیم وزن‌های شبکه برای حل یک مسئله خاص نگاه کرد، به گونه‌ای که شبکه بتواند به‌طور کارآمد آن مسئله بخصوص را با معماری ارائه شده در شکل ۲ حل کند.

الگوریتم‌های زیادی برای یادگیری شبکه‌های عصبی مصنوعی وجود دارد که سه مدل اصلی آن‌ها عبارتند از یادگیری نظارتی، یادگیری بدون نظارت و یادگیری ترکیبی. یادگیری نظارتی[۲۳]، که معمول‌ترین روش یادگیری بوده و در این مقاله استفاده شده است، بر پایه ارائه مجموعه بزرگی از ورودی‌ها و جواب صحیح متناظر با هر ورودی به شبکه‌های عصبی است. در طی این فرآیند، ورودی‌ها یکی یکی به شبکه داده شده و وزن تمام نورون‌ها طوری تعیین و تنظیم می‌شوند که شبکه بتواند نزدیک‌ترین جواب را به جواب صحیح تولید کند. پس از تکرار اعمال ورودی‌ها به مقدار کافی، شبکه قادر است جوابی مناسب به ورودی‌هایی که براساس آن‌ها آموزش دیده است و همچنی نورودی‌های جدید پیدا نماید.

هدف این مقاله پیاده‌سازی شبکه‌های عصبی بر روی شبکه بر روی تراشه است و توضیح کامل الگوریتم‌های یادگیری فراتر از بحث‌های مورد توجه ما به شمار می‌رود. مانند بیشتر پیاده‌سازی‌های سخت‌افزاری شبکه‌های عصبی، فرض ما بر این است که فاز آموزش شبکه و تعیین بردار وزن‌های هر کدام از نورون‌ها به‌صورت نابرخط[۲۴] و از قبل انجام شده است. یکی از قوی‌ترین ابزار موجود برای انجام فرآیند یادگیری و تنظیم وزن‌ها نرم‌افزار MATLAB است. در بیشتر سیستم‌ها، ابتدا آموزش شبکه عصبی توسط MATLAB انجام می‌شود و سپس وزن‌های به دست آمده جهت تولید خروجی نورون‌ها در زمان اجرا مورد استفاده قرار می‌گیرند.

یکی از ویژگی‌های مهم شبکه‌های عصبی خاصیت تقریبی بودن آنهاست. این به آن معنا است که این مدل محاسباتی قادر به یافتن جواب یک مسئله با دقت کامل نیست بلکه جواب را با درصدی از خطا محاسبه می‌کند. هر چه کیفیت و در برخی مسایل اندازه و عمق شبکه عصبی افزایش پیدا کند، درصد خطا در خروجی کمتر خواهد شد و تقریب نزدیکتری به جواب اصلی مسئله انجام می‌گیرد. اما این درصد به صفر نمی‌رسد و بنابراین این مدل محاسباتی برای کاربردهای نیازمند به جواب دقیق (مثلا در کاربردهای بانکداری و یا محاسبات دقیق علمی) مناسب نیست. با این وجود طیف گسترده‌ای از کاربردهای روزمره‌ی سیستم‌های کامپیوتری مانند کاربردهای پردازش سیگنال، چندرسانه‌ای، و تشخیص و تحلیل الگو نیاز به جواب با دقت کامل ندارند و درصدی از خطا در آن‌ها قابل قبول و در برخی موارد (مانند کاربردهای چندرسانه‌ای) نامحسوس است.

علاوه بر مدل شبکه عصبی MLP که یک خانواده از رده شبکه‌های عصبی مصنوعی است، شبکه‌های اسپایکی[۲۵] نیز یکی دیگر از رده‌های مهم شبکه‌های عصبی به شمار می‌روند [۱۷].

شبکه‌های اسپایکی مدل دقیق‌تری از شبکه‌های عصبی بیولوژیکی هستند. آنچه شبکه‌های عصبی مصنوعی (که در این بخش معرفی شدند) از سیستم عصبی زیستی الهام گرفته‌اند، نحوه به هم پیوستن نورون‌ها و مقادیر رد و بدل شده بین آنها بوده، اما زمان رسیدن ورودی برای آنها اهمیتی ندارد. ولی در دنیای واقعی، نورون‌های مغز با یکدیگر از طریق ارسال پالس‌های کوچکی ارتباط برقرار می‌کنند و هر نورون در ورودی خود قطاری از پالس‌ها را گرفته و در خروجی خود نیز

قطاری از پالس‌ها را با فاصله زمانی مشخص برای نورون بعدی ارسال می‌کند. در شبکه‌های پالسی، با استفاده از مدلی که در آن پارامتر زمان نیز وارد شده است و اطلاعات با تغییر فاصله زمانی و فرکانس قطار پالس خروجی کد شده است، عملکرد شبکه بسیار بیشتر از قبل به عملکرد مغز نزدیک می‌شود. از این رو، از این مدل برای تحقیقات مرتبط با مدل‌سازی و شبیه‌سازی مغز استفاده می‌شود.

تمرکز ما در این مقاله بر شبکه‌های عصبی مصنوعی است، اما معماری ارائه شده می‌تواند برای پیاده‌سازی شبکه‌های اسپایکی نیز به کار گرفته شود.

۲-۳- کاربردهای شبکه‌های عصبی مصنوعی

امروزه گستردگی کاربرد شبکه‌های عصبی به تمام زمینه‌های علمی و صنعتی که نیازی به تشخیص الگو، تحلیل، تصمیم‌گیری، تخمین، و پیش‌بینی دارند رسیده است. به طور خاص برخی از کاربردهای شبکه‌های عصبی در ادامه آمده است.

شناسایی و طبقه‌بندی الگو[۲۶]: هدف از این مسئله تشخیص این مطلب است که تعیین شود یک ورودی متعلق به کدام الگو و یا رده‌های موجود است. به عنوان مثال می‌توان از تشخیص صدای یک فرد از روی امواج صوتی، تشخیص چهره با استفاده از تطبیق دادن تصویر شخص با ورودی‌هایی که از قبل به سیستم داده شده‌اند، و یا شناسایی اثر انگشت نام برد. هم‌چنین از دیگر کاربردهای آن می‌توان به طبقه‌بندی امواج مغز و قلب غیره اشاره کرد.

خوشه‌بندی[۲۷] و دسته‌بندی: در خوشه‌بندی (که از آن به‌عنوان طبقه‌بندی الگو بدون ناظر نیز یاد می‌شود)، هیچ طبقه‌بندی از قبل مشخص شده‌ای وجود ندارد. یک الگوریتم خوشه‌بندی، شباهت بین الگوها را کشف کرده و آن‌ها را در یک خوشه قرار می‌دهد.

برازش توابع[۲۸]: شبکه عصبی قادر است تا تنها با برخورد با تعداد محدودی نمونه، یک قانون کلی از آن را به دست آورده و نتایج این آموخته‌ها را به سایر نمونه‌ها تعمیم دهد. فرض کنید n الگوی آموزشی (به شکل زوج مرتب ورودی- خروجی به‌ازای ورودی) از یک تابع ناشناخته و مجهول به فرم $\{(x1,y1), (x2,y2), \ldots, (xn,yn)\}$ داده شده باشد. تقریب زدن توابع یعنی پیدا کردن یک تخمین مناسب از تابع ناشناخته‌ی داده شده به شکلی که اگر ورودی دیگری غیر از این مجموعه به آن بدهیم، این تابع تقریبی جوابی نزدیک به تابع ناشناخته را به ما بدهد. دقت این تابع تقریبی می‌تواند با توجه به ورودی‌های آموزشی از قبل داده شده به آن و هم‌چنین پیچیدگی خود تابع، کم یا زیاد شود. شبکه عصبی یکی از قوی‌ترین ابزارها برای تقریب زدن توابع است. علاوه بر سیستم‌های هوشمند، از این خاصیت شبکه عصبی برای پیاده‌سازی کم‌هزینه‌تر توابع سنگین با بار پردازشی زیاد استفاده می‌شود.

نوع خاص این مسئله پیش‌بینی و تخمین است که اگر مقدار یک پدیده در n واحد زمانی اخیر داده شده باشد، مقدار نمونه بعدی که در زمان آینده یعنی n+1 خواهد آمد را پیش‌بینی کند.

بهینه‌سازی[۲۹]: طیف گسترده‌ای از مسائل موجود در علوم مهندسی، پزشکی و اقتصاد را می‌توان در قالب مسئله بهینه‌سازی مطرح کرد که سعی در بهینه کردن یک تابع هزینه/خطا دارند. شبکه‌های عصبی یکی از ابزارهای مفید برای حل این مسائل به شمار می‌روند.

