



Paper Type: Original Article



Enhancing Supply Chain Efficiency Using Lean Strategies and Distributed Computing: A Coordinated Approach to Waste Reduction and Resource Optimization

Amin Razani¹, Ashkan Mozdgir^{1,*}

¹ Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran;
enjoy.math@yahoo.com; a.mozdgir@khu.ac.ir.

Citation:



Razani, A., & Mozdgir, A. (2024). Enhancing supply chain efficiency using lean strategies and distributed computing: A coordinated approach to waste reduction and resource optimization. *Management: Modeling, analysis and applications*, 1(1), 47-73.

Received: 08/09/2023

Reviewed: 20/11/2023

Revised: 15/12/2023

Accepted: 25/03/2024

Abstract

Purpose: This study aims to improve supply chain efficiency by examining the integration of lean strategies with distributed computing architecture. The main goal is to reduce waste and optimize resource consumption by coordinating processes and utilizing modern processing technologies.

Methodology: In this study, a three-layer architecture is designed to implement lean concepts and distributed computing architecture simultaneously. The first layer provides real-time access to data throughout the supply chain. The second layer is responsible for coordinating lean processes with real demand, and the third layer offers optimal data distribution with observability and controllability. Algorithms are presented to manage node behavior and prevent network disruption.

Findings: The results of implementing this hybrid approach show that the proposed structure effectively reduces waste due to overproduction, product defects, unnecessary transportation, waiting times, and redundant processing. Also, reducing inventory levels, improving information flow, increasing agility, and the ability to respond quickly to changes in demand are other benefits of the designed system.

Originality/Value: The innovation of this research lies in simultaneously combining lean concepts with distributed computing technology and providing a coherent layered architecture to improve supply chain performance. This framework can be used as an intelligent and forward-looking model for designing efficient, flexible, and low-cost supply chains in various industries.

Keywords: Efficiency, Lean supply chain, Distributed computing, Waste reduction, Resource optimization.



Corresponding Author: a.mozdgir@khu.ac.ir

<https://doi.org/10.22105/mmaa.v1i1.70>

Licensee. **Management: Modeling, Analysis and Application**. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).



۶

نوع مقاله: پژوهشی

ارتقای کارآمدی زنجیره تامین با استفاده از استراتژی های ناب و محاسبات توزیع پذیر: یک رویکرد هماهنگ برای کاهش ضایعات و بهینه سازی منابع

امین رازانی^۱، اشکان مزدگیر^۲

^۱گروه مهندسی صنایع، دانشکده مهندسی، دانشگاه خوارزمی، تهران، ایران.

چکیده

هدف: این پژوهش با هدف ارتقای کارآمدی زنجیره تامین به بررسی تلفیق استراتژی های ناب با معماری محاسبات توزیع پذیر می پردازد. هدف اصلی کاهش انواع ضایعات و بهینه سازی مصرف منابع از طریق هماهنگ سازی فرایندها و بهره گیری از فناوری های نوین پردازشی است.

روش شناسی پژوهش: در این مطالعه، یک معماری سه لایه ای برای پیاده سازی هم زمان مفاهیم ناب و ساختار محاسبات توزیع پذیر طراحی شده است. لایه اول به فراهم سازی دسترسی آبی به داده ها در سراسر زنجیره تامین می پردازد. لایه دوم وظیفه هماهنگ سازی فرایندهای ناب با تقاضای واقعی را بر عهده دارد و لایه سوم نیز توزیع بهینه داده ها با قابلیت مشاهده پذیری و کنترل پذیری را فراهم می سازد. الگوریتم هایی برای مدیریت رفتار گره ها و جلوگیری از بروز اختلال در شبکه، ارائه شده اند.

یافته ها: نتایج حاصل از پیاده سازی این رویکرد ترکیبی نشان می دهد که ساختار پیشنهادی باعث کاهش موثر ضایعات ناشی از تولید بیش از نیاز، نقص در محصول، حمل و نقل های غیر ضروری، زمان های انتظار و پردازش های زائد می شود. همچنین، کاهش سطح موجودی، بهبود جریان اطلاعات، افزایش چابکی و توان پاسخگویی سریع به تغییرات تقاضا از دیگر مزایای سیستم طراحی شده، است.

اصالت/ارزش افزوده علمی: نوآوری این پژوهش در ترکیب هم زمان مفاهیم ناب با فناوری محاسبات توزیع پذیر و ارائه یک معماری لایه ای منسجم برای ارتقای عملکرد زنجیره تامین است. این چارچوب می تواند به عنوان مدلی هوشمند و آینده نگر برای طراحی زنجیره های تامین کارآمد، منعطف و کم هزینه در صنایع مختلف مورد استفاده قرار گیرد.

کلیدواژه ها: کارآمدی، زنجیره تامین ناب، محاسبات توزیع پذیر، کاهش ضایعات، بهینه سازی منابع.

۱- مقدمه

با گذشت زمان، پیچیدگی زنجیره تامین افزایش یافته است و وجود یک استراتژی، زنجیره تامین ملموس و موثر را حیاتی تر کرده است. داشتن یک استراتژی مدیریت ناب در مقایسه با یک زنجیره تامین معمولی می تواند مزیت رقابتی مناسبی در ایجاد زنجیره تامین کارآمد^۱ ایجاد نماید [1]. علی رغم مزایای فراوان استفاده از مدیریت ناب در زنجیره تامین، اجرای آن با چالش های متعددی نیز مواجه است که از جمله آن ها می توان به مقاومت در برابر تغییر در سطح کل شبکه، کمبود منابع، عدم ارتباط مناسب میان تامین کنندگان، تولید کنندگان و مشتریان، عدم وجود معیارهای اندازه گیری و کنترل کننده به هنگام و همچنین پیاده سازی ناسازگار و پراکنده فرهنگ ناب در سطح شبکه بدون یک رویکرد جامع و یکپارچه، اشاره نمود [2]؛ لذا ارتقای کارایی زنجیره تامین که با هدف بهینه و هماهنگ نمودن فرایندها در سطح شبکه برای کاهش زمان تحویل، ضایعات و هزینه ها مورد توجه قرار

^۱ Efficient supply chain

می‌گیرد، با همگرا شدن با استراتژی‌های ناب، قابلیت پیاده‌سازی و اجرا دارد. ایجاد این همگرایی از طریق استفاده از فناوری‌هایی مانند سیستم‌های اطلاعاتی و تحلیل داده‌ها، تسهیل می‌یابد. در واقع با شناسایی هوشمند هدررفت‌هایی مانند زمان‌های اضافی، موجودی‌های اضافی و حذف مراحل غیر ضروری می‌توان ارتقای کارایی زنجیره‌تأمین را همگرا با استراتژی‌های ناب در نظر گرفت [3]؛ لذا یک زنجیره‌تأمین کارآمد که بر اصول ناب بنا شده، می‌تواند ایجاد ارتباطات سریع و موثر میان تمامی زیرسیستم‌ها را تضمین نماید.

یکی از فرایندهای موثر در اجرای موثر تحلیل داده‌ها، محاسبات توزیع‌پذیر است که می‌تواند به زنجیره‌تأمین کمک کند تا ریسک‌هایی مانند تاخیر در تحویل، تغییرات ناگهانی در تقاضا و مشکلات تأمین مواد اولیه را به‌طور موثری مدیریت کند [4]. این امر از طریق مدل‌سازی و شبیه‌سازی سناریوهای مختلف و استفاده از داده‌های واقعی برای پیش‌بینی رویدادهای آتی میسر می‌شود. پیاده‌سازی استراتژی ناب که کاربرد متنوعی در بسیاری از حوزه‌های صنعتی دارد، با چالش‌های متعددی همراه است که از مهم‌ترین آن‌ها می‌توان به وجود داده‌های ناقص [5]، عدم ایستاد بودن شبکه و تغییرپذیری مداوم آن مبتنی بر عملکرد زیرسیستم‌های متصل به شبکه [6]، غیرخطی بودن عملکردهای سیستم [7] و لزوم مدل‌سازی به‌صورت ترکیبی از حالت‌ها و روش‌ها [8] اشاره نمود. به همین منظور شناخت رفتار و عملکرد استراتژی‌های ناب در شبکه‌های توزیع‌پذیر می‌تواند طیف متنوعی از نتایج را بازگو نماید تا در شرایط مختلف به بهبود عملکرد سیستم دست یافت.

محاسبات توزیع‌پذیر، در ایجاد شبکه زنجیره‌تأمین کارآمد می‌تواند به بهبود عملکرد، کاهش هزینه‌ها، افزایش انعطاف‌پذیری و تسهیل هماهنگی بین اجزای مختلف زنجیره‌تأمین کمک کند. با توجه به ظرفیت شبکه‌های محاسبات توزیع‌پذیر در تعریف گره‌های کانونی، تسهیل در یافتن گره‌های مخرب در سیستم و بهره‌مندی از لایه‌های محاسباتی مستقل، می‌توان با استفاده از الگوریتم‌هایی با ساختار نه‌چندان پیچیده، گره‌های ورودی به شبکه زنجیره‌تأمین را با مدیریت به‌هنگام به‌سمت گره‌های هدف سوق داد تا مسیر پیاده‌سازی استراتژی ناب در سطح سیستم تسهیل شود؛ بنابراین، استفاده از رویکرد محاسبات توزیع‌پذیر در زنجیره‌تأمین به ارتقای کارایی، کاهش هزینه‌ها، افزایش انعطاف‌پذیری و بهبود تصمیم‌گیری‌های استراتژیک کمک می‌کند. با بهره‌گیری از این رویکرد، سازمان‌ها می‌توانند به یک زنجیره‌تأمین هوشمندتر و کارآمدتر دست یابند که قادر است در دنیای پیچیده و پویا به‌سرعت واکنش نشان دهد.

برای ایجاد یک زنجیره‌تأمین کارآمد با استفاده از محاسبات توزیع‌پذیر، باید از الگوریتم‌های مختلف این رویکرد بهره برد تا بتوان به بهینه‌سازی فرآیندها، کاهش هزینه‌ها و افزایش انعطاف‌پذیری دست یافت. رویکرد محاسبات توزیع‌پذیر در یک زنجیره‌تأمین با توجه به مقیاس‌پذیری، توانایی دریافت و پردازش داده‌ها و همچنین برقراری ارتباط (بدون تاخیر در ارسال داده و تاخیر در تصمیم‌گیری) می‌تواند یکی از روش‌های موثر در ایجاد استراتژی‌های ناب در سیستم باشد. همچنین برای ایجاد یک زنجیره‌تأمین کارآمد با استفاده از محاسبات توزیع‌پذیر، می‌بایست تمام اعضای زنجیره‌تأمین به‌طور یکپارچه به هم متصل شوند، داده‌ها به‌طور لحظه‌ای و هم‌زمان پردازش شوند و تصمیمات مبتنی بر تحلیل‌های پیشرفته و داده‌های به‌روز اتخاذ شوند.

