ششمین همایش بین‌المللی افق های نویـن در
مهندسی برق، کامپیوتـر و مکانیک
6th International Conference on the New Horizons in
Electrical Engineering, Computer and Mechanical
www.mhconf.ir

ارائه یک الگوریتم نگاشت شبکه روی تراشه به‌منظور بهبود پارامترهای کارایی مبتنی بر ساختار گراف و الگوریتم جستجوی هارمونی

سهیل طاهرپور [1]، پریسا دانشجو [2]

[1]کارشناسی ارشد، دانشگاه آزاد اسلامی، واحد تهران غرب، گروه کامپیوتر، تهران،، ایران taherpoor.soheil@wtiau.ac.ir
[1]استادیار، واحد تهران غرب، گروه کامپیوتر، تهران،، ایران daneshjoo.p@wtiau.ac.ir

چکیده

کاهش ترانزیستورها به کمتر از 50 نانومتر، منجر به افزایش تعداد ترانزیستورها به بیش از چندین میلیارد در یک تراشه می‌گردد؛ بنابراین باید روش‌های جدیدی برای مدیریت حجم انبوهی از ترانزیستورها بر روی یک تراشه اعمال شود. سیستم بر تراشه و شبکه بر تراشه دو روش پیاده‌سازی برای این مشکلات هستند. یکی از مهم‌ترین مباحث طراحی شبکه روی تراشه، یافتن یک نگاشت مناسب از گراف هسته‌های پردازشی یک سیستم بر روی گره‌های شبکه روی تراشه هست .به علت پیچیدگی بالای این انتخاب، الگوریتم‌های متفاوتی پیشنهاد شده‌اند که هر یک از مزایا و معایبی برخوردار هستند. رویکردهای کلی نگاشت در شبکه‌های روی تراشه، شامل نگاشت هسته‌های پردازشی بر روی گره‌های یک شبکه روی تراشه و نگاشت وظایف یک برنامه کاربردی بر روی هسته‌های پردازشی است. با توجه به ارزیابی‌های ارائه شده در این پژوهش، نگاشت شبکه روی تراشه بهینه ارائه می‌شود که ضمن تضمین تحمل‌پذیری خطا در روند مسیریابی، قابلیت اطمینان مسیر انتخاب شده را افزایش داده و با ایجاد تعادل بار در شبکه، علاوه بر کاهش ترافیک، منجر به کاهش زمان تحویل بسته‌ها به مقصد می‌شود.

واژه‌های کلیدی

نگاشت، شبکه روی تراشه، زمانبندی، مسیریابی، پارامتر کارایی، جستجوی هارمونیک، گراف.

1.مقدمه

با کاهش روز به روز اندازه در سیستم‌هایVLSI ، تجمیع ده‌ها بلوک از هسته‌های پردازشی در یک سیستم روی تراشه، تبدیل به عملی امکان پذیر گردیده است .در سالهای اولیه پیدایش سیستم‌های بر روی تراشه، گذرگاه‌ها با توجه به استفاده‌ای که در معماری پردازنده‌ها می‌شد به عنوان تنها راه برقراری ارتباط بین عناصر پردازنده و حافظه‌های بر روی تراشه بودند، در همان اوایل با توجه به تعداد کم عناصر پردازنده و حافظه‌های بر روی تراشه و در نتیجه میزان کم انتقال داده بر روی تراشه گذرگاه‌ها به خوبی از عهده این کار برآمدند و انتقال داده بین این عناصر را با کارایی بالا انجام دادند ولی با افزایش تعداد پردازنده‌ها و افزایش پیچیدگی سیستم‌های روی تراشه‌ای (SOC) گذرگاه‌ها دیگر جوابگوی این میزان انتقال داده‌ها نبودند. زیرا پهنای باند گذرگاه بین همه عناصر متصل شده به آن تقسیم می‌شد و این پهنای باند برای این میزان انتقال داده‌ها کافی نبود، با توجه به کارایی پایین گذرگاه‌ها طراحان به دنبال راهی برای حل مشکل انتقال داده‌ها بر روی تراشه‌ها پرداختند که منجر به پیدایش تکنولوژی شبکه بر روی تراشه (NOC) شد، بنابراین با افزایش پیچیدگی مدارات مجتمع، شبکه‌های روی تراشه راه حلی اساسی برای آدرس دهی چالش‌های ارتباطی در معماری‌های SOC به شمار آمدند [1].

شبکه روی تراشه، مدلی نوظهور برای ارتباطات درون سیستم-های VLSI بزرگ پیاده سازی شده بر روی یک تراشه سیلیکونی است. در یک سیستم شبکه روی تراشه پیمانه‌ها مانند هسته‌های پردازشگر، حافظه‌ها و بلوکهای IP ویژه با استفاده از یک شبکه مانند یک زیرسیستم "حمل و نقل عمومی" برای ترافیک اطلاعات، داده را رد و بدل می‌کنند. یک شبکه روی تراشه از چندین ساختار ارتباطی داده نقطه-به نقطه که با استفاده از سوئیچها (مسیریاب‌ها) به هم متصل می‌شوند، ساخته می‌شود به اینگونه که پیام‌ها می‌توانند از هر پیمانه منبع به هر پیمانه مقصد از راه چندین لینک با استفاده از تصمیم‌های مسیریابی در سوئیچها انتقال یابند. یک شبکه روی تراشه شبیه به شبکه مخابراتی مدرن با استفاده از سوئیچینگ بیت بسته دیجیتال در طول لینکهای مالتی پلکس شده هست[2] .

با پیدایش شبکه بر روی تراشه، مشکلات مربوط به انتقال داده‌ها بین پردازندهها، حافظه‌ها یا پردازنده و حافظه حل شد و تکنولوژی سیستم‌های موجود بر روی یک تراشه جای خود را به تکنولوژی شبکه بر روی تراشه داد .با حل شدن مشکلات مربوط به انتقال داده‌ها طراحان دوباره روز به روز بر تعداد پردازنده‌ها و حافظه‌ها بروی یک تراشه می‌افزودند تا بتوانند کارهای متفاوتی را با یک تراشه انجام دهند که این کار باعث پیچیده شدن شبکه و نامتقارن شدن آن شد و کارایی کل قطعه پایین آمد، همچنین مجتمع سازی سریع در تکنولوژی و افزایش پیچیدگی مدارات نیز از دلایلی است که می‌تواند رشد سریع احتمال روبرو شدن با خرابی‌ها را در اجزای مختلف شبکه‌های روی تراشه به دنبال داشته باشد، با توجه به موارد بالا طراحی NOC قابل اطمینان که در آن احتمال بازیابی از شکست و تحمل‌پذیری خرابی در حد مطلوب باشد، اهمیت زیادی یافته است. در حالت کلی افزایش کارایی سیستم‌های روی تراشه چندپردازنده‌ای، به‌عنوان یک از چالش مهم در محیط شبکه‌های روی تراشه است. مشکلات زیادی در استفاده از سیستم‌های روی تراشه چندپردازنده‌ای مبتنی بر شبکه روی تراشه وجود دارند، تخصیص مناسب و کارآمد منابع به وظایف یا همان نگاشت، یکی از این مشکلات است. هدف اصلی الگوریتم‌های نگاشت در سیستم‌های روی تراشه چندپردازنده‌ای مبتنی بر شبکه روی تراشه، کوتاه کردن زمان تکمیل محاسبات و بالا بردن توان عملیاتی سیستم است. همواره راهکارهای جدیدی برای انجام نگاشت در سیستم‌های روی تراشه چندپردازنده‌ای مبتنی بر شبکه روی تراشه پیشنهاد می‌شوند که هرکدام از این روش‌ها بخشی از مشکلات روش‌های قبلی را حل می‌کنند. ولی در کل هدف همه آن‌ها افزایش کارایی و کاهش زمان تکمیل محاسبات است [3].