۲-۴- شبکه‌های عصبی سخت‌افزاری در صنعت

یکی از شرکت‌هایی پیش‌رو در فناوری که در تلاش برای طراحی تراشه‌ای مانند مغز انسان می‌باشد، شرکت IBM است. این شرکت در قالب پروژه SyNAPSE تراشه‌هایی را تولید کرده است که تماماً شبکه عصبی بوده و برای کاربردهای نیازمند به یادگیری و هوش مصنوعی استفاده می‌شوند [۱]. به عنوان نمونه، تراشه TrueNorth یکی از محصولات این پروژه است. این تراشه ۴۰۹۶ هسته پردازشی را در قالب یک شبکه روی تراشه جای داده است. هر هسته پردازشی می‌تواند تا ۲۵۶ نورون را در خود جای دهد و بنابراین کل تراشه توانایی اجرایی کمی لیون نورون را دارد. این تراشه خاص منظوره فقط شامل شبکه عصبی است و به دلیل حذف سربار نرم‌افزار و با وجود آن‌که از بیش از ۵ میلیارد ترانزیستور ساخته شده است دارای توان مصرفی کمتر از یک وات است [۱].

شرکت گوگل نیز در اوایل سال ۲۰۱۶ از ساخت یک پردازنده خاص منظوره براساس شبکه‌های عصبی خبر داد [۲]. این پردازنده که TPU نام دارد در کنار پردازنده‌های اصلی در مراکز داده قرار گرفته و انجام پردازش‌های مربوط به هوش مصنوعی و تحلیل داده‌ها که توسط شبکه‌های عصبی به خوبی قابل پیاده‌سازی است را بر عهده دارد. همچنین شرکت مایکروسافت از تراشه‌های قابل پیکربندی مجدد (FPGA) برای پیاده‌سازی پردازنده‌های هوشمند بر پایه شبکه عصبی استفاده می‌کند. طبق گزارش‌های این شرکت، از این پردازنده‌ها در عملیات‌های مختلف تحلیل داده و نیز جستجوی هوشمند در موتور جستجوی Bing استفاده می‌شود [۳].

شرکت nVidia نیز که پیشگام طراحی و ساخت پردازنده‌های گرافیکی است قابلیت‌های سخت‌افزاری اجرای سریع شبکه‌های عصبی را در جدیدترین نسل پردازنده‌های خود، یعنی پردازنده‌های گرافیکی GP100 قرار داده است. طبق گزارش این شرکت، پردازنده جدید می‌تواند شبکه‌های عصبی را ۱۷ بار سریع‌تر از پردازنده نسل قبلی خود اجرا نماید [۴].

Zeroth یک پلاتفرم برای اجرای شبکه‌های عصبی در دستگاه‌های قابل حمل هوشمند (مثلا تبلت‌ها) بوده که شامل یک سخت‌افزار تخصصی اجرای شبکه عصبی و کتابخانه‌های نرم‌افزاری مربوطه است [۱۹]. این پلاتفرم محصول شرکت Qualcomm است که یکی از پیشروترین سازندگان پردازنده برای تلفن‌های هوشمند و تبلت‌ها به شمار می‌رود. این شرکت Zeroth را در آخرین نسل پردازنده‌های موبایل خود (Snapdargon 8x) و برای تشخیص چهره و نیز مدیریت هوشمند مصرف باتری دستگاه قرار داده است.

۳- کارهای پیشین

تاکنون، کارهای پژوهشی و صنعتی بسیار زیادی بر روی پیاده‌سازی سخت‌افزاری شبکه‌های عصبی هم به صورت دیجیتال و هم آنالوگ انجام شده است.

در این میان هدف بیشتر کارهای انجام شده، علاوه بر افزایش سرعت اجرای شبکه‌های عصبی، مدیریت ترافیک بین نورون‌ها، کاهش توان مصرفی، کاهش نیاز به پهنای باند حافظه بوده است.

در زمینه توان مصرفی، بیشتر کارهای موجود بر دو مشخصه مهم شبکه‌های عصبی تمرکز کرده‌اند: (۱) تحمل‌پذیری در برابر کاهش دقت و (۲) حجم زیاد محاسبات تکراری.

در [۲۰] وجود مقدار قابل توجهی عملیات ریاضی تکراری در شبکه‌های عصبی گزارش شده است که می‌توان آنها را جهت ذخیره انرژی حذف کرد. با توجه به این مشاهده، یک شتاب‌دهنده شبکه عصبی با نام CORN پیشنهاد شده است که از با استفاده مجدد از محاسبات گذشته به نورون‌ها اجازه می‌دهد تا نتایج محاسبات تکراری خود را با یکدیگر به اشتراک بگذارند. این استفاده مجدد از محاسبات به طور میانگین ۲۶٪ از انرژی شبکه‌های عصبی را در مقایسه با برخی طراحی‌های مدرن کم توان کاهش می‌دهد.

استفاده از ویژگی ذاتی شبکه‌های عصبی در تحمل‌پذیری در برابر کاهش دقت یکی دیگر از ابزارهای مهم برای کاهش توان مصرفی آن‌ها است. به طور کلی

روش‌های موجود در این دسته شامل روش‌های کاهش پهنای بیتی (دقت) اعداد در محاسبات عصبی [۲۱] و یا استفاده از واحدهای محاسبات تقریبی است [۲۲]. در برخی کارها نشان داده شده است که حتی با نصف کردن دقت بیتی اعداد (از اعداد ممیز ثابت ۱۶ بیتی به ۸ بیتی)، خطای خروجی شبکه عصبی برای بسیار از کاربردها قابل قبول است. با این وجود، بسیار از روش‌ها به دنبال کاهش هوشمندانه دقت بیتی هستند. مثلا، در [۲۱]، تاثیر کاهش دقت تک تک نورون‌ها در نتیجه خروجی شبکه عصبی اندازه‌گیری شده و پهنای بیتی هر نورون براساس حساسیت خروجی نسبت به آن تعیین می‌گردد.

پهنای باند حافظه یکی دیگر از ملاحظات اساسی در طراحی سخت‌افزاری شبکه‌های عصبی است [۲۳]. دلیل این امر آن است که یک شبکه عصبی تعداد زیادی داده ورودی و وزن را از حافظه واکشی می‌کنند که نیازمند پهنای باند زیادی از حافظه است. به طور خاص، یک شبکه کانولوشنی بزرگ نوعی که برای پردازش تصویر ساخته می‌شود (مانند پردازش‌هایی که در عینک هوشمند گوگل انجام می‌شود) نیاز به چند مگابایت حافظه جهت نگهداری وزن‌ها و ۳۰ تا ۶۰۰ هزار عملیات به ازای هر پیکسل از یک تصویر دارد [۲۳] که فراهم آوردن این پهنای باند مسئله‌ای چالش برانگیز برای یک سیستم نهفته است.

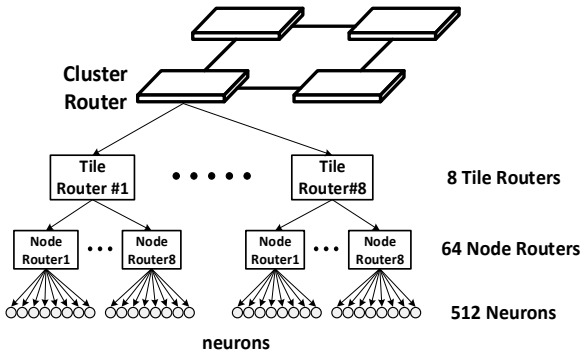
روش ارائه شده در [۲۴] یکی از شاخص‌ترین تلاش‌ها برای کاهش نیاز به پهنای باند در شبکه‌های عصبی هستند. در این مقالات یک شتاب‌دهنده برای شبکه‌های عصبی کانولوشنی در مقیاس بزرگ، با تأکید بر تأثیری که حافظه در کارایی و توان مصرفی طراحی شتاب‌دهنده‌ها دارد طراحی شده است. با بهینه‌سازی الگوی دسترسی به حافظه، این شتاب‌دهنده که با فنّاوری ۶۵ نانومتری پیاده‌سازی شده است قابلیت اجرای ۴۵۲ میلیون عملیات ممیز شناور در یک مساحت کوچک ۳ میلیمتر مربعی با مصرف انرژی ۴۸۵ میلی وات را دارد. سپس نشان داده شده است که روش ارائه شده نسبت به یک پردازنده مدرن برداری ۱۲۸ بیتی که با فرکانس ۲ گیگاهرتز کار می‌کند حدود ۱۱۷ برابر سریع‌تر بوده و ۲۱ برابر انرژی مصرفی کمتری دارد.

همان‌گونه که گفته شد، به دلیل ترافیک زیاد، شبکه ارتباطی بین واحدهای پردازشی در یک سخت‌افزار چندهسته‌ای شتاب‌دهنده شبکه عصبی نقش زیادی در کارآیی کلی سیستم دارد. از این رو طراحی مناسب این بخش مورد توجه پژوهشگران زیادی قرار گرفته است.