استفاده از محاسبات توزیع‌پذیر موجب افزایش انعطاف‌پذیری، سرعت، دقت و کارایی زنجیره‌تأمین می‌شود و در نتیجه شرکت‌ها می‌توانند هزینه‌ها را کاهش داده و به‌طور بهینه‌تر به تقاضای مشتریان پاسخ دهند. استفاده از محاسبات توزیع‌شده باعث می‌شود که اطلاعات مربوط به تقاضا، موجودی و حمل‌ونقل به‌طور هم‌زمان در سراسر زنجیره‌تأمین پردازش شود و تصمیم‌گیری در مورد تأمین و توزیع، سریع‌تر و به‌روزتر باشد تا از انباشت موجودی‌های اضافی و ضایعات ناشی از تاخیر در حمل‌ونقل و تولید جلوگیری شود؛ لذا کاهش ضایعات و هزینه‌ها از طریق پردازش توزیع‌شده امکان‌پذیر خواهد شد و استراتژی ناب برای ارتقای کارآمدی زنجیره‌تأمین ملموس و در دسترس می‌باشد. همچنین فرآیندهای پیش‌بینی تقاضا و موجودی‌ها می‌توانند به‌طور موازی در گره‌های مختلف اجرا شوند. به این ترتیب، زمان صرف‌شده برای شبیه‌سازی‌ها و محاسبات پیچیده کاهش می‌یابد و امکان تصمیم‌گیری سریع‌تر و دقیق‌تر فراهم می‌شود؛ لذا پژوهش حاضر به دنبال پاسخگویی به این پرسش است که رویکرد محاسبات توزیع‌پذیر برای ارتقای کارآمدی زنجیره‌تأمین در هنگام مواجهه با چالش‌های پیاده‌سازی استراتژی ناب چگونه است؟

۲- مبانی و ادبیات تحقیق

زنجیره تامین کارآمد

شبکه‌های زنجیره تامین حجم زیادی از داده‌ها را از نقاط مختلف جمع‌آوری می‌کنند: از جمله اطلاعات مربوط به موجودی کالا، زمان‌بندی حمل و نقل، وضعیت بازار و پیش‌بینی تقاضا. این داده‌ها به صورت روزانه یا حتی در زمان واقعی تولید می‌شوند و برای تحلیل‌های دقیق باید پردازش شوند. در یک زنجیره تامین پویا، داده‌ها باید در زمان واقعی پردازش شوند تا بتوان تغییرات در شرایط بازار یا موجودی را به سرعت شبیه‌سازی کرد و بر اساس آن تصمیمات جدید گرفت [9].

زنجیره تامین کارآمد به مجموعه‌ای از فرآیندها و فعالیت‌ها اطلاق می‌شود که در آن منابع و محصولات از تامین‌کنندگان به مصرف‌کنندگان با حداقل هزینه، کمترین زمان و بالاترین کارایی ممکن منتقل می‌شوند. لذا یک زنجیره تامین کارآمد به معنای زمان پردازش سفارش کوتاه‌تر، مدیریت موجودی بهتر، تحویل سریع‌تر و در برخی موارد سود بیشتر است. کارآمدی و کارایی زنجیره تامین به عنوان یک استاندارد داخلی شناخته می‌شود و به دنبال کاهش حداکثری معایب سیستم و زیرسیستم‌های متصل به آن قبل از توزیع محصول است. اجرای کارآمد مدیریت زنجیره تامین، با وجود مزایای متنوعی که ارائه می‌دهد، با چالش‌هایی همراه است [10]. زنجیره‌های تامین مدرن در مسیر تولید محصول اغلب پیچیده بوده و شامل تامین‌کنندگان و تولیدکنندگان در زیرسیستم‌های متعدد و گاهی با وظایف غیر مرتبط هستند. برخی از این چالش‌ها به صورت زیر فهرست شده است.

۱. عدم اطمینان از هماهنگی بین زیرسیستم‌های مسیر تولید، تمایل به ساده‌سازی فرآیندها را با چالش همراه می‌نماید.
۲. اجرای کارآمد مدیریت زنجیره تامین، به شدت به اشتراک به موقع و دقیق اطلاعات در مسیر تولید متکی است. با این حال، دستیابی به شفافیت و همکاری می‌تواند با مسایلی مانند سیلوهای داده، سیستم‌های فناوری اطلاعات ناسازگار و موانع ارتباطی بین نهادهای مختلف در زنجیره تامین مختل شود [11].
۳. پیش‌بینی رفتار زیرسیستم‌ها و روند هماهنگی آن‌ها به ویژه در هنگامی که استراتژی‌های مشخصی مورد توجه قرار گرفته است، چالش‌هایی را برای صنایع پویا (تغییرات مستمر مبتنی بر شرایط بازار) ایجاد می‌نماید [12].
۴. کسب و کارها اغلب به تامین‌کنندگان خود متکی هستند تا مواد و قطعات با کیفیت را به موقع تهیه کنند. با این حال، مسایلی مانند قابلیت اطمینان تامین‌کننده، کنترل کیفیت و ناهماهنگی میان زیرسیستم‌های متصل، ممکن است زنجیره تامین را مختل کرده و بر عملیات کلی آن تاثیر منفی داشته باشد [13].
۵. در حالی که فناوری می‌تواند کارایی را از طریق توزیع دقیق اطلاعات، ردیابی به هنگام و مدیریت زیرسیستم‌های مختل کننده، افزایش دهد، یکپارچه‌سازی زیرسیستم‌های جدید با فناوری‌های انحصاری می‌تواند پیچیده و پرهزینه باشد [14].

کارایی زنجیره تامین را می‌توان از طریق معیارهای که بر روی زمان لازم برای انجام یک مرحله تولید، هزینه و کیفیت محصول نهایی تمرکز می‌کنند، بررسی نمود. از مهم‌ترین این معیارها می‌توان به نرخ مشاهده و کنترل زنجیره تامین، نرخ خدمات یا تحویل به موقع، فرکانس ارسال سفارشات در بازه زمانی وعده داده شده، نسبت گردش موجودی و دقت سفارش اشاره نمود. با توجه این‌که هدف این پژوهش، شناخت رویکردهای متنوع محاسبات توزیع‌پذیر در تاثیرگذاری بر ساختار دریافت مواد اولیه (ورودی به زیرسیستم) تا تولید محصول (هدف و خروجی زیرسیستم) است، معیار نرخ دید زنجیره تامین مورد توجه است.

مشاهده‌پذیری به معنای ردیابی بهتر شبکه برای تولید محصول است [10]. در یک زنجیره تامین توزیع‌شده، اطلاعات و پردازش‌ها به صورت غیر متمرکز در بین چندین گره از تامین‌کنندگان که متناسب با سطح مشارکت خود در زیرسیستم‌های مستقلی حضور دارند، تقسیم می‌شود. شبکه محاسبات توزیع‌پذیر این امکان را فراهم می‌کند که پردازش‌ها به صورت موازی انجام شوند، به این معنا که عملیات مختلف در نقاط مختلف شبکه به طور هم‌زمان انجام شود. یکی از مزایای شبکه‌های توزیع‌شده این است که در صورت بروز خرابی در یک گره از یک زیرسیستم مستقل در شبکه، سایر گره‌های موجود در آن زیرسیستم، می‌توانند به کار خود ادامه دهند. این ویژگی می‌تواند موجب افزایش پایداری و قابلیت اطمینان سیستم شود.

محاسبات توزیع پذیر

محاسبات توزیع پذیر به سیستم‌هایی اطلاق می‌شود که در آن‌ها پردازش داده‌ها بین چندین گره تقسیم می‌شود، به طوری که هر گره مستقل عمل کرده و می‌تواند پردازش‌ها را به طور موازی انجام دهد. در چنین سیستم‌هایی، داده‌ها در چندین مکان فیزیکی (داده‌های توزیع شده) ذخیره و پردازش می‌شوند که این ویژگی می‌تواند به طور قابل توجهی در بهینه‌سازی کارایی و زمان پاسخ‌دهی سیستم‌های پیچیده مانند زنجیره‌های تامین، مفید باشد [15]. در شبکه محاسبات توزیع شده، عملیات و داده‌ها به صورت غیر متمرکز در نقاط مختلف شبکه پردازش می‌شوند. در واقع تامین کنندگان نسبت به وظایف خود به طور مستقل و در زمان‌های مختلف، داده‌های خود را پردازش می‌کنند [16]. یکی از چالش‌های اصلی در چنین سیستم‌هایی هماهنگی و هم‌زمان‌سازی بین این نقاط است. در شبکه‌های محاسبات توزیع شده، پردازش اطلاعات به صورت موازی^۱ انجام می‌شود؛ این یعنی هر گره یا واحد محاسباتی مسئول انجام بخشی از پردازش داده‌ها است. این نوع پردازش می‌تواند به شدت به بهبود سرعت و کارایی شبکه کمک کند. اگر یک مدل پیش‌بینی تقاضا در چرخه مواد اولیه برای تولید طراحی شده باشد، این مدل می‌تواند به صورت موازی بر روی داده‌های مختلف زیرسیستم‌ها، بخش‌های مختلف انبارها یا حتی داده‌های تولید مختلف پردازش شود. در این صورت، پردازش‌های مختلف در زمان کوتاه‌تری به نتیجه می‌رسند [17]. این نوع پردازش باعث می‌شود که منابع به طور بهینه‌تری مصرف شوند، چرا که محاسبات به طور هم‌زمان در گره‌های مختلف انجام می‌شود و نیازی به پردازش تک‌تک داده‌ها به طور خطی (سری) نیست. به این ترتیب، می‌توان حجم داده‌ها را به سرعت تحلیل کرد و تصمیمات سریع‌تری گرفت.

شبکه‌های محاسبات توزیع شده می‌توانند به راحتی مقیاس پذیر باشند، به این معنا که می‌توانند بدون تاثیر منفی بر عملکرد، گسترش یابند یا تغییرات جدید را بپذیرند؛ لذا می‌توان با اضافه کردن گره‌های جدید به شبکه، ظرفیت پردازشی و ذخیره‌سازی سیستم را افزایش داد. این ویژگی به ویژه برای شبکه‌های زنجیره تامین که به طور مداوم در حال تغییر و گسترش هستند، بسیار حایز اهمیت است. در زنجیره‌های تامین‌های بزرگ که نیاز به پردازش حجم بالای داده‌ها دارند، این ویژگی می‌تواند به طور چشمگیری کارایی و سرعت پاسخ‌دهی را افزایش دهد [18]. از طرفی در صورتی که تغییرات جدیدی در فرآیندهای زنجیره تامین نیاز باشد، سیستم‌های توزیع شده می‌توانند به راحتی خود را به روزرسانی کرده و تغییرات جدید را بپذیرند. این ویژگی باعث می‌شود که شبکه زنجیره تامین از تغییرات در شرایط بازار یا فرآیندها به راحتی عبور کند.