در نگاشت ایستا، تخصیص وظایف به هسته‌های پردازشی، به‌صورت آفلاین و قبل از اجرای کاربرد انجام می‌گیرد. این نوع نگاشت سعی دارد که همیشه بهترین مکان وظایف را در زمان طراحی مشخص کند. ازآنجایی‌که نگاشت قبل از اجرای کاربرد انجام می‌گیرد، الگوریتم

نگاشت فقط یک‌بار اجرا می‌شود. برای شبکه روی تراشه، نگاشت ایستا پیشنهاد می‌شود، زیرا سربار محاسباتی موجود در نگاشت پویا به‌صورت چشمگیری بر روی عملکرد سیستم، تأثیر می‌گذارد و تأخیر کلی سیستم را افزایش می‌دهد. در کل تکنیک‌های نگاشت پویا، استراتژی‌های نگاشت آنلاین هستند. بنابراین مکان وظایف در شبکه روی تراشه می‌تواند در طول اجرای کاربرد تغییر کند. بررسی الگوریتم‌های نگاشت مختلف نگاشت درشبکه روی تراشه با دیدگاه بهبود شاخص اصلی کارایی که شامل توان مصرفی، گذردهی شبکه روی تراشه، تأخیر به‌عنوان تابعی از نرخ متغیر تزریق داده، توزیع مبدأ و مقصد، تعداد متغیر گره‌ها هست و اعمال یک الگوریتم نگاشت بهینه روی همبندی انتخاب‌شده و محاسبه کارایی آن برحسب پروتکل ارتباطی از قبیل استراتژی و الگوریتم مسیریابی، سایز بافر و تأخیر بسته است [4].

ساختارکلی مقاله به صورت زیر سازماندهی شده است. در ادامه و در بخش دوم، مروری کلی بر رویکردهای نگاشت در شبکه‌های روی تراشه، به همراه مزایا و معایب مرتبط با این رویکرد ارائه می‌شود. در بخش سوم راهکار پیشنهادی برای بهبود مولفه‌های کیفیت سرویس بیان شده و ساختار کلی آن ارائه می‌شود. در بخش چهارم، نتایج ارزیابی و شبیه‌سازی راهکار پیشنهادی ارائه می‌شود در نهایت در بخش پنجم، نتیجه‌گیری و پیشنهادات آتی ارائه می‌شود.

2. پیشینه پژوهش

روش‌های نگاشت در شبکه‌های روی تراشه، با توجه به زمانی که وظایف یک کاربرد برای پردازش به هسته‌های عملیاتی شبکه روی تراشه، تخصیص داده می‌شوند، به نگاشت پویا و نگاشت ایستا طبقه‌بندی می‌شوند. در نگاشت ایستا، برنامه کاربردی که قرار است بر روی شبکه بر تراشه اجرا شود، در زمان طراحی مشخص است و به طور پویا تغییر نمی‌کند. بنابراین نگاشت مربوط به وظایف قبل از اجرای کاربرد و در زمان طراحی مشخص می‌شود. برای یک کاربرد و زیرساخت‌های ارتباطی داده شده، با توجه به در دسترس بودن تمامی اطلاعات لازم، الگوریتم نگاشت ایستا سعی می‌کند بهترین مکان وظایف را در زمان طراحی مشخص کند [5]. در این روش چون نگاشت قبل از اجرای کاربرد کامل می‌شود، الگوریتم نگاشت تنها یک بار در زمان کامپایل اجرا می‌شود. بنابراین روی کارایی مربوط به کاربرد در زمان اجرا تأثیر نمی‌گذارد. در نگاشت ایستا چون پلتفرم برای اجرای کاربرد موردنظر اختصاص یافته، هیچ گونه رفتار پویایی از قبیل اضافه‌شدن، حذف یا مهاجرت وظایف در طول زمان اجرا، قابل قبول نیست. بنابراین انتظار می‌رود که الگوریتم‌های نگاشت ایستا، راه‌حل‌های نزدیک به راه‌حل بهینه بیشتری ایجادکنند. روش‌های نگاشت ایستا به دو دسته‌ی نگاشت دقیق[1] و نگاشت مبتنی بر جستجو[2] تقسیم می‌شوند [6].

نگاشت دقیق نگاشتی است که بر مبنای برنامه‌نویسی و فرمول‌بندی ریاضی راه‌حل بهینه را به دست می‌آورد. برنامه‌نویسی خطی عدد صحیح[3]، برنامه‌نویسی خطی غیر عدد صحیح[4] و برنامه‌نویسی خطی ترکیبی[5] سه نوع از مهم‌ترین الگوریتم‌های نگاشت دقیق هستند [6].

در مقاله [7] یک روش MILP جهت نگاشت وظیفه برای سیستم‌های چندپردازنده‌ای ناهمگن ارائه شده است. در این جا ابتدا به طور حریصانه هسته‌ها بر روی همبندی خاص نگاشت و سپس در مرحله بهبود، موقعیت‌های نسبی هسته‌ها توسط جستجوی ممنوع[6] ثابت می‌شوند. در مقاله [8] از روش MILP برای ساخت معماری NOC استفاده شده است که هدف، بهینه سازی و کمینه کردن مصرف توان با در نظر گرفتن محدودیت‌های کارایی می‌باشد. درILP ، گلوگاه اصلی زمان اجرا می‌باشد که برای کاهش زمان اجرا، گراف وظیفه‌ی کاربرد مورد نظر به تعدادی خوشه تقسیم بندی شده است. در مقاله [9] یک روش ILP دو مرحله‌ای برای تخصیص وظایف و نگاشت داده روی سیستم چندپردازنده‌ای متقارن[7] ارائه شده است. در مقاله [10]، رقابت در شبکه مورد بررسی قرار گرفته است. در این روش از یک فرمول بندی ILP برای نگاشت کاربرد رقابت-آگاه[8] جهت کمینه کردن رقابت استفاده شده است. در شبکه بر تراشه، سیم‌ها با یک شبکه-ای از پیوندهای مشترک جایگزین شده‌اند و مسیریاب‌ها بسته‌های داده را از طریق پیوندها به طور هم‌زمان جابجا می‌کنند. بنابراین ازدحام ترافیک در پیوندها ممکن است موجب کاهش عملکرد سیستم شود. رقابت در شبکه ممکن است در منبع، در مقصد و یا در مسیر باشد.

---

[1] Exact mapping
[2] Search based mapping
[3] Integer Linear Programming (ILP)
[4] Non Integer Linear Programming
[5] Mixed Integer Linear Programming
[6] Tabu search
[7] Symmetric Multi-Processing
[8] Contention-aware

کاهش رقابت در شبکه موجب کاهش تأخیر بسته‌ها می‌شود ولی اتلاف انرژی ارتباطی را افزایش می‌دهد. به دلیل آن که برخی از این روش‌ها زمان پردازش بالایی دارند در مقاله [11] جهت غلبه بر این مشکل، گراف وظایف کاربرد را خوشه‌بندی کرده و براساس تعداد خوشه‌ها، معماری توری را به شبکه‌های توری با ابعاد کوچک‌تر تقسیم می‌کند. برای نگاشت خوشه‌ها به زیرشبکه‌های توری مربوطه از فرمولبندی ILP استفاده شده است. در انتها همه‌ی زیرشبکه‌های توری برای تعیین راه‌حل نهایی ادغام می‌شوند. در این مورد زمان پردازنده بهبود پیدا می‌کند اما هزینه ارتباطی خوبی به دست نمی‌آید. دو نوع الگوریتم نگاشت ایستا از نوع مبتنی بر جستجو وجود دارد :

1) نگاشت مبتنی بر جستجوی قطعی[9]

2) نگاشت مبتنی بر جستجوی اکتشافی[10] [6]