از همبندی توری دو بعدی در بسیاری از پیاده‌سازی‌های قبلی شبکه‌های عصبی در مقیاس صنعتی [۹] و پژوهشی [۷] استفاده شده است. در [۱۰] از شبکه میان‌ارتباطی چند مرحله‌ای Clos جهت مدیریت مؤثرتر ارتباطات میان نورونی در داخل تراشه استفاده شده است. در این مقاله، نشان داده شده است که به دلیل تشابه میان الگو ترافیک شبکه‌های عصبی (ترافیک بر پایه چندپخشی چند سطحی) و ویژگی‌های همبندی‌های میان‌ارتباطی چند مرحله‌ای، این همبندی‌ها گزینه مناسبی برای استفاده در شبکه‌های عصبی هستند. شبکه Clos، یکی از مهم‌ترین کلاس‌های همبندی‌های میان ارتباطی چند مرحله‌ای است و نتایج ارزیابی‌های مقاله نشان می‌دهد که می‌تواند ترافیک چندپخشی شبکه‌های عصبی را بهتر از توپولوژی توری که در بسیاری از پیاده‌سازی‌های شبکه عصبی مورد استفاده قرار گرفته است مدیریت کند و میانگین تأخیر کمتری را تولید می‌کند.

در پروژه‌ی بزرگ اروپایی EMBRACE، یک معماریِ مبتنی بر شبکه‌های روی تراشه‌ی سلسله مراتبی[۳۰] (H-NoC) برای شبکه‌های عصبی ارائه شده است [۸]. در شبکه‌های میان‌ارتباطیِ سلسله مراتبی، ترافیک شبکه به دو بخش ترافیک محلی (درون هر ناحیه) و ترافیک سرتاسری (بین ناحیه‌ها) تقسیم می‌شود و از یک همبندیِ ترکیبی به نام درخت- توری، برای بهره‌وریِ بهتر الگوریتم‌های چندپخشی سود می‌برد. بلوک‌های اصلی تشکیل‌دهنده‌ی هر تراشه در این پروژه، Cluster، Tile و Neuron نام دارند که در واقع لایه‌های مختلف سلسله مراتب این شبکه هستند. هر واحد Neuron میتواند ۱۰ نورون از شبکه عصبی را در خود جای دهد. سپس، در هر واحد Tile ۱۰ عدد واحد Neuron و در هر واحد Cluster ۴ عدد واحد Tile قرار می‌گیرند. در مجموع ۴۰۰ نورون در هر واحد Cluster قرار دارد که در قالب شبکه سلسله مراتبی به هم متصل هستند. شکل ۳ معماری قرارگیری خوشه‌ها را در همبندی درختی EMBRACE نشان می‌دهد [۸].

در این سیستم، برای میزبانی از شبکه‌های عصبی بزرگتر، واحدهای Cluster با همبندی توری مدور به همدیگر متصل می‌شوند.



شکل ۳- همبندی سلسله مراتبی درختی در EMBRACE [۸]

۴- معماری ارائه شده

بیشترین پیاده‌سازی‌های موجود شبکه روی تراشه موجود برای شبکه‌های عصبی از همبندی پایه یا تغییریافته‌ی توری استفاده می‌کنند. همبندی یک شبکه توری شامل یک ماتریس چند بعدی از گره‌ها است که توسط یک ساختار ارتباطی منظم، که هر گره را به طور مستقیم به گره‌های قبلی و بعدی خود در هر بعد متصل می‌کند، تشکیل شده است. با افزایش اندازه شبکه، میانگین تاخیر شبکه توری به طور قابل توجهی افزایش پیدا می‌کند که ناشی از عدم‌وجود مسیر مستقیم بین گره‌هایی است که در ساختار ماتریسی از هم فاصله زیادی دارند. به علاوه، مدل ترافیک شبکه‌های عصبی به صورت چندپخشی است که همبندی توری برای این نوع ترافیک مناسب نمی‌باشد. بنابراین استفاده از همبندی‌هایی با قابلیت ذاتی چندپخشی می‌تواند گزینه مناسب‌تری برای پیاده‌سازی ارتباط داخلی درون شبکه‌های عصبی باشد.

در این بخش، یک معماری سفارشی موازی برای شبکه‌های عصبی پیشنهاد می‌کنیم که از هم‌بندی میان ارتباطی dragonfly برای ارتباط بین نورون‌ها استفاده می‌کند.

همبندی میان ارتباطی dragonfly یکی از هم‌بندی‌های جدید شبکه‌های میان ارتباطی است که در ابر رایانه‌های مدرن سری XC محصول شرکت Cray به کار رفته است [۱۴]. ویژگی مهم این همبندی فراهم آوردن قطر کم با برقراری هوشمندانه‌تر اتصالات[۳۱] بین گره‌ها نسبت به توری است.

این همبندی در شکل ۴ نشان داده شده است. در این همبندی، گره‌های شبکه در قالب چندین گروه دسته‌بندی می‌شوند که تمام مسیریاب‌های داخل هر گروه به وسیله یک شبکه کاملا پیوسته[۳۲] با یکدیگر در ارتباط هستند. درجه هر مسیریاب a+p+h است که دارای p درگاه[۳۳] برای اتصال به پردازنده، a درگاه برای اتصال به سایرمسیریاب‌های هم‌گروهی، h درگاه برای اتصال به سایر مسیریاب‌ها در گروه‌های دیگر است. تعداد اتصالات بین گروه‌ها بسته به تعداد گروه‌ها (که می‌تواند متغیر باشد) است: بین هر دو گروه باید دست کم یک اتصال وجود داشته باشد و در صورت کم بودن گروه‌ها می‌توان چند اتصال بین هر دو گروه داشت. در شکل ۴، یک شبکه با a=3، p=2 و h=2 نشان داده شده و ۵ گروه از پردازنده‌ها را به هم متصل کرده است.

در این شکل، از آنجا که هر گروه دارای ۴ مسیریاب و هر مسیریاب دارای ۲ اتصال به بیرون گروه است (مجموعاً ۸ اتصال)، هر گروه می‌تواند ۲ اتصال به هریک از ۴ گروه دیگر در این شبکه داشته باشد.

در این همبندی، مسیریابی از هر گره به گره دیگر در سه گام انجام می‌پذیرد. در گام اول، بسته به آن گره‌ای در گروه خود ارسال می‌شود که به گروه مقصد متصل است. نظر به همبندی کاملا پیوسته داخل گروه، این کار به صورت بیشینه نیاز به پیمایش یک گام دارد. اگر خود گره مبدا مستقیماً به گروه مقصد وصل باشد، در این مرحله نیاز به حرکت بسته نیست. سپس بسته، اتصال بین گروه‌ها را پیموده و به یک مسیریاب در گروه مقصد می‌رسد و از آنجا با بیشینه یک گام به مقصد اصلی تحویل داده می‌شود.

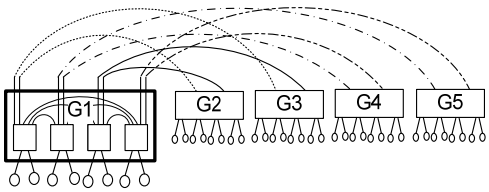
در این مقاله ما با تغییراتی این همبندی را برای پیاده‌سازی روی تراشه مناسب سازی کرده و یک مدل همه‌پخشی برای آن ارائه می‌کنیم.

اولین مشکل این همبندی برای پیاده‌سازی روی تراشه شبکه کاملا پیوسته درون هر گروه است که هم نیاز به سیم‌بندی زیاد و متقاطع دارد و هم تعداد درگاه‌ها و در نتیجه اندازه کراسبار و تعداد میانگیرها[۳۴] را افزایش می‌دهد.

برای رفع این مشکلات ما از یک گذرگاه مشترک[۳۵] برای اتصالات داخل گروه استفاده می‌کنیم. گذرگاه مشترک برای اتصال تعداد زیادی از عناصر به دلیل مقیاس ناپذیری توان و پهنای باند مناسب نیست. اما در این سیستم، ما با محدود کردن عناصر پردازشی داخل هر گروه، مانع از بروز مشکل مقیاس‌پذیری گذرگاه مشترک می‌شویم.

در این سیستم با فرض داشتن ساده‌ترین حالت سیستم، یعنی p=1 و h=1، هر مسیریاب دارای یک درگاه برای اتصال به پردازنده، یک درگاه گذرگاه مشترک برای اتصال به سایر مسیریاب‌های هم‌گروهی، و یک درگاه برای اتصال به سایر مسیریاب‌ها در گروه‌های دیگر است. این ساختار ساده، مساحت و نیز توان مصرفی شبکه روی تراشه تا حد زیادی کاهش می‌دهد. می‌توان با افزایش پارامترهای این همبندی، یعنی a، p، و h، مصالحه‌ای بین مساحت و برون‌دهی شبکه ایجاد کرد. ما در این مقاله فقط همبندی پایه با p=1، h=1، و a=1 را ارزیابی کرده و ارزیابی این مصالحه را به کارهای بعدی موکول می‌کنیم.