یکی از اولین کاربردهای محاسبات توزیع شده در زنجیره تامین، استفاده از فناوری شناسایی با امواج رادیویی برای ردیابی کالاها بوده است. با استفاده از این فناوری در سیستم‌های توزیع شده، شرکت‌ها توانستند به طور دقیق‌تری موجودی‌ها و محل کالاها را در زنجیره تامین ردیابی کنند. استفاده از محاسبات توزیع پذیر در زنجیره تامین، از دهه ۱۹۹۰ تاکنون به طور مداوم تکامل یافته است [19]. این رویکرد در ابتدا برای بهبود ارتباطات درون سازمانی و سپس به عنوان ابزاری برای پردازش داده‌ها و تصمیم‌گیری سریع‌تر و موثرتر در زنجیره تامین به کار گرفته شد. با گسترش فناوری‌های جدید مانند اینترنت اشیا، داده‌های بزرگ و یادگیری ماشین، این رویکرد به یکی از ارکان اصلی در بهینه‌سازی زنجیره‌های تامین تبدیل شده است و باعث افزایش مقیاس پذیری، انعطاف پذیری و کارایی این سیستم‌ها می‌شود. ایجاد همگرایی بین محاسبات توزیع پذیر و زنجیره تامین کارآمد می‌تواند به طور قابل توجهی کارایی، سرعت و انعطاف پذیری زنجیره‌های تامین را افزایش دهد [20]. با استفاده از پردازش موازی، اتوماسیون، یکپارچه‌سازی سیستم‌ها و استفاده از هوش مصنوعی، می‌توان به طور موثرتر داده‌ها را در سرتاسر زنجیره تامین مدیریت و بهینه کرد. این همگرایی به سازمان‌ها این امکان را می‌دهد که در دنیای پیچیده و پرچالش امروزی، زنجیره‌های تامین خود را با سرعت و دقت بهینه کنند و در برابر ریسک‌ها و بحران‌ها مقاوم‌تر شوند [7].

ساختار محاسبات توزیع پذیر از ۳ لایه: ارایه^۲، منطق و برنامه^۳ و ذخیره داده^۴ تشکیل شده است. در لایه PL، رابط‌های کاربر و تعاملات در سطح یک سیستم مستقل را مدیریت می‌کند. در این لایه بررسی و ارزیابی شناسه‌های مربوط به هر گره، ارزیابی و فراخوانی آن به صورت تصادفی انجام و اعتبارسنجی اولیه ورودی‌ها از جمله فرمت‌های داده‌ای و بررسی صحت داده‌ها صورت می‌گیرد. در لایه PL، توزیع درخواست‌ها توسط گره‌های مرکزی هر زیرسیستم انجام می‌شود تا بار به طور یکنواخت میان گره‌های مختلف توزیع شود. در لایه LAL، قوانین، چارچوب‌ها و الزامات شبکه پیاده‌سازی می‌شوند و از مهم‌ترین آن‌ها می‌توان به شناسایی و مدیریت گره‌های مخرب و نیز همسوسن‌م‌دن عملکرد گره‌های منطقی با گره‌های صحیح،

¹ Parallel processing² Presentation Layer (PL)³ Logic and Application Layer (LAL)⁴ Data Storage Layer (DSL)

اشاره نمود. لایه *LAL* مسئول پردازش و تحلیل داده‌ها بر اساس درخواست‌هایی است که از لایه *PL* دریافت می‌کند و داده‌های ورودی به بخش‌های کوچک‌تر تقسیم می‌شوند و هر بخش به‌طور موازی پردازش می‌شود. در لایه *DSL*، داده‌های صحیح در هر زیرسیستم نگهداری شده و توسط گره مرکزی هر زیرسیستم به گره اصلی و رهبر سیستم ارسال می‌شود. در واقع در این لایه، اطلاعات از پایگاه‌های داده یا سایر سیستم‌های ذخیره‌سازی، بازیابی می‌شود؛ لایه *DSL* مسئول ذخیره‌سازی، مدیریت و بازیابی داده‌ها است. در معماری‌های محاسبات توزیع‌شده، این لایه از پایگاه‌های داده توزیع‌شده و سیستم‌های ذخیره‌سازی ابری استفاده می‌کند که داده‌ها را در مکان‌های مختلف (گره‌ها) ذخیره می‌کند [21].

در محاسبات توزیع‌پذیر، داده‌ها به‌صورت مستقل در هر زیر سیستم پردازش می‌شوند. این فرآیند به محاسبات لبه^۱ نیز معروف است که به معنای پردازش داده‌ها در محل‌های نزدیک به محل تولید آن‌ها است. لذا داده‌ها به‌طور محلی در نقاط مختلف زنجیره‌تأمین پردازش می‌شوند. این موضوع باعث کاهش زمان تاخیر و افزایش سرعت تصمیم‌گیری می‌شود. از مهم‌ترین چالش‌ها برای پیاده‌سازی استراتژی‌های اجماع در سیستم‌های توزیع‌پذیر، وجود گره‌های مخربی است که به‌صورت دلخواه عمل می‌کنند. گره مخرب، هیچ پیامی در سیستم ارسال نکند و یا پیام‌های اشتباهی به گره‌های مجاور خود ارسال کند و همچنین در مورد مقادیر ورودی، اطلاعات نادرستی به اشتراک بگذارد [22].

الگوریتم‌های محاسبات توزیع‌پذیر

الگوریتم‌های محاسبات توزیع‌پذیر به‌دنبال شناخت و پیاده‌سازی مجموعه‌ای از وظایف در سطح سیستم است که در آن هر وظیفه بر روی یک پردازنده واحد اجرا می‌شود و با وظایف دیگر ارتباط برقرار می‌کند. ساختار الگوریتم‌های محاسبات توزیع‌پذیر، یک فرآیند را به وظایف متعدد تقسیم می‌کند و آن وظایف به پردازنده‌ها (گره) اختصاص داده می‌شود تا روند اجرا بتواند هزینه و زمان کل را به حداقل برساند [23]. یکی از اولین گام‌ها در حل مسایل توزیع‌شده، تقسیم سیستم به زیرسیستم‌های کوچک‌تر است. این کار از پردازش کامل داده‌ها در یک گره جلوگیری می‌کند. مراحل این فرآیند عبارتند از: ۱- تقسیم داده‌های ورودی سیستم در زیرسیستم‌های کوچک‌تر، ۲- تخصیص یک گره مرکزی به هر زیرسیستم و ۳- پردازش محلی داده‌ها توسط گره مرکزی با انجام محاسبات، تجزیه و تحلیل یا به‌روزرسانی اطلاعات [24].

الگوریتم‌های محاسبات توزیع‌پذیر به‌گونه‌ای طراحی شده‌اند که پردازش داده‌ها یا حل مسایل پیچیده را میان چندین زیرسیستم مستقل تقسیم می‌کند. معماری همتا به همتا^۲ در ساختار محاسبات توزیع‌پذیر به گره‌های موجود در زیرسیستم‌ها این امکان را می‌دهد که به‌صورت مستقل و با یک گره مرکزی داخلی با یکدیگر ارتباط برقرار کنند [25].

در محاسبات توزیع‌شده، الگوریتم همگام‌سازی^۳ مجموعه‌ای از الگوریتم‌ها هستند که برای هماهنگ‌سازی دسترسی به منابع مشترک و جلوگیری از شرایط رقابتی در محیط‌های توزیع‌شده طراحی می‌شوند. در این محیط‌ها، چندین گره به‌طور هم‌زمان می‌توانند به داده‌ها یا منابع مشترک دسترسی داشته باشند و اگر این دسترسی‌ها به‌درستی مدیریت نشوند، می‌توانند منجر به مشکلاتی مانند عدم انسجام داده‌ها، تضادها یا خرابی‌های سیستمی شوند. الگوریتم اجماع^۴ در محاسبات توزیع‌شده، مکانیزمی است که به گره‌های مختلف یک سیستم توزیع‌شده کمک می‌کند تا بتوانند با هم هماهنگ شوند و به یک توافق نهایی درباره وضعیت سیستم برسند. این فرآیند بسیار مهم است؛ زیرا گره‌ها ممکن است در مکان‌های مختلف قرار داشته باشند و نتوانند با یکدیگر ارتباط مستقیم داشته باشند یا حتی ممکن است برخی از گره‌ها دچار خطا شوند. هدف الگوریتم اجماع این است که سیستم حتی در صورت وقوع خرابی، به‌درستی کار کند و به یک تصمیم مشترک برسد [26]. در سیستم‌های توزیع‌شده، برای اجماع و هم‌زمان‌سازی گره‌ها به‌صورت هم‌زمان می‌توان از الگوریتم‌های اجماع استفاده نمود؛ این الگوریتم‌ها برای هماهنگ‌سازی تصمیمات و اطمینان از هم‌راستایی وضعیت‌ها در گره‌های مختلف شبکه به‌کار می‌روند.

ساختار پیاده‌سازی الگوریتم‌های ارایه‌شده، مبتنی بر الگوریتم *Raft* است که در آن از یک رهبر برای هماهنگی تصمیمات استفاده می‌کند. فرآیند *Raft* شامل انتخاب رهبر، ثبت لاگ‌ها و اطمینان از همگام‌سازی داده‌ها میان گره‌ها است. در این فرآیند، رهبر تصمیمات را به‌صورت مرکزی می‌گیرد و در صورت تغییر در داده‌ها، این تغییرات را به سایر گره‌ها اطلاع می‌دهد. لذا این فرآیند به مراتب از سایر الگوریتم‌های اجماع مانند *Paxos*

¹ Edge computing

² Peer-to-peer

³ Synchronization algorithms

⁴ Consensus algorithms

ساده‌تر بوده و برای استفاده در سیستم‌های عملیاتی مناسب‌تر است [27]. روش‌های حل در این الگوریتم‌ها معمولاً شامل تقسیم داده‌ها، پردازش موازی، همگام‌سازی و تجمیع نتایج است.

در روش پردازش موازی داده‌ها به‌طور موازی در چندین گره پردازش می‌شوند. این نوع پردازش سرعت حل مسایل را به‌طور چشمگیری افزایش می‌دهد. در واقع هر گره وظیفه مشخصی دارد که می‌تواند به‌طور مستقل انجام دهد، به‌طوری‌که هیچ وابستگی خاصی به پردازش‌های دیگر ندارد. این امکان باعث می‌شود که الگوریتم‌ها بتوانند سرعت پردازش را با استفاده از منابع محاسباتی متعدد افزایش دهند. در محاسبات توزیع‌شده، خرابی یا نقص یکی از گره‌ها نباید باعث توقف یا از دست رفتن داده‌ها شود. برای این منظور از پروتکل‌های بازگشت‌پذیر^۱ و نسخه‌برداری داده‌ها استفاده می‌شود تا در صورت خرابی یک گره، عملیات ادامه یابد. همچنین پروتکل‌های بازیابی به سیستم این امکان را می‌دهد که در صورت خرابی یک گره، پردازش‌ها را به گره‌های دیگر منتقل کرده و اطلاعات را بازسازی کنند [28].