الگوریتم‌های نگاشت قطعی تمام فضای جستجو را بررسی می‌کنند. الگوریتم شاخه وحد[11] جزء این دسته از الگوریتم‌های نگاشت است. این الگوریتم با جستجوی راه‌حل در شاخه‌های درخت و محدود کردن راه‌حل‌های غیرمجاز، نگاشت مناسب را پیدا می‌کند [6]. این الگوریتم برای مسائل کوچک مناسب است زیرا زمان جستجو با افزایش اندازه مسئله به طور نمایی افزایش می‌یابد [5]. در مقاله [12] یک نگاشت انرژی آگاه و در مقاله [13] یک نگاشت انرژی و کارایی- آگاه[12] با استفاده از الگوریتم شاخه و حد برای معماری NOC ارائه شده است که با هدف کمینه کردن انرژی ارتباطی، محدودیت‌های طراحی را از طریق رزوکردن پهنای باند برآورده می‌کند. در این جا الگوریتم سعی می‌کند هم‌زمان یک نگاشت بهینه و یک تابع مسیریابی پیدا کند که علاوه بر رعایت محدودیت پهنای باند، انرژی ارتباطی را بهینه کند. در مقاله [14] نیز از محدودیت پهنای باند و روش شاخه و حد برای نگاشت استفاده شده است. در روش مذکور ابتدا با استفاده از درخت جستجو یک مجموعه نگاشت با کم‌ترین هزینه ارتباطی پیدا می‌شود و سپس درگام بعدی نگاشت‌هایی با کم‌ترین تأخیر ارتباطی و توان مصرفی باقی می‌مانند. این روش نسبت به روش‌های قبلی از نظر کاهش تأخیر ارتباطی و توان مصرفی بهتر است. از آن‌جایی که مسائل نگاشت جزء مسائل NP-hard هستند بنابراین برای حل آن‌ها در اندازه‌های عملی‌شان از روش‌های جستجوی اکتشافی استفاده می‌شود. مسائل در اندازه‌های کوچک‌تر می‌توانند با روش‌های قطعی مانند روش شاخه وحد راه حل بهینه را ایجاد کنند [6]. روش‌های اکتشافی شیوه‌های بهینه‌سازی و جستجوی شبه‌تصادفی هستند که کشف و استخراج فضای راه‌حل را براساس تجربه به دست آمده، انجام می‌دهند. این روش‌ها هنگامی که جستجوی کامل فضای راه‌حل و روش‌های قطعی بسیار مشکل باشند و پیچیدگی زمانی به طور نمایی با اندازه مسئله رشد کند، استفاده می‌شوند. روش‌های اکتشافی در زمان نسبتاً کوتاه جواب مناسب را ایجاد می‌کنند [15]. روش‌های اکتشافی برای حل مسئله نگاشت کاربرد به دو دسته‌ی روش‌های اکتشافی قابل تغییر[13] و روش‌های اکتشافی سازنده[14] تقسیم می‌شوند. این روش‌ها راه‌حل‌های نگاشت موجود را تغییر می‌دهند تا به راه‌حل بهتر برسند. نمونه‌ای از این روش‌های اکتشافی روش‌های تکاملی شامل الگوریتم ژنتیک، الگوریتم بهینه‌سازی ازدحام ذرات[15]، الگوریتم کلونی مورچه[16] و موارد دیگر می‌باشند .[13]

3. روش پیشنهادی

هر فرایندی پتانسیل بهینه شدن دارد و مسائل پیچیده می‌توانند در زمینه‌های علوم مهندسی، اقتصادی و تجاری به‌صورت مسائل بهینه‌سازی، مدل‌سازی شوند. هدف از مدل‌سازی مسائل بهینه‌سازی، حداقل کردن زمان، هزینه و ریسک یا حداکثر کردن سود، کیفیت و اثربخشی است. بعضی از مسائل بهینه‌سازی پیچیده بوده و به دست آوردن پاسخ‌های بهینه در زمان معقول با روش حل دقیق مانند روش‌های برنامه‌ریزی پویا و شاخه و کران مشکل است. از این رو توسعه روش‌های حل در این نوع مسائل که بتوانند در زمان معقول جواب‌های بهینه یا نزدیک به بهینه به دست آورند، از نظر اقتصادی به صرفه‌تر است. در سالیان اخیر محققان در اکثر مسائل پیچیده بهینه‌سازی با پیاده‌سازی روش‌های فراابتکاری به نتایج مناسبی دست یافته‌اند. الگوریتم جستجوی هارمونی۱۷ یکی از ساده‌ترین و

---

[9] Deterministic search
[10] Heuristic search
[11] Branch-and-Bound
[12] Energy and Performance aware mapping
[13] Transformative heuristics
[14] Constructive heuristics
[15] Particle Swarm Optimization (PSO)
[16] Ant Colony Optimization (ACO)
[17] Harmony Search (HS)

جدیدترین روش‌های فراابتکاری است که در فرایند جستجوی جواب بهینه در مسائل بهینه‌سازی، از فرایند نواختن همزمان گروه ارکستر موزیک الهام گرفته شده است. به عبارت دیگر، میان پیدا کردن یک حل بهینه در مسئله پیچیده و فرایند اجرای موزیک، تشابهاتی وجود دارد. این روش حل را برای نخستین بار گیم در سال 2001 میلادی ارائه کرد مطابق با منطق این روش فراابتکاری، تلاش برای به دست آمدن هماهنگی در یک فرایند موزیک، مشابه پیدا کردن حل بهینه در مسائل بهینه‌سازی است.

HS مانند الگوریتم ژنتیک، جزء الگوریتم‌های بهبود دهنده است. به عبارت دیگر، با نسلی از بردارهای حل شروع و برای ایجاد نسل‌های جدید از فرایند انتخاب استفاده می‌شود. اما برخلاف الگوریتم ژنتیک (که در آن از دو کروموزوم برای تولید کروموزوم یا بردار حل جدید استفاده می‌شود) در این روش از همه بردارهای حل موجود در حافظه برای تولید راه حل جدید استفاده می‌شود. از مزایای این الگوریتم، همگرایی سریع آن به دلیل ساختار مناسب آن است و از معایب آن گیر افتادن در نقاط بهینه محلی به دلیل جستجو با تنوع کم در تکرارهای پایانی الگوریتم است که برای رفع آن از تکنیک فاز شروع دوباره و تغییر در قواعد الگوریتم به خصوص در تکرارهای پایانی استفاده می‌شود.