در این همبندی می‌توان با یک زمانبندی ایستا، همانگونه که در فصل‌های قبل به آن اشاره شد، بخش کنترلی مسیریاب‌ها را نیز بسیار ساده کرده و با یک مسیریاب ساده به برون‌دهی قابل قبول رسید.



شکل ۴- همبندی dragonfly

همبندی میان ارتباطی dragonfly به کار رفته در ابر رایانه Cray یک نمونه از شبکه‌های درجه بالا[۳۶] است. در همبندی‌های میان ارتباطی درجه بالا، مسیریاب‌ها دارای درگاه‌های بیشتری نسبت به شبکه‌هایی درجه پایین (مانند توری) هستند، اما پهنای بیتی درگاه‌ها کمتر است. درگاه‌های بیشتر در شبکه‌های درجه بالا، منجر به تولید قطر کمتر می‌شود و از این رو، این شبکه‌ها قابلیت مقیاس‌پذیری بهتری از نظر تاخیر زمانی دارند. اما از سوی دیگر، زمانی که پهنای بیتی اتصال کمتر از عرض بسته باشد تعداد فلیت‌های یک بسته و در نتیجه تاخیر ارسال بیشتر می‌شود. اما این مشکل در شبکه‌های عصبی که دارای بسته‌های ذاتی کوچکی هستند مشکلی محسوب نمی‌شود: بسته‌های تولید شده توسط شبکه عصبی حامل خروجی نورون‌ها است که یک عدد اعشاری است ۱۶ یا ۳۲ یا بیتی به همراه چند بیت کنترلی است که در برابر بسته‌های ۵۱۲ بیتی (شامل یک بلوک حافظه) در بسیاری از سیستم‌های چندپردازنده‌ای روی تراشه بسیار کوچک‌تر است. اما در پیاده‌سازی خاص ما از این همبندی، با جایگزینی بخشی از همبندی با یک گذرگاه مشترک، ضمن سود بردن از قطر کم، مشکل نیاز به درگاه‌های زیاد هم حل می‌شود.

۴-۱- نگاشت[۳۷] شبکه‌های عصبی روی dragonfly

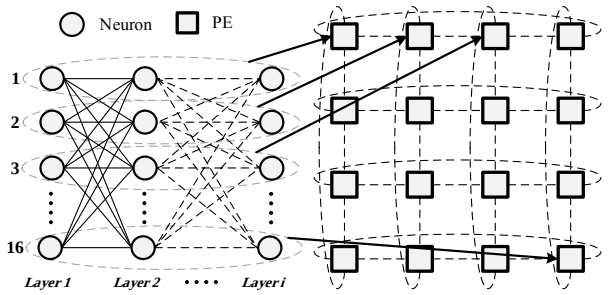
در این بخش، از آنجا که نحوه نگاشت نورون‌ها تاثیر مستقیمی بر زمان‌بندی شبکه دارد، ابتدا یک روش برای نگاشت نورون‌ها بر رویگره‌های هم‌بندی dragonfly ارائه کرده و سپس یک زمان‌بندی مناسب برای ارتباطات بین نورون‌ها ایجاد می‌کنیم.

از آنجایی که بسیار محتمل است که در یک شتاب‌دهنده‌ی سخت‌افزاری، تعداد نورون‌ها بسیار بیشتر از تعداد هسته‌های پردازشی باشد، لازم است تا با تکنیکی مناسب، نورون‌ها دسته‌بندی شده و هر دسته روی یک هسته‌ی پردازشی نگاشته شود. در بسیاری از کارهای قبلی، نورون‌های یک لایه را در یک گره یا خوشه‌ای از گره‌ها قرار می‌دادند. در این حالت، برخی از گره‌ها که فقط شامل نورون‌های لایه‌ی ورودی هستند، فقط فرستنده بوده و برخی دیگر که نورون‌های لایه‌های میانی را در بردارند، در یک فاز فرستنده (به لایه بعدی) و در فاز دیگر گیرنده (از لایه قبلی) خواهند بود. لذا بار ترافیکیِ شبکه به طور متعادل در تمام گره‌ها پخش نخواهد شد.

برای رفع این مشکل در معماری پیشنهادی، ما نورون‌هایی از لایه‌های مختلف را در یک دسته قرار می‌دهیم. با استفاده از این تکنیک، هر دسته شامل نورون‌هایی از تمام لایه‌های شبکه عصبی خواهد بود. بنابراین، اگر هر دسته روی یک هسته‌ی پردازشی نگاشت داده شود، آن هسته‌ی پردازشی در هر لحظه می‌بایست هم داده‌های تولید شده از نورون‌های خود را ارسال و هم داده‌های رسیده از دسته‌های دیگر (دیگر هسته‌های پردازشی) را دریافت کند، در نتیجه در یک زمان، هم فرستنده و هم‌گیرنده خواهد بود. در این مدل نگاشت، تمام گره‌ها یکسان بوده و کار یکسان و مشابه‌ای انجام می‌دهند.

همان‌طور که در شکل ۵ نمایش داده شده است، تکنیک دسته‌بندی بر روی یک شبکه‌ی چندلایه‌ی که هر لایه از آن ۱۶ نورون دارد، اعمال شده و دسته‌ها به طور منظم روی عناصر پردازشیِ یک شبکه روی تراشه با هم‌بندیِ توری نگاشت داده شده‌اند.

در این مدل نگاشت، لزومی بر برابری تعداد نورون‌های لایه‌ها وجود ندارد و اگر تعداد نورون‌های لایه‌های ورودی و خروجی کمتر از لایه‌های میانی باشد (که معمولا هست) می‌توان تعداد مختلفی از نورون‌های لایه در هر دسته داشت. همچنین می‌توان در صورت لزوم چند دسته را در یک گره شبکه روی تراشه جای داد.



شکل ۵- دسته‌بندی نورون‌های برای نگاشت بر روی شبکه روی تراشه

با این کار، بار کاریِ تمام هسته‌های پردازشی یکنواخت خواهد بود و در نتیجه ترافیک تولید شده در شبکه به صورت متعادل در سرتاسر شبکه توزیع خواهد شد. ارتباطات بین نورون‌ها در این مدل تبدیل به یک مدل ترافیکی ویژه می‌شود که در آن هر گره داده‌های خود را به همه گره‌های دیگر ارسال می‌کند و از تمام گره‌های دیگر داده دریافت می‌نماید.

۴-۲- زمان‌بندی ارتباطات شبکه‌های عصبی روی dragonfly

در این بخش، یک زمان‌بندی ایستای برای ارتباطات بین نورون‌ها در شبکه روی تراشه dragonfly طراحی می‌گردد.

در این روش با استفاده از زمان‌بندی ایستا تمام اجزای هوشمند مسیریاب‌ها، مانند واحدهای مسیریابی و داوری و کنترل جریان و نیز میان‌گیرها حذف شده و در نتیجه، توان، تاخیر، و مساحت مرتبط به آنها از مسیریاب حذف می‌گردد. در عوض، مسیری که بسته‌ها در هر مسیریاب طی می‌کنند از قبل تعیین شده و در قالب یک جدول اتصال ورودی- خروجی در مسیریاب‌ها قرار می‌گیرد. این جدول تعیین می‌کند که در هر سیکل، کدام ورودی به کدام خروجی متصل باشد تا ارتباطات در نظر گرفته شده برقرار گردد.

با اعمال روش دسته‌بندی و نگاشت نورون‌ها، هر عنصر پردازشی در هر لحظه باید خروجی نورون‌های خود را برای دیگر نورون‌ها ارسال کند و متقابلاً، بقیه‌ی عناصر پردازشیِ شبکه نیزخروجی خود را برای نورون‌های آن عنصر می‌فرستند.

این زمان‌بندی در چند سیکل انجام می‌شود و در پایان آن، تمام گره‌ها داده تمام گره‌های دیگر را دارند. در ادامه این روال را برای همه پخشی داده گره شماره ۱ هر گروه در یک شبکه dragonfly با پنج گروه چهار گره‌ای دنبال می‌کنیم.

- سیکل ۱: ابتدا گره شماره ۱ در هر گروه داده خود را از طریق گذرگاه مشترک به تمام گره‌های دیگر گروه ارسال می‌نماید. در پایان این سیکل تمام گره‌های هر گروه داده گره شماره ۱ را خواهند داشت.
- سیکل ۲: تمام گره‌های هر گروه از طریق اتصال خارجی خود (که به یک گروه دیگر وصل است) داده‌ی گره شماره ۱ گروه خود را برای گروه دیگری که به آن متصل هستند ارسال می‌نماید. برای مثال، اگر به گروه شماره ۱ توجه کنیم، گره‌های ۱ تا ۴ داده‌ی گره ۱ گروه خود را که در سیکل قبل دریافت کرده‌اند را به ترتیب برای گره مربوطه (گره شماره ۱، همان‌طور که در شکل ۴ مشاهده می‌شود) در گروه‌های ۱ تا ۴ ارسال می‌دارند. در پایان این سیکل، اگر مجددا به گروه شماره ۱ توجه کنیم، گره‌های ۱ تا ۴ این گروه، داده‌ی گره ۱گروه‌های به ترتیب ۱ تا ۴ را دارند.
- سیکل‌های ۳ تا ۶: گره‌های شماره ۱ تا ۴ در هر گروه داده دریافتی از گروه دیگر را از طریق گذرگاه مشترک به تمام گره‌های دیگر گروه خود ارسال می‌نماید. در پایان این سیکل تمام گره‌های هر گروه داده گره شماره ۱ تمام گروه‌های دیگر را خواهند داشت.