با توجه به این‌که در این بررسی از الگوریتم‌های اجماع و هماهنگ‌سازی استفاده شده است، در وهله اول انتخاب رهبر در هر زیرسیستم توسط الگوریتم *Raft* انجام شده و در ادامه برای افزایش سطح تحمل خطا و مقیاس‌پذیری آن در مواجهه با تغییرات سیستم و همچنین جلوگیری از اختلال در به حد نصاب رسیدن فرآیند انتخاب رهبر در صورت عدم تمایل برخی از گره‌ها برای انتخاب رهبر جدید از الگوریتم انتخاب رهبر پویا^۲ که توسط مادیستی و پاندا [29] معرفی گردید، استفاده شده است. در این الگوریتم، تعیین گره رهبر که مبتنی بر وضعیت پویای سیستم است با چالش کمتری همراه است. در بسیاری از موارد می‌توان پیش‌بینی کرد که رهبر در شرف شکست است؛ در این حالت یک الگوریتم رهبر پویا می‌تواند قبل از خروج (یا شکست) پیش‌بینی‌شده رهبر فعلی، به ۳ حالت مختلف به انتخاب رهبر جدید اقدام کند. در این حالت می‌توان تعداد پیام‌های مبادله‌شده را به‌عنوان نمونه‌ای از تابع هزینه تفسیر نمود. در حالت اول، هنگامی که خروج رهبر فعلی از گروه پیش‌بینی می‌شود، رهبر فعلی همه گره‌ها را بررسی می‌کند و (با توجه به تابع هزینه) تصمیم می‌گیرد کدام واحد می‌تواند به‌عنوان رهبر جدید عمل نماید. در حالت دوم؛ پس از انتخاب رهبر در گروه، گره دیگری برای اشتراک وضعیت رهبری تعیین شده و لیست رهبری را قبل از خروج پیش‌بینی‌شده از گروه، حفظ می‌کند. در حالت سوم، پس از انتخاب رهبر، مجموعه‌ای از گره‌ها توسط رهبر جاری سیستم، تعیین می‌شود تا در هر زمان، وضعیت خود را برای کمک به دریافت موقعیت رهبری احتمالی آینده اعلام نمایند. در این حالت لیستی از رهبران آینده‌نگر پیشنهاد می‌شود که همگی دارای وضعیت رهبر هستند. گره‌های ورودی لزوماً ساختار مشخصی برای تحقق اهداف سیستم ندارند در واقع هیچ‌گونه نظارت و کنترلی از سوی سیستم برای گره‌های ورودی اعمال نمی‌شود؛ اما ایجاد غربالگری، یکسان‌سازی و مشابهت‌یابی برای ساخت گره‌های هدف برعهده ساختارهای کنترل‌پذیر در سیستم است. در هنگام ارزیابی مجموعه‌ای از گره‌های ورودی ساختار محاسبات توزیع‌پذیر، یک گره به‌عنوان گره رهبر برای ارزیابی و کنترل در نظر گرفته می‌شود. گره رهبر به‌عنوان گره ارزیاب، پارامترهای محاسبات توزیع‌پذیر را در سیستم اعمال نموده و به‌دنبال دریافت گره‌ها، هدف بیشتری از گره‌های ورودی در حال بررسی، است. گره‌ها با استفاده از توزیع نامشخص با یکدیگر ارتباط برقرار می‌کنند.

استراتژی ناب

استراتژی ناب به مجموعه‌ای از اصول اطلاق می‌شود که به‌دنبال کاهش هدررفت‌ها در فرآیندهای تولید و تامین است. این هدررفت‌ها می‌توانند در قالب زمان‌های تلف‌شده، منابع غیرضروری، موجودی‌های اضافی یا فرآیندهای پیچیده و ناکارآمد ظاهر شوند. هدف ناب، رسیدن به کارایی بالاتر با حداقل هزینه و منابع است [30]. استراتژی‌های ناب بر حذف هدررفت‌ها و بهینه‌سازی فرآیندها تمرکز دارد و می‌تواند در کنار استفاده از فناوری‌ها و رویکردهای مدرن در شبکه‌های زنجیره‌تامین، به حداکثر بهره‌وری و بهبود عملکرد کلی سیستم کمک کند. در استراتژی ناب، توجه به کیفیت محصولات و فرآیندها از اهمیت بالایی برخوردار است و کیفیت پایین می‌تواند منجر به ضایعات، هزینه‌های اضافی و اتلاف زمان شود؛ لذا در یک شبکه زنجیره‌تامین کارآمد، با استفاده از روش‌های ایجاد فرآیندهای بهبود مستمر، کیفیت و کارایی فرآیندها به‌طور مداوم افزایش می‌یابد [31]. رویکردهای ناب به شناسایی مشکلات قبل از وقوع و حل آن‌ها از طریق بهبود فرآیندها توجه دارند. این به معنای پیشگیری از ضایعات و افزایش کیفیت در تمام مراحل زنجیره‌تامین است. در استراتژی ناب، انعطاف‌پذیری یک ویژگی حیاتی است که به سازمان‌ها این امکان را می‌دهد که به تغییرات تقاضا یا مشکلات غیرمنتظره پاسخ دهند؛ لذا محاسبات توزیع‌پذیر به‌ویژه در شرایط بحرانی که تغییرات سریع در محیط رخ می‌دهد، کمک

¹ Fault-tolerant protocols

² Dynamic Leader Election Algorithm (DLEA)

می کند تا زنجیره تامین به طور سریع به این تغییرات واکنش نشان دهد [32]؛ زیرا سیستم های توزیع شده به سرعت می توانند وضعیت های بحرانی را شبیه سازی کرده و به طور آتی اطلاعات مربوط به وقوع تغییرات در تقاضا، مشکلات تامین یا اختلالات حمل و نقل را پردازش کنند. این اطلاعات به سرعت به تمامی بخش ها ارسال می شود و اقدامات اصلاحی به سرعت انجام می شود. با هماهنگ سازی محاسبات توزیع پذیر و استراتژی های ناب، می توان یک زنجیره تامین کارآمد، هوشمند و انعطاف پذیر ایجاد کرد. محاسبات توزیع پذیر با فراهم آوردن پردازش داده ها در زمان واقعی، کاهش پیچیدگی ها، تسهیل ارتباطات و تسريع تصمیم گیری ها، به استراتژی های ناب کمک می کند تا منابع بهینه تری تخصیص یابند، هدررفت ها کاهش یابد و انعطاف پذیری و واکنش پذیری زنجیره تامین افزایش یابد. این هم افزایی نه تنها باعث کاهش هزینه ها و زمان ها می شود بلکه تجربه مشتری را نیز بهبود می بخشد [33].

یکی از اهداف استراتژی ناب، کاهش زمان های فرآیند^۱ است. در زنجیره تامین، این زمان ها شامل زمان تولید، زمان حمل و نقل، زمان پردازش سفارش و زمان دریافت محصول است؛ لذا همگرایی استراتژی های ناب با شبکه زنجیره تامین کارآمد یک رویکرد قدرتمند برای بهبود عملکرد و کاهش هزینه ها است. با حذف هدررفت ها، کاهش زمان های فرآیند، بهینه سازی موجودی ها و ارتقا همکاری میان اعضای زنجیره تامین می توان به یک زنجیره تامین با بهره وری بالا و هزینه های پایین دست یافت. همچنین، استفاده از سیستم های اطلاعاتی پیشرفته و تحلیل داده ها در زیر سیستم های مستقل می تواند به پیاده سازی موفق این استراتژی ها کمک کند و زنجیره تامین را به یک شبکه بهینه و کارآمد تبدیل نماید. همچنین محاسبات توزیع پذیر باعث می شود که تمامی بخش های زنجیره تامین به طور آتی از وضعیت ها و تغییرات مطلع شوند [34]. این اطلاع رسانی سریع و بدون تاخیر به تصمیم گیری سریع تر و بهینه تر کمک می کند و از تصمیمات اشتباه ناشی از اطلاعات قدیمی یا ناکامل جلوگیری می کند. تمرکز استراتژی های ناب بر کاهش هزینه از طریق بهینه سازی فرآیند است در حالی که پیوند آن با ساختارهای محاسبات توزیع پذیر بهبود کیفیت با شناسایی و مدیریت عملکردهای مخرب را به ویژگی های استراتژی ناب اضافه می کند، است [8]. الگوریتم های ارایه شده در لایه های محاسبات توزیع پذیر می تواند به خوبی چالش های پیاده سازی استراتژی ناب را در شبکه زنجیره تامین پوشش داده و کارایی آن را ارتقا دهد که جدول ۱ دسته بندی مشخصی را در این خصوص ارایه داده است.

جدول ۱- رویکرد محاسبات توزیع پذیر در پیاده سازی استراتژی ناب.

Table 1- Distributed computing approach in implementing lean strategy.

فرآیندهای استراتژی ناب	رویکرد محاسبات توزیع پذیر
موجودی اضافی	مدیریت عملکرد سیستم
انتظار در محل تولید	
ضایعات ناشی از نقص	مشارکت پذیری در سیستم
تولید زباله قبل از مصرف	
ضایعات ناشی از پردازش بیش از حد	تعادل پردازش
تولید بیش از نیاز سیستم	
حمل و نقل بعد از تولید	مقیاس پذیری سیستم
حمل و نقل در مسیر تولید	

در واقع رویکرد محاسبات توزیع پذیر در شبکه زنجیره تامین کارآمد فقط برای انجام بهتر کارها نیست، بلکه راهی برای انجام کارهای بهتر است. استقرار ساختارهای محاسباتی در شیوه های زنجیره تامین کارآمد، به ارتقای بیشتر آن در مواجهه با چالش های بیان شده کمک می کند؛ زیرا با تاثیر مثبت بر سطوح عملیاتی، تاکتیکی و استراتژیک و فعالیت های مرتبط، کارایی عملیات را افزایش می دهد. در واقع به عنوان یک ابزار مدیریت استراتژیک، همگرایی ساختارهای محاسبات توزیع پذیر و استراتژی های ناب در شبکه زنجیره تامین کارآمد به عنوان یک فلسفه مدیریتی مهم در نظر گرفته می شود که از سازمان ها در تلاش هایشان برای ساده سازی عملیات، جلب رضایت مشتریان و ایجاد مزیت های متمایز حمایت می کند.