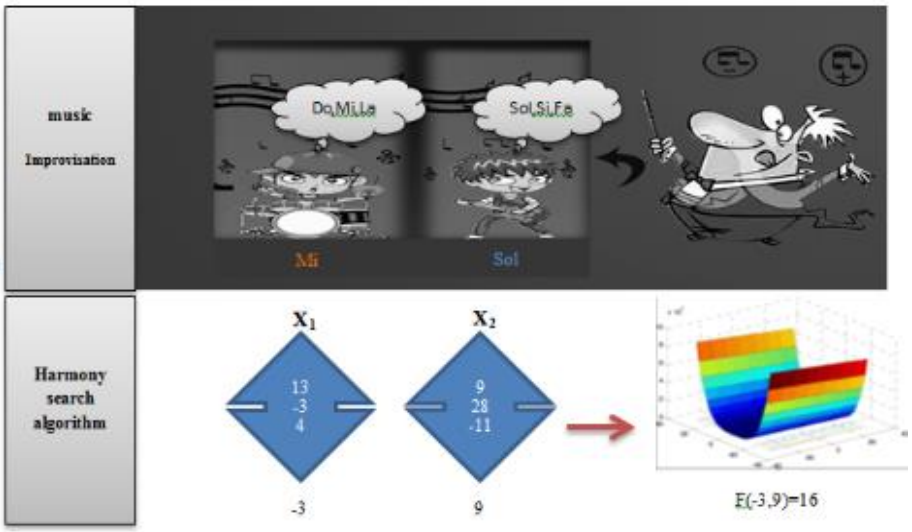
در دهه‌های اخیر برای حل مسائل بهینه‌سازی، روش‌های مختلفی توسعه داده شده است. این الگوریتم‌ها به دو دسته قطعی و احتمالی طبقه‌بندی می‌شوند الگوریتم‌های قطعی روش‌های جستجوی محلی بر پایه گرادیان هستند که به اطلاعات اساسی حرکتی برای پیدا کردن یک جواب شدنی نیاز دارند. در مسائلی که فضای حل غیر محدب باشد، پیدا کردن جواب بهینه سراسری با استفاده از این الگوریتم‌ها کار آسانی نیست. به عبارتی، بعضی از مسائل که غیر محدب هستند شامل چندین بهینه محلی در فضای حلشان هستند. در چنین مسائلی کیفیت جواب‌های نهایی به مقادیر اولیه معینی نیاز دارد. به همین دلیل محققین، در پژوهش‌ها به استفاده از الگوریتم‌های بهینه‌سازی احتمالی روی آورده‌اند. این الگوریتم‌ها از پدیده‌های طبیعی الهام گرفته‌اند و به اطلاعات حرکتی برای پیدا کردن جواب نیاز ندارند. مهم‌ترین الگوریتم‌های بهینه‌سازی احتمالی شامل الگوریتم ژنتیک، جستجوی ممنوع، بهینه‌سازی ذرات گروهی، شبیه‌سازی تبرید، کلونی مورچگان و جستجوی هماهنگی است. در سالیان اخیر الگوریتم‌های ترکیبی محلی سراسری برای حل مسائل غیر محدب با موفقیت اجرا شده‌اند. این الگوریتم‌ها جنبه‌های جستجوی محلی و جستجوی سراسری را در ساختارشان برای جستجوی مؤثر ترکیب می‌کنند. در این الگوریتم‌ها فرایند جستجوی سراسری با چندین نقطه شروع و جستجوی فضای حل از این نقاط شروع می‌شود. سپس الگوریتم‌های جستجوی محلی، پاسخ بهینه را در این فضاها پیدا می‌کنند. الگوریتم HS به دلیل کاربردی بودن برای مسائل بهینه‌سازی گسسته و پیوسته، محاسبات ریاضیاتی کم، مفهوم ساده، پارامترهای کم و اجرای آسان به یکی از پرکاربردترین الگوریتم‌های بهینه‌سازی در سال‌های اخیر در مسائل مختلف تبدیل شده است. این الگوریتم در مقایسه با سایر روش‌های فراابتکاری الزامات ریاضیاتی کمتری داشته و می‌توان آن را در مسائل مختلف مهندسی با تغییر در پارامترها و عملگرها تطبیق نمود. این الگوریتم به طور صعودی در سالیان اخیر توجه زیادی به خود معطوف کرده به طوری که تاکنون در مسائل بهینه‌سازی عملی نظیر بهینه‌سازی ساختاری، تخمین پارامترهای مدل غیرخطی مسیریابی، مدل‌های انتقال انرژی، برنامه‌ریزی و ...HS به کار رفته است. در بعضی از تحقیقات نشان داده شده است که در به دست آوردن جواب‌های مناسب، در زمان سریع‌تری نسبت به روش ژنتیک عمل می‌کند.

از مزیت‌های دیگر این روش نسبت به روش ژنتیک این است که برای ایجاد حل جدید برخلاف روش ژنتیک که از دو بردار حل در هر نسل استفاده می‌کند، این الگوریتم از همه حل‌های موجود در حافظه‌اش استفاده می‌کند. این ویژگی انعطاف الگوریتم را در جستجوی فضاهای بهتر حل افزایش می‌دهد. از ویژگی‌های دیگر الگوریتم جستجوی هارمونی این است که در مدت زمان مناسبی فضاهای حل با محدوده عملکرد بهتری را شناسایی می‌کند. این ویژگی در صورتی که مسئله مورد مطالعه از بهینگی محلی[18] برخوردار باشد، دچار مشکل می‌شود و در بهینگی محلی متوقف شده و نمی‌تواند به نقطه بهینه سراسری[19] برسد. دلیل این مشکل عدم کارایی مناسب الگوریتم در اجرای جستجوی محلی در مسائل بهینه‌سازی گسسته است. به‌منظور رفع این مشکل و تطبیق روش حل با مسئله، محققان با تغییر در پارامترها و عملگرهای الگوریتم، انواع مختلفی از این الگوریتم ارائه کرده‌اند تا دقت حل و نرخ همگرایی را افزایش دهند.

---

[18] Local optimum
[19] Global optimum

مطابق شکل 1، الگوریتم HS از فرایند بداهه‌سرایی موزیک گروه ارکستر پیروی می‌کند. هر نوازنده موزیک گام‌هایی از ابزارهای موسیقی خود را می‌نوازد تا شرایطی بهتر از هماهنگی در ارکستر به وجود آورد. هدف از این فرایند رسیدن به شرایطی است که هماهنگی کاملی از نواها ایجاد گردد. خروجی این هماهنگی کامل، صدای خوشایندی است که با استانداردهای زیبا مقایسه می‌گردد.



شکل 1. تشابه الگوریتم جستجوی هماهنگی و بداهه‌سرایی موزیک

1.3.ساختار روش حل

در این الگوریتم، هر حل یک هارمونی نامیده شده و با یک بردار N بعدی نمایش داده می‌شود. این الگوریتم دارای سه فاز اصلی می‌باشد:

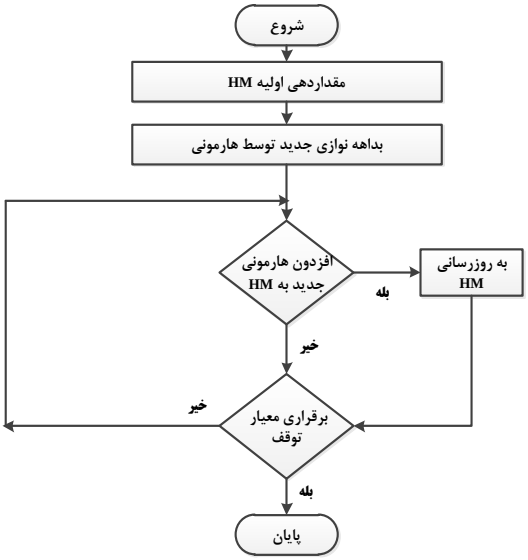
1) نسل اولیه (مقداردهی اولیه)
2) بهبود بردار هارمونی جدید
3) به روز کردن حافظه الگوریتم.

مطابق با فاز اول، یک نسل اولیه از بردارهای هارمونی به طور تصادفی ایجاد و در حافظه هارمونی[20] ذخیره می‌شود. در فاز دوم یک بردار هارمونی جدید (حل جدید) با استفاده از قواعد در نظر گرفتن حافظه، تطبیق گام و دوباره نسل‌سازی تصادفی از حل‌های موجود در حافظه الگوریتم ایجاد می‌گردد. گام‌های الگوریتم فوق عبارت‌اند از:

- گام اول: تعیین مقادیر اولیه پارامترهای الگوریتم
- گام دوم: تعیین مقادیر ابتدایی حافظه هارمونی
- گام سوم: تولید بردار حل جدید
- گام چهارم: به‌روزآوری حافظه هارمونی
- گام پنجم: تست قاعده توقف

فرایند کلی الگوریتم جستجوی هارمونی در شکل 2 نشان داده شده است.

[20] HMS: Harmony Memory size



شکل 2. دیاگرام الگوریتم جستجوی هارمونی

2.3. تعیین اولیه مسئله و پارامترها

در مرحله اول، مسئله بهینه‌سازی به صورت رابطه زیرمشخص می‌گردد.

$$Min:\{f(x)|x\in X\} \qquad (2)$$
$$Subject\ to\ :g(x)\geq 0\text{-}and\text{-}h(x)=0$$

بطوریکه f(x) تابع هدف، h(x) تابع قیود مساوی وg(x) تابع قیود غیرمساوی می‌باشد. x دسته‌ای از متغیر تصمیم گیریx_i وX مجموعه محدوده ممکن مقادیر هر متغیر تصمیم‌گیری می‌باشد و $X_{il}\leq X_i\leq X_{iU}$ است.