روال بالا، داده گره ۱ تمام گروه‌ها را در شش سیکل به تمام شبکه ارسال می‌دارد. با تکرار این روال برای چهار دفعه (در دفعه i، داده گره iام هر گروه)، می‌توان داده تمام گره‌ها را به تمام گره‌های دیگر رسانید.

۵- ارزیابی

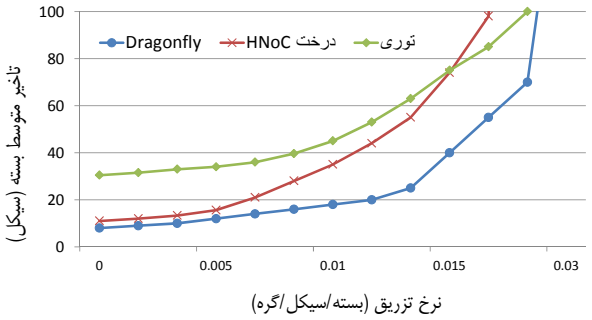
جهت ارزیابی معماری‌های ارائه شده، از چندین مجموعه داده‌ی معروف که مربوط به زمینه‌ی یادگیری ماشین است، به‌عنوان محک جهت ارزیابی استفاده می‌کنیم. برای مشخصات محک‌ها نیز ازکارهای قبلی [۱۰] [۲۵] بهره می‌گیریم. ساختار شبکه عصبی متناسب با هر محک را در جدول ۱ آورده شده است. لازم به ذکر است که شبکه به‌دست آمده برای هر برنامه جهت رسیدن به کمترین خطای خروجی با کوچک‌ترین تعداد نورون، پس از اجراها و جستجوهای مختلف در نظر گرفته می‌شود. برای ساخت و آموزش شبکه عصبی نیز، از نرم‌افزار MATLAB استفاده می‌کنیم.

جدول ۱- برنامه‌های محک و داده‌های ورودی

برنامه محک	ساختار لایه‌ها
Performance Modeling and Evaluation	128:256:128
Object Classification	3072:3000:10
Census Data Analysis	14:12:12:2
Hand Writing Digit Recognition	784:700:10

برای ارزیابی، علاوه بر هم‌بندی dragonfly، هم‌بندی درختی H-Noc در EMBRACE [۸] و همبندی توری را جهت انجام مقایسه در نظر می‌گیریم. برای هر سه شبکه، ۱۲۸ گره (مسیریاب) در نظر گرفته شده است. تمام درگاه‌های ورودی مسیریاب‌ها مجهز به میان‌گیرهای ۸ فیلیتی هستند. در شبکه‌های مورد مقایسه (توری و درخت) تمام مسیریاب‌ها بسته‌ها را بعد از انجام عمل ذخیره کردن در میانگیر، مسیریابی، و داوری در سه سیکل به گره بعد ارسال می‌کنند، اما در صورت بازنده شدن بسته‌ها در داوری، این زمان افزایش خواهد یافت.

در هم‌بندی dragonfly، شبکه به ۱۶ گروه ۸ گرهی تقسیم می‌شود. هر مسیریاب یک درگاه به پردازنده، یک درگاه به گذرگاه مشترک (برای ارتباطات درون‌گروهی) و دو درگاه برای ارتباط با سایر گروه‌ها دارد. در هم‌بندی درخت، ۸ واحد پردازشی به یک مسیریاب نورون، ۸ مسیریاب نورون به یک مسیریاب tile، و دو مسیریاب tile به وسیله یک مسیریاب خوشه متصل شده‌اند. در توری، ۴ گره به یک مسیریاب متصل هستند که یک توری ۴×۸ را تشکیل می‌دهند.



شکل ۶- میانگین تاخیر همه‌پخشی توری،dragonfly، و درخت

در آزمایش‌های انجام شده، ما فرض کرده‌ایم که گره‌های پردازشی عملیات ریاضی بر روی یک ورودی را به اندازه‌ای سریع انجام می‌دهند که تمام داده دریافتی به‌وسیله شبکه بدون تأخیر صف و در یک سیکل انجام می‌شود.

شکل ۶ کارایی همه‌پخشی سه شبکه با نشان دادن میانگین زمان تأخیر همه پخشی آن‌ها در ترافیک یکنواخت را مقایسه کرده است. تأخیر همه‌پخشی یک بسته از زمان تولید شدن تا زمانی که بسته در آخرین مقصد دریافت می‌شود است. همان‌گونه که شکل نشان می‌دهد، قطر کم شبکه dragonfly منجر به تأخیر کمتری برای همه‌پخشی در شبکه می‌شود. در این نمودار، بدترین عملکرد متعلق به شبکه توری است که بیشترین قطر را دارد.

پس از بررسی تحت ترافیک ساختگی[۳۸]، شبکه پیشنهادی را بر روی شبکه‌های عصبی واقعی لیست شده در جدول ۱ مورد ارزیابی قرار دادیم. ما تمامی محک‌ها را بر روی سه شبکه مورد بحث با ۱۲۸ گره پیاده‌سازی کردیم. در مورد برخی از شبکه‌های عصبی، لازم می‌شود که چندین گروه نورون را بر روی یک گره شبکه

نگاشت کرد که در این مورد از روش نگاشت اشاره شده در بخش قبل استفاده شده است.

جدول ۲ کارایی این سه هم‌بندی را بر حسب توان عملیاتی مورد مقایسه قرار داده است. ما توان عملیاتی را به عنوان نرخ ورودی ای از شبکه‌های عصبی (تعداد ورودی‌های شبکه عصبی در هر سیکل) تعریف می‌کنیم در آن نرخ، شبکه روی تراشه می‌تواند با تاخیر زیر ۱۰۰ سیکل در حال کار باشد. با افزایش نرخ ورودی از این مقدار، شبکه با افزایش تاخیر مواجه شده و به اشباع می‌رود (از کار می‌افتد). لذا معیار تعریف شده در این بخش برای توان عملیاتی، بیشینه توان شبکه روی تراشه برای پذیرفتن داده جدید است و معیاری مناسب برای مقایسه شبکه‌ها به شمار می‌رود.

برای مقایسه بهتر، اعداد جدول ۲ براساس نتایج به دست آمده از توری نرمال‌سازی شده است. همانطور که جدول ۲ نشان می‌دهد، شبکه ارائه شده باز هم به دلیل قطر کمتر و مسیریاب‌های سریع‌تر توان عملیاتی بیشتری نسبت به شبکه‌های مورد مقایسه ارائه می‌دهد. در این جدول نشان داده شده است که توان عملیاتی هم‌بندی ارائه شده به ترتیب بیش از ۸۹ و ۴۵ درصد بالاتر از توری و درخت است.

جدول ۲- مقایسه توان عملیاتی هم‌بندی‌های در نظرگرفته شده

برنامه محک	توان عملیاتی نرمال‌سازی شده (ورودی در هر سیکل)		
	dragonfly	درخت	توری
Performance Modeling	1.7	1.3	1
Object Classification	2.4	2.0	1
Census Data Analysis	1.4	1.1	1
Hand Writing Digit Recognition	1.9	1.5	1

ارزیابی توان مصرفی برای یکی از محک‌های نمونه (Object Classification) توان مصرفی ۸۱۲ میلی وات برای درخت، ۹۴۵ میلی وات برای توری، و ۵۸۰ میلی‌وات برای dragonfly را نشان می‌دهد. همان‌گونه که بحث شد، قطر کم و مسیریاب‌های ساده دلیل برتری dragonfly بر شبکه‌های دیگر از نظر مصرف توان است. اعداد توان فقط شامل توان ارتباطات است و توسط ابزار تخمین توانDSENT [۲۶] در فناوری ۴۵ نانومتر به‌دست آمده است.