¹ Lead time

۳- پیشینه پژوهش

مطالعه چوپرا و سودی [35] با عنوان "کاهش خطر اختلالات زنجیره‌تامین" نشان داد، شرکت‌هایی که زنجیره‌تامین خود را بر پایه استراتژی ناب برنامه‌ریزی کرده‌اند، چنانچه عملکرد زیرسیستم‌های قبل از تولید محصول را به‌خوبی مدیریت نکنند، دچار اختلالات پرهزینه می‌شوند؛ لذا برای محافظت از زنجیره‌تامین در برابر اختلالات عمده، شرکت‌ها می‌توانند با تقسیم‌بندی یا منطقه‌ای کردن زنجیره‌های تامین، انعطاف‌پذیری ایجاد کنند و با اجتناب از تمرکز بیش از حد منابع، ضرر در عملکرد را محدود کنند.

عوامل کلیدی مرتبط با به‌کارگیری استراتژی زنجیره‌تامین موفق و کارآمد توسط دانشور و همکاران [14] مورد بررسی قرار گرفت. این نویسندگان بیان داشتند، با توجه به نیاز روزافزون سازمان‌ها به افزایش کارایی و کاهش هزینه‌ها، تمرکز بر استراتژی‌های زنجیره‌تامین کارآمد از اهمیت بالایی برخوردار است. نتایج نشان داد، بین اجرای استراتژی زنجیره‌تامین کارآمد با توجه به ویژگی‌های تحقیقات یکپارچگی داخلی، پشتیبانی مدیریت ارشد و فناوری اطلاعات با عملکرد زنجیره‌تامین رابطه مثبت وجود دارد؛ بنابراین، برای کاهش هزینه‌های اجرای استراتژی زنجیره‌تامین کارآمد، سرمایه‌گذاری بر روی عواملی که عملکرد زنجیره‌تامین را به‌طور مثبت تحت تأثیر قرار می‌دهند، ضروری است.

ژائو و همکاران [36] بیان داشتند، تعداد زیاد و تنوع گره‌ها دلیل اصلی پیچیدگی ساختاری در شبکه‌های زنجیره‌تامین به‌ویژه در هنگام پیاده‌سازی استراتژی‌های ناب هستند؛ لذا تلاش برای اصلاح و کاهش حجم شبکه می‌تواند در پیاده‌سازی استراتژی ناب نتیجه مطلوبی داشته باشد.

ماکیورا و همکاران [37] بیان داشتند، ادبیات ناب بیشتر بر جنبه‌های داخلی سیستم تمرکز دارد و مسایل یکپارچه‌سازی مشتری و تامین‌کننده را تحت پوشش قرار نمی‌دهد و به‌نظر می‌رسد محققان این واقعیت را نادیده گرفته‌اند که سیستم‌های پردازشی می‌تواند در زمینه‌های زنجیره‌تامین ناب نیز مفید باشد؛ زیرا انعطاف‌پذیری و فرآیند تصمیم‌گیری کارآمدتری را در زنجیره‌تامین فراهم می‌کند. در اغلب مطالعات انجام‌شده مانند پژوهش وو و همکاران [4]، دیودی و همکاران [38] و بوسکابادی و همکاران [39] سیستم‌های ناب تنها به‌عنوان یک مجموعه توزیع‌پذیر در بخش حمل‌ونقل و توزیع سفارشات محدود شده است. همچنین برخی از پارامترهای استراتژی‌های ناب در ایجاد شبکه‌های توزیع‌پذیر در پژوهش‌های ولیدی و همکاران [40] و دمال و همکاران [41] دیده شده است؛ اما تاکنون برای ارتقای کارآمدی زنجیره‌تامین، همگرایی میان رویکرد محاسبات توزیع‌پذیر و استراتژی ناب با در نظر گرفتن لایه‌های محاسباتی، مطالعه‌ای دیده نشده است. پژوهش کیم و همکاران [12] یک مدل ریاضی دو مرحله‌ای را با هدف افزایش کارایی زنجیره‌تامین با ارزیابی استراتژی‌های منابع متعدد و بهینه‌سازی تدارکات اضافی ارائه می‌کند. در مرحله اول، مدل شکاف بین فروش و برنامه‌ریزی عملیات و حداکثر مقدار محصول قابل دستیابی با سطوح موجودی فعلی را به حداقل می‌رساند و مرحله دوم بر بهینه‌سازی استراتژی تدارکات اضافی، با در نظر گرفتن زمان‌های هدایت، مقرون‌به‌صرفه بودن تمرکز دارد. در پژوهش آیکوج و همکاران [42]، تجزیه و تحلیل بهینه داده‌ها بر ارتقای کارایی زنجیره‌تامین بررسی و بر نقش آن در افزایش کارایی، کاهش هزینه‌ها و بهبود عملکرد مورد تأکید قرار گرفت. نتایج پژوهش نشان داد، ارتقای کارایی زنجیره‌تامین مستلزم توجه عوامل کلیدی مشاهده‌پذیری در زمان واقعی، تضمین امنیت و انتقال صحیح داده‌ها، مدیریت موجودی در مسیر تولید، حذف یا کاهش تاخیرهای عملیاتی و همچنین مدیریت بهینه ریسک است.

۴- روش‌شناسی

هدف از انجام این پژوهش بهره‌مندی از ساختار محاسبات توزیع‌پذیر در تحقق استراتژی‌های ناب در شبکه زنجیره‌تامین برای ارتقای کارآمدی آن است. به همین منظور با بهره‌مندی از همگرایی میان الگوریتم‌های اجماع و همگام‌سازی، ساختار شبکه زنجیره‌تامین را به زیرسیستم‌هایی مستقل (با عملکرد غیر مشابه) تقسیم و با تعیین گره‌های مرکزی برای هر یک از این زیرسیستم‌ها، الگوریتم‌هایی در سطح لایه‌های سه‌گانه محاسبات توزیع‌پذیر برای پیاده‌سازی موثر استراتژی‌های ناب فراهم نماید. با توجه به این‌که ارتقای کارآمدی زنجیره‌تامین در راستای پیاده‌سازی استراتژی ناب مورد توجه قرار گرفته است، گستردگی سطح ارتباط و اتصال بین گره‌ها در زیرسیستم‌ها منجر به افزایش ریسک‌پذیری، افزایش سطح خطای عملکردی، افزایش هزینه‌های متعدد می‌شود. لذا با استفاده از لایه‌های سه‌گانه و متوالی محاسبات توزیع‌پذیر می‌توان این چالش‌ها را در گام‌های متوالی و پی‌درپی مدیریت نمود. لایه‌های ۳ گانه محاسبات توزیع‌پذیر به‌عنوان فرآیندهای مستقل در ارزیابی و رابطه‌های بین آن‌ها به‌عنوان فرآیندهای وابسته به گره مرکزی سیستم شناخته می‌شود که گره‌های ورودی در هر زیرسیستم را پس از اعمال لایه‌های محاسبات توزیع‌پذیر به گره هدف تبدیل می‌کند. با توجه به این‌که شبکه زنجیره‌تامین کارآمد در استراتژی ناب قرار دارد، هر یک از گره‌های ورودی سیستم باید توسط حداقل یک فراخوان به

هر یک از فرآیندهای مستقل در ارزیابی، در ارتباط باشد. در این بررسی با توجه به پژوهش بلینگر و کاسی [43] وزن یال با توجه به شدت رابطه بین گره‌های فرعی با گره مرکزی در هر زیرسیستم تعریف می‌شود. همچنین ارجحیت لایه سوم به عنوان سومین فرآیند مستقل ارزیابی، از دو فرآیند دیگر بالاتر خواهد بود. تاثیر حذف گره مخرب در مکان‌شناسی شبکه زنجیره‌تأمین، یک موضوع مهم برای سیستم است که با مرکزیت اصلی شبکه به لایه ذخیره‌سازی داده (L_{SD}) واگذار شده است. با توجه به اهمیت این لایه، درجه و قدرت آن در سطح شبکه بالاتر است و به همین دلیل رابطه بین L_{SD} و سایر زیرسیستم‌های شبکه شدیدتر است و از طرفی، بیشترین تاثیرگذاری گره‌های مخرب بر روی گره اصلی L_{SD} است. در فرآیند مدل شبکه زنجیره‌تأمین ناب، گره‌های اصلی می‌توانند مقیاس گره‌های فرعی بالادستی (اتصال بالا به پایین) و پایین‌دستی (اتصال پایین به بالا) را کاهش دهند؛ بنابراین، وجود گره‌های مخرب، منجر به تاخیر در جهت‌دهی گره‌های ورودی به سمت گره‌های هدف خواهد شد و یک هزینه نامشخص برای سیستم تعریف می‌شود که متناسب با تعداد اتصالات با گره‌های اصلی متغیر است. در این لایه فرآیندهای کنترل‌پذیری، مدیریت و پیاده‌سازی می‌شوند. در بررسی کنترل‌پذیری و مشاهده‌پذیری در ساختار لایه‌های محاسبات توزیع‌پذیر، شناخت حالت‌های سیستم از اهمیت بالایی برخوردار است که با توجه مطالعات گذشته، معادله حالت به صورت رابطه (۱) در نظر گرفته می‌شود.

$$\dot{x}'(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0, \quad (1)$$

که معادله خروجی آن به صورت رابطه (۲) می‌باشد.

$$y(t) = Cx(t) + Du(t), \quad (2)$$

که در آن $x(t) \in R^n$ ، $y(t) \in R^m$ و همچنین $A \in R^{n \times n}$ ، $C \in R^{m \times n}$ ، $B \in R^{n \times r}$ ، $D \in R^{m \times r}$ در واقع n مرتبه سیستم، r تعداد ورودی و همچنین m تعداد خروجی را نشان می‌دهد. در تابع فرم خروجی با فرض $n > m \rightarrow D = 0$ و با فرض $n = m \rightarrow D = \text{Fix}$ با توجه به این که ساختارهای پیوسته می‌توانند با یک تابع پیوسته تناظر یک به یک داشته باشند، باید یک تابع پیوسته بردار $x(t) \in R^n$ و $y(t) \in R^m$ را به یکدیگر تناظر دهد تا ساختار مشتق‌پذیری آن در فضای حالت‌های $n \times m$ و $n \times n$ ارایه شود. با توجه به این که مشتق توابع نمایی در ساختار برداری بدون تغییر می‌باشد برای سادگی مشتق مرتبه‌های بالاتر فرض می‌شود رابطه (۳) برای بردارهای $x(t)$ برقرار است.

$$x(t) = e^{A(t-t_0)}x(t_0). \quad (3)$$

پس می‌توان ساختار مشتق‌پذیری را به راحتی بر روی ساختار قابل اندازه‌گیری سیستم اعمال نمود که به صورت رابطه (۴) ارایه می‌شود.

$$\begin{aligned} y(t_0) &= Cx(t_0), \\ y'(t_0) &= Cx'(t_0) = CAx(t_0), \\ y''(t_0) &= Cx''(t_0) = CA^2x(t_0), \\ &\vdots \\ y^{(n-1)}(t_0) &= Cx^{(n-1)}(t_0) = CA^{n-1}x(t_0). \end{aligned} \quad (4)$$