بطوریکه X_{il} و X_{iU} حدود بالایی و حدود پایینی برای هر متغیر تصمیم‌گیری است. پارامترهایHSA نیز در این مرحله مشخص می‌شوند که شامل اندازه حافظه هارمونی و یا تعداد بردارهای جواب در حافظه هارمونی، نرخ ملاحظه هارمونی[۲۱]، نرخ تنظیم صدادهی[۲۲]، تعداد متغیرهای تصمیم‌گیری و تعداد بداهه‌سازی[۲۳] یا ناحیه توقف الگوریتم می‌باشند. حافظه هارمونی[۲۴] یک محل حافظه است بطوریکه تمام بردارهای جواب (دسته‌های متغیر تصمیم‌گیری) ذخیره می‌شوند. در واقع HM شبیه به حوضچه ژنتیک در الگوریتم ژنتیک می‌باشد. HMCR و PAR پارامترهایی هستند که هر دو در مرحله سوم برای بهبود بردار جواب به کار می‌روند.

3.3. تعیین اولیه حافظه هارمونی

در این مرحله ماتریسHM با تعداد بسیار زیادی بردار جواب تولید شده به صورت تصادفی و با توجه بهHMS پر می‌گردد.

$$HM=\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} \end{bmatrix}$$

تولید هارمونی جدید بر اساس بداهه‌نوازی

[21] Harmony Memory Considering Rate (HMCR)
[22] Pitch Adjusting Rate (PAR)
[23] Number os Solution Vector Generations
[24] Harmony Memory Matrix (HM)

یک بردار هارمونی جدید بر مبنای سه قانون تولید می‌گردد که به آن بداهه‌نوازی می‌گویند و عبارتند از:

1) ملاحظات حافظه

2) تنظیم صدادهی

3) انتخاب تصادفی. در ملاحظات هارمونی، مقدار اولین متغیر تصمیم‌گیری برای بردار جدید از هر مقداری که در محدوده($x_1^1 - x_1^{HMS}$) جود دارد انتخاب می‌گردد. مقادیر دیگر متغیرها نیز به روش مشابه انتخاب می‌گردند.

HMCR که مقداری بین 0 تا 1 و متغیر است، نرخ انتخاب یک مقدار از مقادیر محاسبه شده قبلی که درHM ذخیره شده است، می‌باشد بطوریکه 1-HMCR نرخ گزینش تصادفی یک مقدار از محدوده ممکن مقادیر می‌باشد.

$$x_i' \Leftarrow \begin{cases} x_i' \in \{x_i', ..., x_i^{HMS}\}, & \Pr(HMCR) \\ x_i' \in X_i, & \Pr(1-HMCR) \end{cases} \quad (3)$$

برای مثال،HMCR برابر 0/85 نشان می‌دهد کهHSA مقدار متغیر تصمیم‌گیری را از مقادیر حساب شده قبلی موجود درHM با احتمال 85 درصد و یا از کل محدوده ممکن مقادیر با احتمال 15 درصد انتخاب می‌کند. هر جزء به دست آمده توسط ملاحظات حافظه برای تعیین اینکه تنظیم صدادهی مناسب‌تر است یا خیر امتحان می‌شود. این عمل با استفاده از پارامترPAR که نرخ تنظیم صدادهی است و به صورت زیر تعریف می‌شود، انجام می‌گردد.

$$x_i' \Leftarrow \begin{cases} Yes, & \Pr(PAR) \\ No, & \Pr(1\text{-}PAR) \end{cases} \quad (4)$$

مقدار (1-PAR) نرخ انجام هیچ عمل را مشخص می‌کند.

4.3. بروز رسانی حافظه هارمونی

اگر بردار هارمونی جدید از بدترین بردار هارمونی درHM بر مبنای تابع هدف انتخابی بهتر باشد، هارمونی جدید داخلHM قرار گرفته و بدترین هارمونی موجود از مجموعهHM کنار گذاشته می‌شود. اگر ناحیه توقف (حداکثر تعداد بداهه‌نوازی) ارضا شده باشد، محاسبات پایان‌یافته و در غیراینصورت مراحل 3 و 4 تکرار می‌گردند.

4. الگوریتم جستجوی هارمونی بهبود یافته

در این الگوریتم روش جدیدی در ایجاد بردار حل جدید ارائه شده است که دقت و همگرایی آن نسبت به روش جستجوی هارمونی ساده بهبود می‌یابد. نقش پارامترPAR در افزایش تنوع در شروع جستجوی الگوریتم در فضای حل و نقشbw در جستجوی محلی الگوریتم به منظور افزایش نرخ همگرایی تأثیرگذار است.

از معایب روشHS استفاده از مقادیر ثابتPAR وbw است که تنظیم مناسب این پارامترها مشکل است. همچنین از معایب دیگرHS این است که تعداد تکرارهایی که الگوریتم نیاز دارد تا حل بهینه را پیدا کند مناسب نیست. اگرPAR مقدارش کوچک وbw مقدارش بزرگ باشد عملکرد الگوریتم ضعیف است و باید برای رفع آن تعداد بهبودها را افزایش دهیم تا به حل بهینه دست یابیم. هرچه مقدار پارامترbw در تکرارهای ابتدایی بزرگتر باشد موجب می‌شود که الگوریتم، تنوع جستجو در کل فضای حل را افزایش دهد و در تکرارهای بعدی به منظور جستجوی محلی مقدار کمتر آن مناسب است. بنابراین مقادیر بزرگPAR و مقادیر کوچکbw در تکرارهای پایانی منجر به رسیدن به فضای بهینه و همگرایی به بهینگی می‌گردد.IHS مشابهHS است با کمی تفاوت که مطابق آن مقادیر پارامترهایPAR وbw به‌صورت پویا در هر تکرار جداگانه مطابق روابط زیر حاصل می‌گردند.

$$PAR(t) = PAR_{\min} + \frac{PAR_{\max} - PAR_{\min}}{NI} \times t \quad (5)$$

$$bw(t) = bw_{\max} \times e^{\left(\frac{\ln\left(\frac{bw_{\min}}{bw_{\max}}\right)}{NI} \times t\right)} \quad (6)$$

PAR_{min}، PAR_{max}، bw_{min}، bw_{max} مقادیر ثابتی هستند که در ابتدای الگوریتم IHS تعریف می‌شوند.T شماره تکرار ایجاد بردار حل جدید الگوریتم است.NI شماره تعداد بهبودها برای رسیدن به توقف است. به طور کلی در تمامی مطالعات نشان داده شده است کهIHS عملکرد بهتری نسبت به HS دارد.

5. شبیه‌سازی و ارزیابی نتایج

در این پژوهش ، برای شبیه‌سازی روش پیشنهادی، از نرم‌افزار ناکسیم[۲۵] و همچنین از یک رایانه شخصی با حافظه اصلی 8 گیگابایت و پردازنده 2.8 گیگاهرتزی در سیستم‌عامل ویندوز[۲۶] ده شرکت مایکروسافت استفاده‌شده است. معیارهای ارزیابی استاندارد از راهکار ارائه شده در مقاله [16] در این ارزیابی استفاده می‌شوند. بعنوان مثال، نرخ ورود بسته (PIR) اشاره به نرخی دارد که بسته‌ها وارد شبکه NoC می‌شوند. برای هر مسیریاب گره مشخص در NoC، تعداد نرمال سازی شده‌ی بسته‌های ارسال شده به ازای هر سیکل کلاک برای PIR است و دارای بازه‌ی $0 < PIR \leq 1$ است. اگر $PIR = 0/02$ باشد، به این مفهوم است که گره، 0/2 بسته به ازای هر سیکل کلاک یا 2 بسته در هر 10 سیکل کلاک ارسال کرده است. معیارهای کارایی توان عملیاتی، T و میانگین تأخیر D استفاده می‌شوند و در روابط (7) و (8) استفاده می‌شوند.

$$T = \frac{R_{flits}}{N_{nodes} - N_{clk}} \quad (8)$$

در رابطه (8)، R_{flits} کل فلیت های دریافتی، N_{nodes} کل تعداد گره ها و N_{clk} تعداد سیکل‌های کلاک ها از اولین فلیت تولید شده تا آخرین فلیت دریافت شده است. بنابراین، توان عملیاتی بصورت کسری از ماکزیمم بار محاسبه می‌شود که شبکه قادر به کنترل فیزیکی روی مسیر مورد نظر است. تأخیر بصورت تعداد سیکل‌های کلاکی تعریفی می‌شود که بین وقوع ورود فلیت هدر به شبکه در گره مبدأ و وقوع دریافت فلیت نهایی در گره مقصد سپری می‌شود.