۶- نتیجه‌گیری و کارهای آینده

در ساله‌ای اخیر حجم قابل‌توجهی از پژوهش بر روی پیاده‌سازی سخت‌افزاری شبکه‌های عصبی انجام شده است. دلیل عمده این توجه نیاز به وجود هوشمندی در بسیاری از سیستم‌های کامپیوتری، از یک سیستم نهفته کنترلی کوچک گرفته تا سرویس‌دهنده‌های بزرگ در مراکز داده، است. شبکه‌های عصبی یکی از کاراترین روش‌ها در پیاده‌سازی سیستم‌های دارای یادگیری و هوشمندی هستند. سرعت انجام محاسبات و توان مصرفی در بسیاری از این سیستم‌ها یک پارامتر محدود کننده است، بنابراین پیاده‌سازی سخت‌افزاری شبکه‌های عصبی با سفارشی‌سازی ساختار سخت‌افزار و حذف سربار نرم‌افزار سهم به‌سزایی در بهینه‌سازی توان و تاخیر انجام محاسبات در این مدل محاسباتی دارد. با توجه به اهمیت ارتباطات در پیاده‌سازی سخت‌افزاری شبکه‌های عصبی، در این مقاله یک شبکه روی تراشه با هم‌بندی dragonfly برای این سیستم‌ها ارائه کردیم. این شبکه دارای قابلیت مناسبی برای پیاده‌سازی همه‌پخشی است و با ارائه یک زمان‌بندی ایستا دارای مسیریاب‌هایی تا حد ممکن ساده است. نتایج ارزیابی نشان‌دهنده کاهش قابل توجه توان مصرفی و افزایش کارآیی نسبت به شبکه‌های روی تراشه ارائه شده قبلی است.

یکی از راه‌ها برای ادامه این کار، یافتن یک چینش مناسب در سطح مدار برای این شبکه است تا بتوان آن را با مساحت قابل قبول در سطح تراشه پیاده‌سازی کرد. طرح جاری برای یک شتاب‌دهنده ساخته شده است که با توجه پیش‌بینی‌پذیری ارتباطات در این سیستم‌ها می‌توان با انجام زمان‌بندی ایستا از پیچیدگی مسیریاب‌ها کاست. اما استفاده از این هم‌بندی به عنوان یک زیرساخت کلی با قابلیت همه‌پخشی مناسب برای اجرای بدون زمان‌بندی شبکه‌های عصبی و نیز یافتن مدل‌هایی با کارآیی بالاتر برای نگاشت نورون‌ها بر روی گره‌های شبکه از دیگر راه‌ها برای ادامه پژوهش جاری است.

مراجع

[1] IBM SyNAPSE project, http://www.research.ibm.com, Jan. 2015.

[2] Tensor Processing Unit Architecture, https://cloudplatform.googleblog.com, Jan. 2015.

[3] Microsoft to Accelerate Bing Search with Neural Network, http://blog.microsoft.com, Jan. 2015.

[4] GP100 Pascal Whitepaper, http://www.nvidia.com, Jan. 2015.

[5] J. Hauswald, and et. al., "DjiNN and Tonic: DNN as a service and its implications for future warehouse scale computers," Proc. *International Symposium on Computer Architecture*, 2015.

[6] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for general-purpose approximate programs," *Proc. International Symposium on Microarchitecture*, pp. 449–460, 2012.

[7] D. Vainbrand, and R. Ginosar, "Network-on-chip architectures for neural networks," *Proc. Network-on-chip Symposium*, 2010.

[8] S. Carrillo, and et. al., "Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 45, no. 22, 2012.

[9] E. Painkras, and et. al., "SpiNNaker: A 1-W 18-Core System-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, pp. 1943-1953, 2013.

[10] A. Yasoubi, R. Hojabr, H. Takshi, M. Modarressi, and M. Daneshtalab, "CuPAN: high throughput on-chip interconnection for neural networks," *Proc. International Conference of Neural Information Processing*, 2015.

[11] D. Y. Kim, and et. al., "A neural network accelerator for mobile application processors," in *IEEE Transactions on Consumer Electronics*, vol. 61, no. 4, pp. 555-563, 2015.

[12] W. J. Dally, and B. Towles, *Principles and practices of interconnection networks*, Morgan-Kaufmann Publishers, 2004.

[13] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," *Proc. International Symposium on Computer Architecture*, Beijing, pp. 77-88, 2008.

[14] B. Alverson, "Cray high speed net working," *Proc. 20th Annual Symposium on High-Performance Interconnects (HOTI)*, 2012.

[15] S. Haykin, *Neural networks: A comprehensive foundation*, Upper Saddle River, NJ, USA: Prentice-Hall, 2008.

[16] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Journal of Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[17] W. Maass, and C. M. Bishop, *Pulsed neural networks*. MIT press, 2001.

[18] P. Merolla, and et. al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668-673, 2014.

[19] https://www.qualcomm.com/invention/cognitive-technologies/machine-learning, Jan. 2015.

[20] A. Yasoubi, R. Hojabr, and M. Modarressi, "Power-efficient accelerator design for neural networks using computation reuse," in *IEEE Computer Architecture Letters*, 2015.

[21] Q. Zhang, T. Wang, Y. Tian, F. Yuan, and Q. Xu, "ApproxANN: an approximate computing framework for artificial neural network," *Proc. Design, Automation & Test in Europe Conference*, 2015.

[22] S. Venkataramani, A. Ranjan, K. Roy, and A. Raghunathan, "Axnn: Energy-efficient neuromorphic systems using approximate computing," *Proc. International Symposium on Low Power Electronics and Design*, 2014.

[23] Y. Chen, and et. al., "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," *Proc. ISSCC*, pp. 262-263, 2016.

[24] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "A high-throughput neural network accelerator," *IEEE Micro*, vol. 35, no. 3, pp. 24-32, 2015.

[25] A Firuzan, M. Modarressi, and M. Daneshtalab, "A reconfigurable network-on-chip for efficient implementation of neural networks," *Proc. International Symposium on Reconfigurable Communication-centric Systems-on-Chip*, 2015.

[26] C. Sun, and et. al., "DSENT: A tool connecting emerging photonics with electroniccs for opto-electronic networks-on-chip modeling," *Proc. Network-on-chip Symposium*, 2012.

نسرین اکبری دانشجوی کارشناسی‌ارشد رشته مهندسی کامپیوتر گرایش معماری سیستم‌های کامپیوتری از دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران است. ایشان دوره کارشناسی خود را در سال ۱۳۹۴ در رشته مهندسی کامپیوتر- سخت‌افزار از دانشگاه صنعتی خواجه نصیرالدین طوسی به اتمام رسانیده‌اند. شبکه‌های روی تراشه، مالتی‌مدیا بر روی شبکه، و بیوانفورماتیک از زمینه‌های تحقیقاتی مورد علاقه ایشان است.
آدرس پست الکترونیکی ایشان عبارت است از:
nasrin.akbari@ut.ac.ir

بیتا دبیری دانشجوی کارشناسی‌ارشد رشته مهندسی کامپیوتر گرایش معماری سیستم‌های کامپیوتری از دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران است. ایشان دوره کارشناسی خود را در سال ۱۳۹۳ در رشته مهندسی کامپیوتر-نرم‌افزار از دانشگاه شهید رجایی به اتمام رسانیده‌اند. شبکه‌های روی تراشه، شبکه‌های کامپیوتری و سیستم‌های نهفته بی‌درنگ از زمینه‌های تحقیقاتی مورد علاقه ایشان است.
آدرس پست‌الکترونیکی ایشان عبارت است از:
bita.dabiri@ut.ac.ir

مهدی مدرسی مدرک کارشناسی خود را در سال ۱۳۸۲ از دانشگاه صنعتی امیرکبیر و مدرک‌های کارشناسی‌ارشد و دکترا را در سال‌های ۱۳۸۴ و ۱۳۸۹ در مهندسی کامپیوتر از دانشگاه صنعتی شریف دریافت کرده است. وی از سال ۱۳۹۱ عضو هیئت علمی گروه معماری سیستم‌های کامپیوتری در دانشکده مهندسی برق و کامپیوتر دانشگاه تهران است. وی همچنین به عنوان پژوهشگر در دانشگاه پلی‌تکنیک لوزان (EPFL) در سوییس (۲۰۰۹ تا ۲۰۱۰) و پژوهشگر غیرمقیم در پژوهشگاه دانش‌های بنیادی (IPM) (از ۱۳۸۵ تا کنون) فعالیت کرده است. زمینه‌های پژوهش مورد علاقه ایشان معماری کامپیوتر، شبکه روی تراشه، پردازش موازی، و سخت‌افزارهای یادگیر است که تاکنون بیش از ۷۰ مقاله درباره آنها منتشر کرده است.
آدرس پست‌الکترونیکی ایشان عبارت است از:
modarressi@ut.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۷/۰۹
تاریخ اصلاح: ۱۳۹۴/۰۹/۱۳
تاریخ قبول شدن: ۱۳۹۴/۰۹/۳۰
نویسنده مرتبط: دکتر مهدی مدرسی، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران.