با توجه به این که گره‌های ورودی برای هر زیرسیستم به صورت یک الگوریتم پیوسته برای تبدیل شدن به گره هدف، ارزیابی می‌شود از طرفی هر یک از زیرسیستم‌های ایجادشده در سطح شبکه، به صورت یک ساختار پیوسته با گره مرکزی (رهبر) خود در ارتباط بوده و به دنبال بهینه‌نمودن استراتژی‌های خود در هر لحظه از عملکرد سیستم هستند؛ لذا ساختارهای کنترل‌پذیری برای سیستم‌های پیوسته مورد مطالعه قرار می‌گیرد. از طرفی مطالعه پارامترهای مشاهده‌پذیری، کنترل‌پذیری و پایداری در ساختار زمان پیوسته، چالش برانگیزتر از حوزه زمان گسسته است که نیازمند آگاهی از برخی از مفاهیم پیشرفته در حوزه جبر خطی می‌باشد. یک سیستم خطی با زمان پیوسته، می‌تواند گره‌های ورودی و خروجی را به صورت یک زوج مرتب در نظر بگیرد که بدون کاستن از کلیت مساله، می‌تواند ورودی‌های اسکالر در این رابطه را مجموعه از گره‌های ورودی در هر یک از زیرسیستم‌های شبکه در نظر گرفت. پس این گونه فرض می‌شود که روابط بررسی شده در این بخش برای هر یک از زیرسیستم‌های موجود در سطح شبکه به صورت مستقل انجام می‌شود. رابطه (۳) را می‌توان در ساختار ابعاد بالاتر توسعه داد و رابطه (۵) را ایجاد نمود.

$$X^n = \frac{d^n}{dt^n} x = A^n x + A^{n-1} b u + A^{n-1} b u' + \dots + b u^{(n-1)}. \quad (5)$$

از طرفی با توجه به رابطه نمایی (۳) ساختار به هم پیوسته فضای حالت سیستم به صورت رابطه (۶) بیان می شود.

$$x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^t e^{A(t-\alpha)} B u(\alpha) d\alpha. \quad (6)$$

رابطه (۶) بیان می کند، هر یک از گره های ورودی در یک زیرسیستم می تواند به مجموعه اسکالر B تعلق داشته باشد که سطح ریمان بالای منحنی استراتژی های ایجاد شده به وسیله گره های ورودی می تواند یک مساحت را ایجاد نماید. این مساحت که در بازه $[t_0, t]$ تعریف می شود، می تواند یک سیستم خطی کنترل پذیر را ارایه دهد که با توجه به خروجی انتگرال می تواند بر روی میزان افزایش ورودی های مربوط به بردار $x(t)$ تاثیرگذار باشد. در ابتدا فرض می کنیم گره های ورودی در یک زیرسیستم به مجموعه $IN_i^j = \{in_1^j, in_2^j, \dots, in_s^j\}$ تعلق دارد. لذا بردار $1 < k < s$ در زمان مشخص t پارامترهای مشاهده پذیری و کنترل پذیری را در مجموعه گره های ورودی IN_i^j بررسی می کند و در نتیجه رابطه (۷) تشکیل می شود.

$$x(in_t^k) = e^{A(in_t^k - t_0)} x(t_0) + \int_{t_0}^{in_t^k} e^{A(in_t^k - \alpha)} B u(\alpha) d\alpha. \quad (7)$$

پس از حل انتگرال، رابطه (۸) ایجاد می شود.

$$e^{-A \times in_t^k} x(in_t^k) - e^{-A \times t_0} x(t_0) = \int_{t_0}^{in_t^k} e^{A \times \alpha} B u(\alpha) d\alpha. \quad (8)$$

از طرفی با توجه به قضیه کیلی-همیلتون، رابطه (۹) به صورت مفروض در دسترس است که در آن به عنوان تابع زمانی اسکالر در نظر گرفته می شود و با توجه به مجموعه اسکالر $B, \beta_i(\alpha)$ می تواند مجموعه گره های ورودی IN_i^j را با توجه به قابلیت مشاهده پذیری سیستم رابطه (۸)، مشاهده و با توجه به رابطه (۷)، ارزیابی نماید.

$$e^{-A \times \alpha} = \sum_{i=0}^{n-1} \beta_i(\alpha) A^i. \quad (9)$$

با توجه به رابطه های (۸) و (۹)، رابطه (۱۰) در مجموعه گره های ورودی IN_i^j ایجاد می شود.

$$e^{-A \times in_t^k} x(in_t^k) - e^{-A \times t_0} x(t_0) = \sum_{i=0}^{n-1} B A^i \int_{t_0}^{in_t^k} \beta_i(\alpha) u(\alpha) d\alpha. \quad (10)$$

در تناظر با رابطه (۴)، رابطه (۱۱) در سیستم های دینامیکی خطی پیوسته ایجاد می شود.

$$X^{(n)}(in_t^k) - A^n x(in_t^k) = \gamma \begin{bmatrix} u^{(n-1)}(in_t^k) \\ u^{(n-1)}(in_t^k) \\ \vdots \\ u'(in_t^k) \\ u(in_t^k) \end{bmatrix}. \quad (11)$$

با توجه به این که در رابطه $e^{-A \times in_t^k} x(in_t^k) - e^{-A \times t_0} x(t_0)$ همه کمیت ها و ساختارها مبتنی بر گره های ورودی در یک زیرسیستم تعریف شده است، تمامی گره های ورودی به زیرسیستم قابل مشاهده است و مجموعه IN_i^j می تواند به پارامتر کنترل پذیری متصل شود. از طرفی در سمت راست رابطه (۱۷) ماتریس کنترل پذیری در برداری ضرب شده است که اجزای آن توابعی از گره های ورودی با ساختارهای کنترل پذیر هستند. در واقع همه کمیت ها و شناسه ها مشخص هستند و یک خروجی ثابت برای هر زیرسیستم تعریف می شود. این خروجی ثابت مبتنی بر زمان های قابل بررسی in_t^k در نظر گرفته شده است که می تواند به یک تابع نمایی از بردارهای وابسته به گره های هدف جهت دهی شود؛ لذا الگوریتم پیشنهادی برای مدیریت

عملکرد گره‌های مخرب، می‌تواند متناسب با درجه و میزان اهمیت گره‌های متصل به گره‌های اصلی تعمیم داده شود. به‌عنوان مثال در لایه اول که به‌عنوان PL در محاسبات توزیع‌پذیر مطرح می‌شود، پردازش داده‌ها به‌طور کامل انجام نمی‌شود.

۴-۱- مفروضات اولیه

۱. تمام گره‌ها در شبکه یک شناسه (ID) منحصر به فرد دارند.
۲. تعداد ثابتی از شناسه‌های گره و برخی اطلاعات اضافی را می‌توان در یک پیام منتقل کرد.
۳. گره‌ها برای اولین بررسی به‌صورت هم‌زمان عمل می‌کنند؛ اما در ادامه با توجه به شناسه و پروتکل اجرایی سیستم ناهم‌زمانی برای دریافت پاسخ برای آن‌ها وجود دارد.
۴. $DLEA$ ، برای هر زیرسیستم مستقل پیاده‌سازی می‌شود.
۵. شناسه، هر گره دلخواه را برای رهبر شدن فعال می‌نماید و سطح عملکردی آن‌ها را در همگرایی مجموعه نشان می‌دهد. در واقع گره‌هایی که برای رهبر شدن در هر مجموعه تلاش می‌کنند، جزو گره‌های مخرب و غیر فعال، قرار نمی‌گیرند.

برای [44] بیان داشت در پروتکل اجماع در شبکه‌های توزیع‌پذیر، اگر تعداد گره‌های غیرصحیح (منطقی و معیوب) از نصف تمام گره‌ها کمتر باشد، این الگوریتم می‌تواند به اجماع باینری با زمان اجرا مورد انتظار $O(2^n)$ دست یابد. در واقع زمانی که گره‌های مخرب حداقل از نصف تمام گره‌ها بیشتر باشند، هیچ الگوریتم توافقی برای انجام دستورات ناهم‌زمان وجود ندارد و خاتمه‌پذیر نخواهد شد. اگر چنین الگوریتمی وجود داشته باشد، در بهترین حالت باید گره‌های موجود در یک زیرسیستم را به دو بخش مجزا و مساوی از گره‌های مخرب B_N و گره‌های صحیح N_A (با اندازه $\frac{n}{2}$) دسته‌بندی کند. این بررسی برای ساختارهای توزیع‌پذیری که می‌تواند توسط گره‌های کانونی مدیریت شود، تقویت گردید و گره‌های منطقی نیز توسط الگوریتم ارایه‌شده توسط کینگ و ساپا [45]، به‌عنوان گره‌های صحیح شناخته شد، زیرا می‌توان شناسه پیام‌های گره کانونی را به‌صورت مستقل به این گره‌ها ارسال نمود و استراتژی‌های این گره‌ها را نیز در راستای گره‌های صحیح در نظر گرفت؛ لذا برای دستیابی به الگوریتم‌های خاتمه‌پذیر بهینه در یک زیرسیستم، تنها می‌توان حداکثر به اندازه $\frac{1}{3}$ از گره‌های زیرسیستم، پتانسیل وجود گره‌های مخرب را بررسی نمود. در واقع باید رابطه $0 < B_N \leq \frac{n}{3}$ برقرار باشد که در این صورت پیچیدگی زمانی الگوریتم به $O(n)$ تقلیل می‌یابد. با توجه به این بررسی، هر گره منطقی در یک زیرسیستم، شناسه یکتای خود را ارسال می‌کند و در مقابل منتظر دریافت پیامی برای تایید آن از سوی گره کانونی می‌ماند و اطلاعات و داده‌های واقعی در یک دور i (در صورت وجود) به اشتراک گذاشته می‌شود. اگرچه این دستورات، امکان همگام‌سازی ساده و سریع را فراهم می‌کند، اما پیام‌های بسیار زیادی از سوی گره‌های زیرسیستم تولید می‌کند که می‌تواند یک ساختار غیرخطی با حداقل مرتبه ۲ (n^2) ایجاد کند و محاسبات یک الگوریتم را افزایش دهد؛ لذا همگام‌سازی در هر زیرسیستم به یک مقداردهی اولیه و ثابت نیاز دارد که توسط گره رهبر به تمامی گره‌ها ارسال می‌شود.