رابطه (9) میانگین تأخیر D را تعریف می‌کند که میانگین مقدار برای کل تعداد پیام‌ها است، جاییکه K کل تعداد پیام‌هایی است که به گره‌های مقصدشان می‌رسند و D_i تأخیر برای گره i است.

$$D = \frac{1}{K}\sum_{i=1}^{k} D_i \quad (9)$$

محیط ارزیابی در جدول1 ارائه شده است. همانطور که بیان شد، شبیه ساز ناکسیم جهت ارزیابی کارایی الگوریتم مسیریابی بسط یافته است. پلتفرم ارزیابی الگوریتم در این مقاله برمبنای سیستم مش دو بعدی بود. برای تضمین صحت نتایج، شبیه سازی در هر نقطه PIR، چندین بار تکرار شد (پنج بار در این راهکار) و نتایج به صورت میانگین ارائه شدند. زمان‌های راه اندازی و اجرا بترتیب 1000 و 10000 سیکل کلاک هستند. الگوریتم مسیریابی تحت الگوهای ترافیک متعددی ارزیابی می‌شود که شامل ترافیک تصادفی[27]، ترافیک یکنواخت[28] و ترافیک ترانهاده[29] است. که الگوهای ترافیک رایج استفاده شده در ارزیابی کارایی مسیریابی هستند. برای ارزیابی روش پیشنهادی با رویکرد ویکور از یک الگوریتم با قابلیت آگاهی از تراکم با عنوان EDAR و دو الگوریتم XY و Odd-Even تحت شرایط ترافیک یکسان استفاده شده است.

جدول1. محیط ارزیابی برای روش پیشنهادی و راهکار مقایسه شده

پارامترهای ارزیابی	مقادیر پارامترها
شبیه ساز	Extended Noxim Simulator
توپولوژی	2D Mesh
الگوی ترافیکی	Random, Transpose, Shuffel
چرخه گرم کردن شبیه	1000

[25] Noxim
[26] Windowse
[27] Traffic-Random
[28] Traffic-Uniform
[29] Traffic-Transpose

سازی [cycle]	
[cycle]زمان اجرا	10000
زمان تکرار شبیه سازی	5

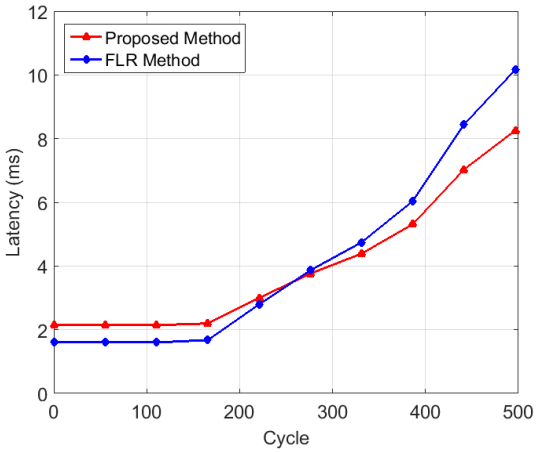
از آنجاییکه معماری‌های این راهکارها متفاوت هستند، نتایج تولید شده با کاهش توان عملیاتی توسط شبیه ساز نرمال سازی می‌شوند. بر اساس ارزیابی‌های انجام شده، جدول 2 نشان دهنده متریک عملکرد برای تأخیر ایجاد شده است که بر اساس تأخیر میانگین برای یک بسته و کل انرژی مصرفی شبکه ارائه شده است. توجه داشته باشید که توزیع به صورت تصادفی و با الگوی توزیع نرمال ارائه شده است. همچنین با توجه به این که الگوی ترافیکی استفاده شده، به صورت یک مجموعه داده از پیش تعیین شده است، در این شبیه‌سازی، جهت مقایسه صحیح با جستجوی هارمونیک فازی هیچ تغییری در داده‌های موجود انجام نشده است و تنها توزیع به صورت تصادفی در شبکه انجام شده است. بر اساس نتایج ارزیابی مشخص است که روش پیشنهادی دارای عملکرد بالایی نسبت به رویکرد جستجوی هارمونیک فازی است.

جدول2. مقایسه عملکرد تأخیر ایجاد شده بر اساس تأخیر میانگین

نرخ تأخیر (بسته / سیکل)	رویکرد مسیریابی پیشنهادی	رویکرد مسیریابی مبتنی بر راهکار فازی
50	1.8 ms	1.95 ms
100	1.89 ms	1.99 ms
150	1.98 ms	2.32 ms
200	2.41 ms	2.96 ms
250	3.46 ms	3.95 ms
300	4.12 ms	5.52 ms
350	4.72 ms	6.41 ms
400	6.73 ms	7.21 ms
450	7.21 ms	9.12 ms
500	8.14 ms	10.03 ms

نمودار مربوط به تأخیر میانگین بر اساس تعداد سیکل‌های شبیه‌سازی برای روش پیشنهادی و جستجوی هارمونیک فازی هیچ در شکل 3 نشان داده شده است، بر این اساس مشاهده می‌شود که به طور میانگین، روش پیشنهادی دارای تأخیر کمتری نسبت به راهکار پیشنهادی جستجوی هارمونیک فازی است. با توجه به این که در روش پیشنهادی از یک ساختار تصمیم گیری مبتنی بر اولویت به منظور انتخاب بهترین مسیر استفاده می‌شود، به نظر می‌رسد، کارایی روش پیشنهادی از سطح بالاتری برخوردار هست. استفاده از یک روند انتخاب مبتنی بر اولویت، ضمن کاهش تأخیر، انرژی مصرفی در شبکه را نیز کاهش می‌دهد. در ادامه، بر اساس ارزیابی‌های انجام شده، نرخ انتقال بسته برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی مقایسه می‌شوند. در جدول 3، نرخ انتقال بسته بر اساس تعداد سیکل‌های ارزیابی شده بیان شده است. بر اساس این جدول، مشاهده می‌شود که نرخ انتقال بسته در روش پیشنهادی نسبت به رویکرد جستجوی هارمونیک از کارایی بالایی برخوردار است و در تمامی سیکل‌ها، نسبت به این رویکرد بهتر عمل کرده است. کارایی بالای نرخ تحویل بسته در روش پیشنهادی را به این صورت می‌توان توجیه کرد که، در روش پیشنهادی از انتخاب مسیرهایی که دارای نرخ انتقال بسته و همچنین قابلیت اطمینان پایین‌تری دارند صرف نظر می‌شود. این روند باعث انتخاب مسیرها نگاشت بهینه برای انتقال بسته می‌شود. بهبود انتقال بسته ضمن کاهش ترافیک و جلوگیری از ایجاد نقطه داغ [30] می‌تواند باعث کاهش وقوع سردرگمی [31] بسته‌ها شود

[30] Hotspot
[31] LiveLock



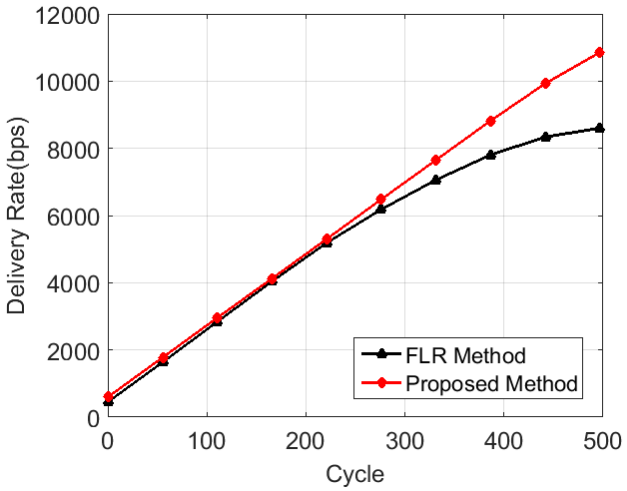
شکل 3. مقایسه تأخیر میانگین بر اساس تعداد سیکل‌های ارائه شده، برای روش پیشنهادی و جستجوی هارمونیک فازی

. زیرا در صورتی که ترافیک انتقالی برای مسیرها مدیریت شود، امکان سردرگمی در شبکه نیز تا حد قابل توجهی کاهش پیدا می‌کند.