[1]Neural Network
[2]Embedded System
[3]Hardware Accelerator
[4]Many-Core Chip
[5]Throughput
[6]Network-on-Chip
[7]Packet-Switch

[8]Topology
[9]Tree
[10]Mesh
[11]Static Scheduling
[12]Circuit Switching
[13]Router
[14]Diameter
[15]Multicast
[16]Multi-Layer Perceptron
[17]Feed-Forward
[18]Activation Function
[19]Sigmoid
[20]Lookup Table
[21]Approximate
[22]Training
[23]Supervised Learning
[24]Offline
[25]Spiking Neural Networks
[26]Pattern Recognition and Classification
[27]Clustering
[28]Fitting
[29]Optimization
[30]Hierarchical
[31]Link
[32]Fully Connected
[33]Port
[34]Buffer
[35]Bus
[36]High-Radix
[37]Mapping
[38]Synthetic

راهنمای تهیه مقالات

نشریه علوم رایانش و فناوری اطلاعات

نسخه نهائی مقالات ارسالی برای چاپ در نشریه "علوم رایانش و فناوری اطلاعات" باید بر طبق اصول مطرح شده در این راهنما تهیه شده باشد. رعایت این اصول در نسخه اولیه نیز قویاً توصیه می‌شود. مقالات به زبان فارسی ارسال گردد.

۱- ساختار مقاله

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

۲- معادله‌ها، شکل‌ها، جدول‌ها، و عکس‌ها

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

۳- نحوه نگارش مراجع

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

[۱] ب. مقدم، ا. تقوی، و ن. طاهری، آشنائی با شبکه‌های کامپیوتری، چاپ دوم، انتشارات نصر، تهران، ۱۳۷۵.
[۲] ی. براون، مقدمه‌ای بر شبکه‌های عصبی، ترجمه م. ع. آرام، انتشارات فجر، مشهد، ۱۳۷۰.
[۳] راهنمای کاربران حسابرس، شرکت پردازش رایانه‌ای ایران، تهران، ۱۳۶۵.

[۴] ج. عارف، استنتاج فازی بوسیله شبکه‌های عصبی، پایان‌نامه کارشناسی ارشد هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شهر، ۱۳۷۴.

[۵] ج. حسینی، و ح. ربانی، "تشخیص چهره انسان در تصویر،" نشریه امیرکبیر، سال هشتم، شماره ۴۲، ص ۱۲۵-۱۴۷، ۱۳۷۷.

[۶] ج. حسینی، و ح. ربانی، "تشخیص چهره انسان در تصویر،" در مجموعه مقالات هفتمین کنفرانس سالانه انجمن کامپیوتر ایران، ص ۲۲۴-۲۳۲، ۱۳۸۰.

[7] M. A. Ahmadi, and M. H. Rahimi, *Fuzzy Set Theory*, New Jersey: Prentice-Hall, 1995.

[8] M. A. Ahmadi, M. H. Rahimi, and A. Fatemi, ″Evidence-Based Recognition of 3D Objects,″ *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 12, no. 10, pp. 811-835, 1994.

[9] A. Taheri, ″On-Line Fingerprint Verification,″ *Proc. IEEE Intl Conf. Pattern Recognition*, pp. 752-758, 1992.

[10] M. A. Ahmadi, *On-line Fingerprint Verification*, Ph. D. Dissertation, MIT, Cambridge, MA, 1982.

[11] A. J. Washington, ″The Fingerprint of MalcomX,″ http://www.dermatoglyphics.com, June 2003.

[12] International Biometrics Group, http://www.biometricgroup.com, May 2003.

۴- واژه‌نامه

هر واژه خارجی در واژه‌نامه **در انتهای مقاله** با شماره‌ای مشخص شود و شماره در معادل فارسی آن واژه در متن، بصورت بالانویس آورده شود.

۵- آماده‌سازی مقاله

- مقاله را با نرم‌افزار Word تایپ نمائید.
- متن چکیده به صورت تک ستونی با طول سطر ۱۸ سانتیمتر و متن مقاله به صورت دو ستونی با طول هر ستون ۸۷ میلیمتر و فاصله دو ستون ۶ میلیمتر تایپ شود. حاشیه‌ها از بالا و پائین برابر ۲۰ میلیمتر و از طرفین برابر ۱۵ میلیمتر اختیار شود.
- فاصله عنوان مقاله در صفحه اول از بالای صفحه برابر ۸۵ میلیمتر باشد و عنوان وسط چین شود.
- کلیه عناوین بصورت پررنگ با قلم **"B Nazanin"** تایپ شوند، اندازه قلم عنوان مقاله ۱۸، عناوین سطح اول ۱۴، و عناوین سطح دوم و سوم ۱۲ انتخاب شوند.
- متن چکیده‌ها با قلم **"B Nazanin"** اندازه ۹، متن مقاله با قلم **"B Nazanin"** اندازه ۱۰، و کلمات و متن انگلیسی با قلم **Times New Roman** اندازه ۹ تایپ شوند.
- تمام متن بصورت تک فاصله تایپ شود. اسامی نویسندگان از عنوان مقاله و اسامی نویسندگان از عناوین محل اشتغال نویسندگان دو خط فاصله داشته باشد. بالای هر عنوان یک سطر فاصله قرار داده شود.
- سعی شود تعداد صفحات مقاله از ۳۰ صفحه بیشتر نباشد.

۶- نحوه ارسال مقاله

- ارسال مقاله فقط از طریق ایمیل مجله (**csitjour@gmail.com**) انجام شود.
- **مقاله ارسالی برای نشریه علوم رایانش و فناوری اطلاعات نباید در جای دیگری به چاپ رسیده باشد و یا در زمان بررسی توسط نشریه برای چاپ به نشریه دیگری ارسال گردد.**
- پس از قبول مقاله، نسخه نهائی تصحیح شده مقاله باید در قالب‌های Word و PDF به نشریه ارسال گردد.
- در نسخه نهائی باید بیوگرافی کلیه نویسندگان (به زبان فارسی) و عکس آنها در انتهای مقاله قرار داده شود، همچنین عنوان مقاله، نام نویسندگان، مؤسسه محل اشتغال، چکیده، کلمات کلیدی به زبان انگلسی در فایل جداگانه ارسال شود.

The CSI Journal on Computing Science and Information Technology

The CSI Journal on
Computing Science and
Information Technology
Vol. 13, No. 2, 2016

A High-Performance and Low-Power Network-on-Chip Architecture for Neural Networks

Nasrin Akbari **Bita Dabiri** **Mehdi Modarressi**

School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

ABSTRACT

Emerging neural network accelerators are often implemented as a many-core chip and rely on a network-on-chip to handle the huge amount of inter-neuron traffic. The well-known mesh and tree are the most popular topologies in prior many-core neural network implementations and research proposals. However, these conventional topologies suffer from high diameter, low bisection bandwidth, and poor collective communication support. In this paper, we present a customized version of the Dragonfly topology for Neural Networks. The capability of dragonfly to support multicast and broadcast traffic in a simple and efficient way, as well as its low diameter, is the major motivation behind proposing a customized version of this topology as the communication infrastructure of neural network accelerators. We also apply a conflict-free static scheduling for neurons to send their data to the network, thereby enable the network to use very simple circuit-switched routers to further improve power/performance profile. We compare Dragonfly with some state-of-the-art NoC topologies adopted in recent neural network hardware accelerators and show that it yields lower average message hop count and higher throughput.

Keywords: Network-on-Chip, Neural Network, Circuit-Switching, Low-Power, Dragonfly Topology.

The CSI Journal on Computing Science and Information Technology

The CSI Journal on
Computing Science and
Information Technology
Vol. 13, No. 2, 2016

Cross-Lingual Opinion Mining using Semantic Features

Shima Esmaeili Taft **Azadeh Shakery**

School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

ABSTRACT

Opinion mining or sentiment analysis is a subtask of text mining that analyzes the sentiment orientation of subjective documents. Both supervised and unsupervised methods have been proposed in the literature for this task. Supervised methods generally perform better than unsupervised methods, but they require a large set of labeled training dataset in the same domain and language of the test dataset. Creating a large training dataset is costly, and thus it is desired to make use of available datasets in one language to train the model in another one. Obviously using the available dataset directly won't have the desired result, so how to transfer the information from the source language to the target language is the challenge. In this paper we propose a cross-lingual opinion mining method which makes use of the available training data in one language to build a classifier and classify new documents in another language. To this end, a bilingual dictionary is used to overcome the language barrier, which is an available translation resource even in resource lean languages. The proposed method suggests dividing the features of both languages into two categories; pivot features and non-pivot features. Then using an unlabeled opinion dataset in both languages, a bipartite graph between these two categories of features is constructed. Bilingual semantic features are extracted by clustering this graph and documents in both languages are transferred into a unified semantic space. Experiment results on an English-German dataset show the significantly better performance of the proposed method compared to other cross-lingual methods.

Keywords: Opinion Mining, Sentiment Analysis, Pivot Feature, Semantic Feature, Cross-Lingual, Domain-Independent Feature, Domain-Specific Feature, Bipartite Graph, Classification.