۴-۲- تجزیه و تحلیل

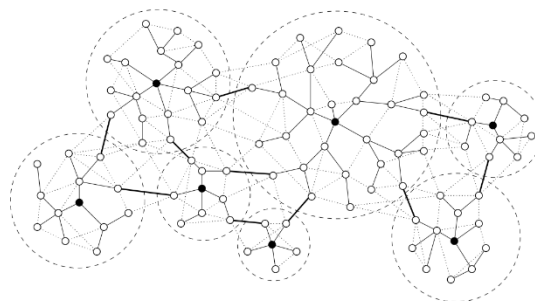
در سیستم‌های زنجیره‌تأمین توزیع‌شده که در آن داده‌ها در نقاط مختلف قرار دارند، ترکیب الگوریتم‌های اجماع و همگام‌سازی می‌تواند برای هم‌زمان‌سازی داده‌ها و اطمینان از یکپارچگی اطلاعات استفاده شوند. این الگوریتم‌ها می‌توانند در بهینه‌سازی منابع، بهبود زمان‌بندی و حمل‌ونقل و ایجاد شفافیت و هماهنگی در سراسر زنجیره‌تأمین نقش کلیدی داشته باشند. برای ارتقای کارآمدی زنجیره‌تأمین در پیاده‌سازی استراتژی ناب و مبتنی بر شبکه محاسبات توزیع‌شده، می‌توان از الگوریتمی استفاده کرد که از اصول استراتژی ناب (از جمله حذف هدررفت‌ها، کاهش موجودی، کاهش زمان چرخه و بهبود بهره‌وری) بهره‌برد و در عین حال از قدرت پردازش توزیع‌شده برای بهینه‌سازی و تصمیم‌گیری سریع‌تر و موثرتر استفاده کند. در واقع ارتقای کارآمدی زنجیره‌تأمین نیازمند یک فرآیند است که بتواند از مزایای توزیع منابع، پردازش موازی و بهینه‌سازی‌های مختلف در سطوح مختلف بهره‌برد. در یک زنجیره‌تأمین مبتنی بر شبکه محاسبات توزیع‌شده، هماهنگی و همگام‌سازی میان گره‌ها به‌عنوان یک بخش اساسی برای تبادل صحیح و به‌موقع داده‌ها و انجام عملیات صحیح در نظر گرفته می‌شود. استفاده از پروتکل‌های اجماع و سیستم‌های هماهنگ‌سازی توزیع‌شده از مهم‌ترین روش‌های همگام‌سازی هستند که می‌توانند به بهبود کارایی و کاهش خطاها در شبکه کمک کنند. با توجه به پژوهش یانگ و لو [46]، قبل از PL برنامه در سیستم، احتمال شناسایی گره‌های مخرب ضعیف است. به همین دلیل الگوریتم تعریف‌شده در PL ، تنها به بررسی شناسه‌های پاسخگویی به درخواست‌های هر گره کانونی محدود می‌شود.

الگوریتم ۱- ارزیابی گره های ورودی در لایه ارایه.

Algorithm 1- Evaluation of input nodes in the presentation layer.

۱. مجموعه ای از گره های ورودی در لایه اول (IN) به عنوان ورودی دریافت شود
۲. اندازه مجموعه (N) شناسایی شود
۳. شناسه مجموعه IN با ساختار $IN_i^j = \{in_1^j, in_2^j, \dots, in_s^j\}$ بررسی شود
۴. تا زمانی که $IN_i^j \neq \emptyset$ انجام بده
۵. با توجه به اولویت و ترتیب گره ها در مجموعه IN_i^j برای $i: 1, \dots, s$ انجام دهید
۶. گره کانونی شناسایی شود
۷. اگر in_i^j با اعمال شرایط $1 \leq t \leq s$ محدودیت های هزینه یا مقیاس پذیری را برآورده نکند، آن گاه؛
۸. اولویت گره in_i^j به اولویت پایین تر از خودش اصلاح شود
۹. پایان؛ اگر
۱۰. به هر گره in_i^j یک عدد صحیح t نسبت داده شود
۱۱. فاصله in_i^j تا گره کانونی در لایه اول را به صورت $d(in_i^j) = t$ نمایش بده
۱۲. اگر گره in_i^j پیام k با شرایط $k < d(in_i^j)$ از همسایگان خود دریافت کند، آن گاه؛
۱۳. گره in_i^j به شرایط $d(in_i^j) = k$ منتقل شود
۱۴. پایان؛ اگر
۱۵. گره in_i^j مجموعه $k + 1$ را به همسایگان خود ارسال می کند
۱۶. پایان

با توجه به این که در لایه اول، عملکرد گره ها در سطح زیر شبکه متعلق به آن، وارد مرحله پردازش نشده است، الگوریتم لایه اول معطوف به پارامتر مقیاس پذیری سیستم است تا جابه جایی های مورد نیاز در ترتیب فراخوانی گره ها، اعمال شود. با دریافت ایمن بودن گره ها، همگام ساز تعیین شده، این اطلاعات را به گره کانونی منتقل می کند و توسط رهبر به تمامی گره ها منتقل می شود؛ لذا از ارسال پیام های متعدد در سیستم جلوگیری می شود. به این ترتیب پیچیدگی زمانی الگوریتم (که در بهترین حالت خود $O(n^2)$ است) به $O(n)$ تقلیل می یابد. هر زمان که گره کانونی یک زیر سیستم متوجه می شود که خوشه آن ایمن است، این واقعیت را به تمام گره های خوشه ها و همچنین به رهبران زیر سیستم های مجاور خود گزارش می دهد و به این ترتیب گره های در یک زیر سیستم دلخواه وارد فاز دوم (تایید به عنوان یک گره ورودی) می شود. این دستور برای یک زیر سیستم دلخواه اجرا شده است و تعمیم آن به سایر زیر سیستم هایی که شناسه واقعی و صحیح را دریافت کرده اند، انجام می شود. شکل ۱، بررسی شناسه های صحیح را در هر زیر سیستم مستقل توسط گره رهبر (گره های سیاه) نشان می دهد که توسط یال های انتقال پیام ایجاد شده است.

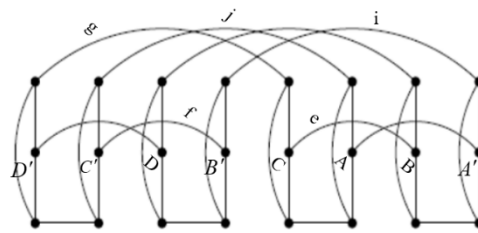


شکل ۱- همگام سازی سیستم در شناخت تصادفی پیام های صحیح.

Figure 1- System synchronization in random recognition of correct messages.

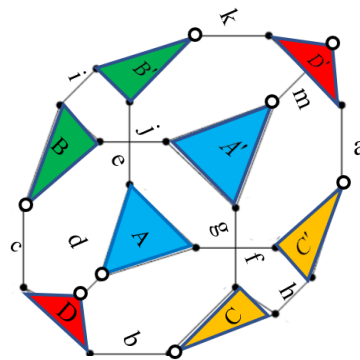
به عنوان یک مثال کاربردی، مجموعه IN_i^j در شکل ۲ از ۲۴ گره تشکیل شده است که با توجه به قرار گرفتن در زیر سیستم های مجزا، دارای شناسه های منحصر به فرد هستند. در ادامه الگوریتم شناسایی گره کانونی به روش DLE، نشان می دهد، برای تعیین یک گره کانونی در این سیستم حالت سوم (تعیین گره های مرکزی اشتراکی در هر زیر سیستم توسط گره مرکزی سیستم) نتیجه بهتری به همراه دارد؛ زیرا زیر سیستم های مشترکی که در این سیستم وجود دارد، می توانند از گره های مرکزی تقریباً مشابهی تبعیت و پیروی داشته باشند. بدون کاستن از کلیت مساله شکل ۳ به عنوان یک سیستم در

نظر گرفته می‌شود و با توجه به ترتیب و توالی در زیرسیستم‌های $A, B, C, D, A', B', C', D'$ راس هر یک از آن‌ها، به‌عنوان یک گره کانونی (گره مشخص شده) تعیین و الگوریتم PL برای هر زیرسیستم و تمامی یال‌هایی که بدون واسطه و مستقیم به آن‌ها متصل هستند، پیاده‌سازی می‌شود.



شکل ۲- اعمال پارامتر مقیاس‌پذیری.

Figure 2- Applying the scalability parameter.



شکل ۳- حمل و نقل و جابه‌جایی غیر ضروری.

Figure 3- Unnecessary transportation and movement.

با توجه به این‌که هر زیرسیستم به‌صورت مستقل به زیرسیستم‌های دیگر متصل است، یک عدد صحیح مانند t برای $d(in_t^j)$ در هر زیرسیستم وجود دارد. به‌عنوان مثال زیرسیستم A می‌تواند شناسه k را از زیرسیستم‌های D و C' و همچنین زیرسیستم A' می‌تواند شناسه k' را از زیرسیستم‌های B و D' به‌گونه‌ای دریافت کند که رابطه‌های $k < d(in_t^j)$ و $k' < d(in_t^{j'})$ برقرار باشد؛ لذا ممکن است یال‌هایی که با گره‌های مرکزی در این زیرسیستم‌ها همسایه نیستند، از مسیر پردازش حذف شوند، به همین دلیل این یال‌ها باید در زیرسیستم‌های متصل به خود، جابه‌جا شوند تا لایه ورودی بتواند بدون حذف حتی یک گره فراخوانی زیرسیستم‌ها را برای لایه دوم شبکه فراهم نماید. با در نظر گرفتن تمامی ساختار، جابه‌جایی به‌صورت زیر انجام می‌شود تا ارزیابی محدودیت‌های هزینه یا مقیاس‌پذیری برای تمامی یال‌ها به یک اندازه و بدون در نظر گرفتن مکان قرار گرفتن هر یال در یک زیرسیستم انجام شود؛ بنابراین، شرایط زیر اعمال می‌شود.

$$\begin{aligned} e: A \rightarrow B'; \quad f: C \rightarrow A'; \quad g: A' \rightarrow C & \Rightarrow e: C \rightarrow B; \quad f: C' \rightarrow B'; \quad g: D' \rightarrow C \\ h: C \rightarrow C'; \quad i: B \rightarrow B'; \quad j: B \rightarrow A' & \Rightarrow h: C \rightarrow A; \quad i: B' \rightarrow A'; \quad j: C' \rightarrow A \end{aligned}$$

با اعمال شرایط بالا، مسیرهای متنوعی برای پوشش یال‌های فراخوان‌شده در دسترس خواهد بود که یکی از این مسیرها می‌تواند به شکل ۳ ارائه شود.