جدول3. نرخ تحویل بسته برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی در هر 50 سیکل

نرخ تأخیر (بسته / سیکل- بیت در ثانیه)	رویکرد نگاشت پیشنهادی	رویکرد جستجوی هارمونیک فازی
50	476bps	541 bps
100	3121bps	3251 bps
150	4156 bps	4521 bps
200	5571 bps	5954 bps
250	6212 bps	6621 bps
300	7176 bps	7654 bps
350	8421 bps	8621 bps
400	9123 bps	92543 bps
450	9826 bps	9935 bps
500	11012 bps	11595 bps

نمودار مربوط به نرخ تأخیر برای سیکل‌های متفاوت از روش پیشنهادی و رویکرد جستجوی هارمونیک فازی در شکل 4 نشان داده شده است. همانطور که مشاهده می‌شود، تقریباً در تمامی سیکل‌های شبیه‌سازی، روش پیشنهادی نرخ تحویل بسته بالاتری دارد. با توجه به نکات بیان شده، روش پیشنهادی به دلیل بهگزینی در انتخاب مسیرهای مورد نیاز برای انتقال بسته، از مسیرها با کارایی بالاتر نگاشت استفاده کرده و در نتیجه می‌تواند ضمن کاهش نقاط داغ، نرخ تحویل بسته را نیز بهبود دهد.



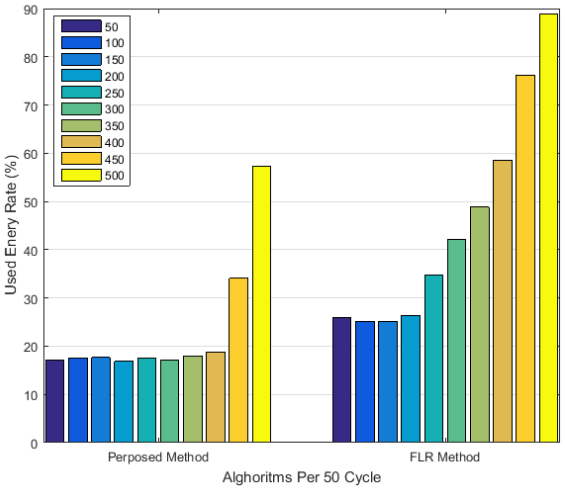
شکل4. مقایسه نرخ تحویل بسته به ازای هر بیت درثانیه بر اساس تعداد سیکل‌های ارائه شده، برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی

در ادامه روند شبیه‌سازی، سطح انرژی مصرفی برای روش مسیریابی پیشنهادی و رویکرد جستجوی هارمونیک فازی ارائه بررسی و ارزیابی شده و نتایج به دست آمده از روند ارزیابی، باهم مقایسه می‌شوند. نتایج ارزیابی مربوط به مصرف انرژی در روش پیشنهادی و راهکار مقایسه شده به ازای هر سیکل از شبیه‌سازی، در جدول 4 ارائه شده است. بر اساس این جدول، مشاهده می‌شود که میزان مصرف انرژی در روش پیشنهادی نسبت به راهکار ارائه شده مبتنی بر منطق فازی، سطح انرژی مصرفی کمتری دارد. باتوجه به این که در روش پیشنهادی، اولویت بندی مسیرهای ارائه شده، در روند کار قرار می‌گیرد، می‌توان ادعا کرد که به دلیل انتخاب بهینه مسیر، روند مصرف انرژی در روش پیشنهادی و حتی تعداد گام‌های مسریابی تا حد قابل توجهی کاهش پیدا خواهد کرد که در ادامه مورد بررسی قرار خواهد گرفت. روند کاهش مصرف انرژی، افزایش طول شبکه را در پی خواهد داشت.

جدول4. انرژی مصرفی برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی در هر 50 سیکل

مصرف انرژی (بسته / سیکل- درصد)	رویکرد مسیریابی پیشنهادی	رویکرد مسیریابی مبتنی بر راهکار فازی
50	17.25٪	28.56٪
100	17.65٪	28.96٪
150	17.86٪	29.32٪
200	17.92٪	30.95٪
250	18.25٪	34.52٪
300	18.89٪	43.56٪
350	19.14٪	49.01٪
400	21.53٪	58.62٪
450	34.78٪	74.36٪
500	56.53٪	85.69٪

در شکل 5، میزان مصرف انرژی بر حسب درصد برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی ارائه شده است. با توجه به این شکل مشاهده می‌شود که میزان مصرف انرژی در رویکرد جستجوی هارمونیک فازی نسبت به روش پیشنهادی در تعداد سیکل‌های بالاتر، سطح بیشتری دارد و به طور میانگین، میزان مصرف انرژی کلی برای روش پیشنهادی بسیار بهتر از روش مسریابی مقایسه شده است.



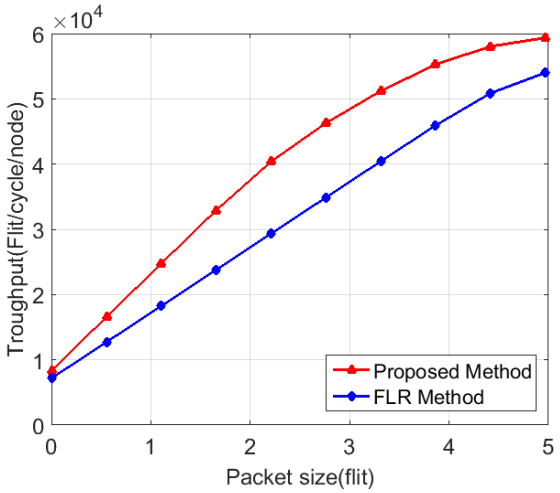
شکل5. مقایسه مصرف انرژی به ازای هر بیت درثانیه بر اساس تعداد سیکل‌های ارائه شده، برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی

در ادامه روند شبیه‌سازی، توان عملیاتی برای روش پیشنهادی و رویکرد مسریابی مبتنی بر منطق فازی مورد بررسی و ارزیابی قرار می‌گیرد. نتایج ارزیابی برای 500 سیکل از شبیه‌سازی، در جدول 5 نشان داده شده است. با توجه به نتایج به دست آمده، به نظر می‌رسد که روش پیشنهادی توان عملیاتی بهتری نسبت به رویکرد جستجوی هارمونیک فازی دارد. باتوجه به این که روند انتخاب بهینه‌تری در بین مسیرهای موجود در روش پیشنهادی انجام می‌شود، میزان توان عملیاتی در روش پیشنهادی نسبت به راهکار مقایسه شده سطح بهتری دارد. و در نتیجه میزان نگاشت در روش پیشنهادی بهتر خواهد بود

جدول5. نرخ توان عملیاتی برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی در هر 50 سیکل