The CSI Journal on Computing Science and Information Technology

The CSI Journal on
Computing Science and
Information Technology
Vol. 13, No. 2, 2016

A New Energy-Aware Mapping and Scheduling Algorithm for Multi-Core Structure

Aminollah Mahabadi **Fateme AsgariBidhendi**

Department of Electrical Engineering, Shahed University, Tehran, Iran

ABSTRACT

Chips can contain hundreds of pre-designed processing core being into a chip and have provided a chip with high complexity. In the case of such chips, as one of the most important issues is resources mapping and scheduling on the chip. It is well known that the mapping and scheduling tasks is an *NP* problem and need to be controlled together to achieve the effective scheduling and mapping. This paper obtains the solution near to the optimal of static tasks and communications mapping and scheduling in network on chip with two-dimension almesh architecture. The solution is an energy-aware heuristic algorithm, using a combination of genetic algorithm and simulated annealing which aims to minimize energy consumption and the implementation time of an application. Experimental results of some real and standard benchmarks show that the model proposed by this research provides average improvement of about %10in scheduling and improvement of more than %90 in execution time in mapping with relation to genetic algorithm

Keywords: Network on Chip (NoC), Mapping and Scheduling, Genetic Algorithm (GA), Simulated Annealing (SA), Integer Linear Programming (ILP).

The CSI Journal on Computing Science and Information Technology
Vol. 13, No. 2, 2016

Prolonging the Lifetime of Non-Volatile Last Level Cache Using Spare Blocks

Mohammad Reza Jokar[1,3] **Mohammad Arjomand[2,3]** **Hamid Sarbazi-Azad[3,4]**

[1]Computer Science Department, the University of Chicago, IL, USA
[2]Electrical Engineering and Computer Science Department, Pennsylvania State University, PA 16801, USA
[3]Computer Engineering Department, Sharif University of Technology, Tehran, Iran
[4]School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

ABSTRACT

Non-volatile memory technologies, such as STT-RAM, have high cell density and near-zero leakage power. Thus, non-volatile STT-RAM caches can be considered as a proper candidate to replace traditional SRAM-based caches. Beside the mentioned advantages, non-volatile technologies suffer from low write-endurance and this can lead to short lifetime of non-volatile caches. In this paper, we propose a new method to increase the lifetime of non-volatile last level caches by adding spare blocks in each set. When a block is failed in a set, we simply remove that block from the set and pad a spare block to the set. Evaluation results show that the proposed method can improve over the state-of-the art by 20% and 8% in lifetime and performance.

Keywords: Non-Volatile Memory, Last Level Cache, STT-RAM, Write Endurance, Lifetime.

The CSI Journal on Computing Science and Information Technology

The CSI Journal on
Computing Science and
Information Technology
Vol. 13, No. 2, 2016

A PSO-Based Task Scheduling Improved by Load-Balancing Technique for Cloud Computing Environment

Fatemeh Ebadifard **Ahmad Akbari**

Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

ABSTRACT

Cloud computing has attracted many researchers and users over the last few years due to the considerable advantages that cloud services provide for their consumers regarding cost and efficiency. Task scheduling in cloud environment is responsible for assigning tasks to virtual machines in a way to consider user requirements in the one hand and to increase resource utilization in the other hand. It is generally a NP-hard problem. In this paper we present a static task scheduling method based on Particle Swarm Optimization algorithm. We have improved our method to increase the resource utilization and performance, reduce make span time and keep the whole system balanced. The proposed method also considers performance of the base PSO method using a load balancing technique to produce better initial population. We have compared our method to Round Robin task scheduling, base PSO task scheduling and a load balancing technique. The simulation results show that our method outperforms all of the mentioned algorithms. It improves resources utilization by 22% and reduces the make span duration by 33% comparing to the base PSO algorithm.

Keywords: Cloud Computing, Scheduling, Particle Swarm Optimization, Load Balancing, Resource Utilization.

The CSI Journal on Computing Science and Information Technology

The CSI Journal on
Computing Science and
Information Technology
Vol. 13, No. 2, 2016

Locating Best Geographical Location for Green Datacenter in Iran with Respect to Continental, Political and Social Conditions

Majid Hajibaba **Saeed Gorgin**

Department of Electrical Engineering and Information Technology, Iranian Research Organization for Science and Technology, Tehran, Iran

ABSTRACT

Identifying and determining the geographic coordinates of the location of data center, is important for organizations. Evaluating, identifying and determining the coordinates depend on various factors, criteria and parameters while this diversity causes decision making be difficult for managers. This paper, for first time, comprehensively evaluates the decision criteria and explains the options available for hosting green data centers in Iran. The system presented in this paper, evaluate a comprehensive list of potential factors with respect to continental, political and social conditions and other factors. Then rank provinces in Iran to determine the geographical coordinates of data center. In this system, more than 40 criteria in four main categories include weather, natural disasters, facilities and unnatural events have been evaluated. The inference process, will be completed in an adaptive and flexible manner with integration and weighting parameters that obtained from experts' judgments and using a hierarchical analytics processing, and then by a major-voting mechanism. Finally, the system recommend appropriate geographical areas in high accuracy rate.

Keywords: Datacenter, Green Datacenter, Datacenter Site Locating, Datacenter Site Locating Criteria, Province Ranking.

The CSI Journal on
Computing Science and Information Technology

Vol. 13 **No. 2** **2016**

ABSTRACTS

- **Locating Best Geographical Location for Green Datacenter in Iran with Respect to Continental, Political and Social Conditions …… 1**
Majid Hajibaba and Saeed Gorgin

- **A PSO-Based Task Scheduling Improved by Load-Balancing Technique for Cloud Computing Environment ……...…… 2**
Fatemeh Ebadifard and Ahmad Akbari

- **Prolonging the Lifetime of Non-Volatile Last Level Cache Using Spare Blocks ……...……. 3**
Mohammad Reza Jokar, Mohammad Arjomand and Hamid Sarbazi-Azad

- **A New Energy-Aware Mapping and Scheduling Algorithm for Multi-Core Structure ……..… 4**
Aminollah Mahabadi and Fateme AsgariBidhendi

- **Cross-Lingual Opinion Mining using Semantic Features …….…… 5**
Shima Esmaeili Taft and Azadeh Shakery

- **A High-Performance and Low-Power Network-on-Chip Architecture for Neural Networks ….. 6**
Nasrin Akbari, Bita Dabiri and Mehdi Modarressi

The CSI Journal on Computing Science and Information Technology

A Semiannual Publication of Computer Society of Iran (CSI)

Editor-in-Chief

A. Khonsari, Associate Professor, University of Tehran, Tehran, Iran.

Editorial Board

H. R. Rabiee, Professor, Sharif University of Technology, Iran
H. Sarbazi-azad, Professor, Sharif University of Technology, Iran
K. Faez, Professor, Amirkabir University of Technology, Iran
A. Ghaffarpour Rahbar, Professor, Sahand University of Technology
E. Kabir, Professor, Tarbiat Modares University, Iran
K. Navi, Professor, Shahid Beheshti University, Iran
N. Yazdani, Professor, University of Tehran, Iran
M. H. Yaghmaee Moghaddam, Professor, Ferdowsi University of Mashhad, Iran
M. Analoui, Associate Professor, Iran University of Science & Technology, Iran
M. Ebrahimi Moghaddam, Associate Professor, Shahid Beheshti University, Iran
H. Asadi, Associate Professor, Sharif University of Technology, Iran
A. Akbari, Associate Professor, Iran University of Science & Technology, Iran
R. Berangi, Associate Professor, Iran University of Science & Technology, Iran
H. Pedram, Associate Professor, Amirkabir University of Technology, Iran
N. Moghadam Charkari, Associate Professor, Tarbiat Modares University, Iran
G. Jaberipur, Associate Professor, Shahid Beheshti University, Iran
J. Habibi, Associate Professor, Sharif University of Technology, Iran
A. H. Jahangir, Associate Professor, Sharif University of Technology, Iran
S. Hesabi, Associate Professor, Sharif University of Technology, Iran
S. H. H. S. Javadi, Associate Professor, Shahed University, Iran
M. Rahgozar, Associate Professor, University of Tehran, Iran
M. Sedighi, Associate Professor, Amirkabir University of Technology, Iran
H. Faili, Associate Professor, University of Tehran, Iran
A. Ghasemi, Associate Professor, K.N. Toosi University of Technology, Iran
M. Abbaspour, Associate Professor, Shahid Beheshti University, Iran
M. Abdollahi Azgomi, Associate Professor, Iran University of Science & Technology, Iran
M. Kargahi, Associate Professor, University of Tehran, Iran
M. Goudarzi, Associate Professor, Sharif University of Technology, Iran
N. Mozayani, Associate Professor, Iran University of Science & Technology, Iran

Assistants

L. Nourani, Publication Assistant
M. Dolati, Editorial Assistant

Disclaimer: Publication of papers in CSI-CSIT does not imply that the editorial board, reviewers, or CSI-CSIT accept, approve or endorse the data and conclusions of authors.