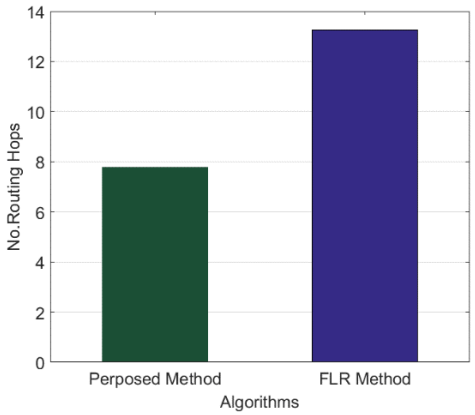
توان عملیاتی (اندازه بسته /فلیت)	رویکرد مسیریابی پیشنهادی	رویکرد جستجوی هارمونیک فازی
1Flits	1.72×10^{7}	1.06×10^{7}
2Flits	3.84×10^{7}	3.12×10^{7}
3Flits	4.36×10^{7}	3.98×10^{7}
4Flits	5.36×10^{7}	4.85×10^{7}
5Flits	6.02×10^{7}	5.12×10^{7}

نتایج ارزیابی برای توان عملیاتی بر اساس اندازه بسته در روش پیشنهادی و رویکرد جستجوی هارمونیک فازی در شکل 6 نشان داده شده است. بر اساس شکل ارائه شده، مشاهده می‌شود که نرخ توان عملیاتی برای راهکار پیشنهادی در تمامی اندازه‌های بسته، نسبت به روش مسیریابی مبتمی بر منطق فازی، از سطح بالاتری برخوردار است. با توجه به این که روش پیشنهادی مسیریابی، یک روش مسریابی آگاه از سطح انرژی بوده و بهینه‌ترین مسیرها را انتخاب می‌کند، می‌توان توان عملیاتی بهتری نسبت به راهکارهای دیگر انتظار داشت. همچنین به دلیل انتخاب بهینه مسیرها بر اساس پارامترهای کیفیت سرویس، امکان وقوع بن بست و یا دور در روش پیشنهادی تا حد قابل توجهی کاهش پیدا می‌کند.



شکل5. توان عملیاتی با توجه به اندازه بسته، برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی

در نهایت، نتایج ارزیابی برای مقایسه تعداد گامهای مسیر یابی در روش پیشنهادی و رویکرد جستجوی هارمونیک فازی در شکل 6 ارائه شده است. همانطور که مشاهده می‌شود، تعداد گام‌های مسیریابی در روش پیشنهادی به دلیل انتخاب بهینه مسیرها نسبت به رویکرد جستجوی هارمونیک فازی، کاهش پیدا می‌کند،



شکل6. مقایسه متوسط تعداد گام‌های مسیریابی برای روش پیشنهادی و رویکرد جستجوی هارمونیک فازی

این روند کاهش در تعداد گام‌های مسیر یابی، موجب افزایش طول عمر شبکه، کاهش مصرف انرژی و همچنین کاهش تأخیر در روش پیشنهادی نسبت به راهکار مقایسه شده می‌شود و در نتیجه کارایی روش پیشنهادی را نسبت به رویکرد جستجوی هارمونیک فازی، در سطح بالاتری قرار می‌دهد.

5. نتیجه‌گیری و پیشنهاد کار آتی

در این مقاله روش جدید مطرح شد. روش پیشنهادی با استفاده از نرم‌افزار ناکسیم شبیه‌سازی شد، سپس روش پیشنهادی با رویکرد درخت هارمونی مقایسه شده و درنهایت نتایج حاصل از شبیه‌سازی به صورت نمودار بیان شدند و نتایج مقایسه به صورت جداول نیز ارائه شد. به طور کلی، نتایج ارزیابی نشان داد که روش پیشنهادی نسبت به رویکردمقایسه شده، نرخ تاخیر را به اندازه 12 درصد بهبود می دهد. همچنین تعداد گام های مسیریابی برای روش پیشنهادی نسبت به رویکرد مقایسه شده به اندازه 7 درصد کاهش پیدا میکند. در ادامه نتایج ارزیابی نشان دهنده کارایی بالای روش پیشنهادی در توان عملیاتی بود، به طوری که نتایج نشان دهنده بهبود 5.6 درصدی

توان عملیاتی روش پیشنهادی نسبت به رویکرد مقایسه شده است. همچنین در این مقاله، نرخ تاخیر نیز مقایسه شده و نشان داد که 12 درصد نسبت به رویکرد مقایسه شده بهبود داشته است. در نهایت نتایج ارزیابی نشان داد که مصرف انرژی برای روش پیشنهادی، نسبت به رویکرد مقایسه شده، به اندازه 5 درصد کارایی بهتری دارد.

کار آتی از مباحثی مانند نحوه‌ی تخصیص مجدد منابع اصلی برای بازپیکربندی سیستم و نحوه‌ی فراهم سازی عملکرد سیستم در ظرفیت کاهش یافته خواهد بود، بطور خلاصه، کار آتی به بررسی نحوه‌ی تصمیم گیری برای اصلاح بهینه خواهد پرداخت و هدفش، ساخت برمبنای استراتژی نگاشت و تشخیص خرابی ارائه شده خواهد بود. به‌منظور بهبود عملکرد روش پیشنهادی می‌توان از سیستم‌های چند معیاری برای بهینه سازی استفاده کرد.

منابع

[1] Chip?, NoC: Network or, "I. Cidon," *First International Symposium on Networks-on-Chip (NOCS'07), Princeton, NJ*, pp. 269-269, 2007.

[2] Jain, Sourabh; Govani, Parimal; Poddar, Kamal B; Lal, A K; Parmar, R M, "Functional verification of DSP based on-board VLSI designs," *2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA)*, pp. 1-4, 2016.

[3] Morteza Mohaqeqi, Mehdi Kargahi,, "Utility accrual object distribution in MPSoC real-time embedded systems,," *Elsever,Journal of Computer and System Sciences,*, pp. 406-420, 2013.

[4] CÃsar Marcon, Thais Webber, Altamiro Amadeu Susin, "Models of computation for NoC mapping: Timing and energy saving awareness," *Microelectronics Journal,Volume 60*, pp. 129-143, 2017.

[5] Mesidis, P, "Mapping of Real-time Applications on Network-on-Chip based MPSOCS," *Department of Computer Science, University of York*, 2011.

[6] Sahu, P.K. and Chattopadhyay, S, "A survey on application mapping strategies for network-on-chip design," *Journal of Systems Architecture, Vol. 59, N0. 1*, p. 60–76, 2013.

[7] Bender, A, "MILP based task mapping for heterogeneous multiprocessor systems," *Proceedings of International conference on Design and Automation, Vol. 96*, pp. 190-197,, 1996.

[8] Srinivasan, K., Chatha, K.S. and Konjevod, G., "Linear-programming-based techniques for synthesis of Network-on-Chip architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 4*, pp. 407-420, 2006.

[9] Ostler, C. and Chatha, K.S, "An ILP formulation for system-level application mapping on network processor architecture", Proceedings of Design," *Automation and Test in Europe (DATE)*, pp. 99-104, 2007.

[10] Hu, J. and Marculescu, R, "Communication and task scheduling of application-specific networks-on-chip," *IEEE Proceedings Computers & Digital Techniques, Vol. 152, No. 5*, pp. 643- 651, 2005.

[11] Tosun, S, "Clustered-based application mapping method for Network-on-Chip," *Journal of Advances in Engineering Software, Vol. 42, No. 10*, pp. 868-874, 2011.

[12] Marculescu, R. and Hu, J, "Energy-aware mapping for tile-based NoC architectures under performance constraints," *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 233-239, 2003.

[13] Marculescu, R. and Hu, J, "Energy- and performance-aware mapping for regular NoC architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 4*, pp. 180-187, 2003.

[14] Reshadi, M., Khademzadeh, A. and Reza, A, "Elixir: a new bandwidth-constrained mapping for networks-on-chip," *IEICE Electronics Express, Vol. 7, No. 2*, pp. 73-79, 2010.

[15] Pop, R. and Kumar, S, "A Survey of Techniques for Mapping and Scheduling Applications to Network on Chip Systems," *Technical Report ISSN 1404-0018 04:4, ING Jönköping*, 2004.

[16] G. Ascia, V. Catania, M. Palesi, and D. Patti, Member, "mplementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip," *IEEE Computer Society*, 2008.