گره‌های ورودی در LAL

LAL یا لایه میانی، منطق حاکم بر شرایط شبکه و پیاده‌سازی قوانین را مدیریت می‌کند. در این لایه حذف و یا کاهش سطح دخالت گره‌های مخرب و حذف واسطه‌های غیر مرتبط در مسیر تبدیل گره ورودی به گره هدف اهمیت دارد. با توجه به الگوریتم PL ، مبنای بررسی مخرب بودن یک گره این است که واکنش کمتری به رفتار و پیام‌های گره کانونی دارد. در واقع، هر چه اندازه مسیر گره $|d(in_t^j)|$ بیشتر باشد، احتمال مخرب بودن و یا سرگردان بودن یک گره بیشتر است. با توجه به این‌که ناشناخته بودن رفتار در گره‌های مخرب یک چالش محسوب می‌شود، برای سادگی ابتدا فرض

می شود، دریافت و پاسخ به گره کانونی در ساختار زمانی با هم، هم پوشانی ندارند و ممکن است یک گره پس از شناسایی و تحلیل سایر گره ها به پیام گره مرکزی پاسخ دهد. قبل از تشریح الگوریتم LAL یک الگوریتم تصادفی بر مبنای پژوهش متیور و همکاران [47]، ارایه شده است. در محاسبات توزیع شده، تصادفی سازی امکان ارایه الگوریتم های نسبتا ساده و در عین حال کارآمد را فراهم می کند. الگوریتم تصادفی بیان می کند که نیاز به فراخوان گره ها به صورت شناسه ترتیبی وجود ندارد؛ زیرا این گونه فراخوانی باعث می شود، گره های مخرب و سرگردان به پیام های ارسال شده توسط گره کانونی، واکنش های زیرکانه ای داشته باشند و با تحلیل رفتار گره های جلوتر از خود، فرآیند شناسایی مخرب بودن خود را دشوار کنند؛ لذا فراخوان و اسکن گره ها به صورت "دلخواه و تصادفی" حایز اهمیت است.

الگوریتم ۲- الگوریتم فراخوان تصادفی گره ها.

Algorithm 2- Random node calling algorithm.

۱. شناسه گره های ورودی در PL بررسی شود
۲. تا زمانی که $IN_i \neq \emptyset$ هر گره in_i توسط خط ۳ تصمیم گیری شود
۳. اگر همه همسایگان in_i با شناسه بزرگتر تصمیم گرفته باشند، آن گاه؛
۴. به فراخوان گره کانونی ملحق شوند
۵. اگر in_i نیز تصمیم گرفت به فراخوان گره کانونی بپیوندد، آن گاه؛
۶. گره in_i نمی تواند مخرب تلقی شود
۷. در غیر این صورت
۸. گره in_i برای بررسی به مجموعه B_N منتقل می شود
۹. پایان؛ تا زمانی که
۱۰. $0 < B_N \leq \frac{n}{3}$ مرحله ۲ اجرا شود
۱۱. پایان

یک زیرسیستم، در هر نقطه از اجرای الگوریتم با یک پیکربندی مجزا تعریف شده است. این پیکربندی شامل وضعیت هر گره و همه پاسخ پیام های دریافت شده و در حال انتقال (پیام ارسال شده اما هنوز دریافت نشده اند) است. به عنوان یک تعریف مستقل، در پیکربندی های ایجاد شده، اگر مقدار تصمیم مستقل از آن چه پس از آن اتفاق می افتد تعیین شود، یک-ظرفیتی نامیده می شود بدون کاستن از کلیت مساله مقادیر ورودی را به باینری بودن محدود می کنیم؛ لذا به دلیل نیاز به تعیین شناسه معتبر مقدار تصمیم هر الگوریتم اجماع نیز باینری فرض می شود. اگر یک زیرسیستم فاقد گره مخرب (عدم ارسال پیام هایی با شناسه نامعتبر و غیر تایید گره رهبر) باشد. پیاده سازی فرآیند به صورت بدیهی از شناخت گره ورودی تا تبدیل آن به گره هدف بدیهی و با پیچیدگی زمانی به مراتب کمتر، امکان پذیر است؛ لذا این بررسی از ساختار پژوهش حذف می شود. هنگامی که در یک زیرسیستم گره ها در پیکربندی اولیه هستند، همه آن ها کد اولیه خود را اجرا کرده و احتمالا برخی از پیام ها را ارسال کرده اند و اکنون منتظر رسیدن اولین پیام هستند. حال اگر یک زیرسیستم دلخواه در شبکه دارای گره مخرب باشد، هم زمان با ارسال شناسه صحیح توسط گره مرکزی به صورت مستقل شناسه هایی نامعتبر در سطح زیرسیستم و سیستم های شبکه ارسال می کند؛ لذا در بهترین حالت، این زیرسیستم حداقل دارای دو پیکربندی جدید است و توسط سایر زیرسیستم های مجاور (و حتی غیر مجاور) به صورت یک پیکربندی دو ظرفیتی شناسایی می شود. از این رو برخی از پیام های ارسال شده توسط گره های عضو زیرسیستم های مجاور پاسخ داده نمی شود و الگوریتم ایجاد شده خاتمه نمی یابد. با توجه به این که حداقل یک انتخاب از مقادیر ورودی وجود دارد به طوری که پیکربندی اولیه آن دو ظرفیتی شود، حذف پیام هایی که از سوی گره های مخرب در یک زیرسیستم دلخواه ارسال شده است ضروری و غیر قابل چشم پوشی تلقی می شود؛ لذا در گام بعد باید ساختارهایی معرفی شود که علاوه بر شناخت پیام هایی با شناسه نامعتبر، گره مخرب را نیز شناسایی و در سیستم غیرفعال نماید.

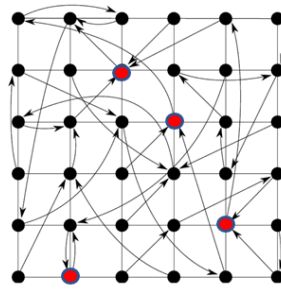
در ادامه با مشخص شدن مجموعه $B_N = \{b_{1i}^1, b_{1i}^2, \dots\}$ الگوریتم LAL برای شناسایی دقیق تر گره مخرب تشریح می شود. از آن جایی که گره مخرب نمی تواند شناسه و آدرس خود را جعل کند، این تضمین وجود دارد که تمامی پیام های ارسال شده به مجموعه IN_i^j تنها توسط $b_{1i}^j \in B_N$ پاسخ داده می شود. از طرفی اشنایدر و واتنهوفر [48] بیان داشتند، یک گره مستقل in_i^j با احتمال $p \geq \frac{1}{4d(in_i^j)}$ به فراخوان گره کانونی پاسخ می دهد؛ لذا در سطح شبکه اگر شرایط $\sum_{in_i^j \in IN_i^j} \frac{1}{2d(in_i^j)} \geq \frac{1}{6}$ برقرار باشد، یک گره صحیح شناسایی شده است.

الگوریتم ۳- الگوریتم لایه منطق.
Algorithm 3- Logic layer algorithm.

۱. الگوریتم در دوره‌های هم‌زمان گروه‌بندی شده، فرآیندهای خود را اعمال می‌کند
۲. تا زمانی که $B_N \neq \emptyset$ مرحله‌های زیر انجام شود
۳. الگوریتم تصادفی برای هر یک از گره‌های b_n انجام شود
۴. هر گره b_n با احتمال $\frac{1}{2d(in_i)}$ علامت‌گذاری می‌شود
۵. اگر همسایه درجه بالاتر b_n نیز علامت‌گذاری نشده باشد، آن‌گاه؛
۶. گره b_n با در نظر گرفتن شرط $\sum_{in_i \in N_i} \frac{1}{2d(in_i)} > \frac{1}{6}$ به عنوان یک گره خوب بررسی می‌شود
۷. پایان؛ اگر
۸. اگر همسایه درجه بالاتر b_n علامت‌گذاری شده باشد، آن‌گاه؛
۹. گره b_n باید به صورت خود اظهاری به مرحله ۳ منتقل شود
۱۰. پایان؛ اگر
۱۱. اگر همسایه‌های b_n شناسه پایین‌تری داشته باشند و به فراخوان گره کانونی پاسخ داده باشند، آن‌گاه؛
۱۲. b_n گره مخرب محسوب می‌شود
۱۳. پایان

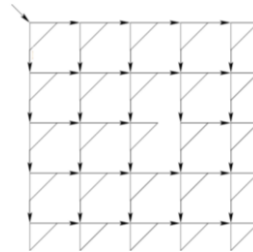
ایده اصلی الگوریتم LAL این است که گره‌ها را به صورت مکرر به مجموعه‌های کوچک‌تر و با یک برچسب یکتا تقسیم نماید. با ذخیره‌شدن این شناسه‌ها، هر گره غیرمخرب می‌داند که در حال حاضر در کدام مجموعه قرار دارد. در ابتدا، همه گره‌ها در یک مجموعه واحد قرار دارند که با رشته بیت خالی مشخص می‌شوند. سپس این مجموعه به دو مجموعه غیرخالی تقسیم می‌شود که با ۰ و ۱ مشخص می‌شوند. به همین ترتیب، همه مجموعه‌ها به طور مکرر به دو مجموعه غیرخالی تقسیم می‌شوند، تا زمانی که یک مجموعه حاوی بیش از یک گره باشد. اگر مجموعه‌ای فقط شامل یک گره باشد، این گره شناسه رایگان بعدی را دریافت می‌کند. این الگوریتم زمانی خاتمه می‌یابد که هر گره در مجموعه خود تنها باشد و مخرب بودن آن تعیین شود. توجه داشته باشید که این فرآیند پارتیشن‌بندی به طور مکرر یک درخت باینری ایجاد می‌کند که پیچیدگی زمان آن به مراتب از $O(2^n)$ نیز کم‌تر است. از آنجایی که این پارتیشن‌بندی برای هر جفت گره دلخواه در یک زیرسیستم دلخواه صدق می‌کند، شناسه پیام‌های منتقل شده با یکدیگر تداخل (انتقال هم‌زمان پیام توسط حداقل دو گره) ندارند؛ زیرا در پیکربندی اولیه شناسه‌های منحصر به فردی توسط گره کانونی ارسال شده است، گره دریافت‌کننده یک پیام می‌تواند تداخل را از گره‌ای که پیام یکتای خود را مخابره نمی‌کند، تشخیص دهد؛ لذا با توجه به این که گره مخرب هنوز تصمیمی برای ارسال پیام نگرفته است و مقدار تصمیم به ترتیب دریافت پیام‌ها یا رویدادهای خرابی بستگی دارد، فرآیند شناسایی آن انجام می‌شود. یعنی همه پیام‌ها را می‌توان به موقع و بدون ازدحام ارسال کرد؛ بنابراین، فرآیند پس از شروع در حداکثر زمان n متوقف می‌شود. در هنگام ارزیابی این الگوریتم باید توجه داشت که عملکرد مخرب سیستم با شناسایی و حذف عملکرد گره‌های مخرب تا حد زیادی کاهش یافته و هزینه‌های سیستم تنها در چگونگی ارتباط با زیرسیستم‌های دیگر معطوف شده تا اشتراک‌گذاری اطلاعات در ایجاد یک شبکه یکپارچه تکمیل شود؛ لذا انتظار می‌رود این پیچیدگی زمانی با توجه به این که زمان اجرا به حداکثر نیمی از اندازه ورودی بستگی داشته باشد، در بدترین حالت خود داری پیچیدگی زمانی $O(\log n)$ باشد که در ادامه به اثبات آن پرداخته می‌شود.

ضایعات ناشی از نقص، منجر به کار اضافی توسط اپراتورها می‌شود که با توجه به بازفرآوری محصولات و تصحیح فعالیت‌ها، کاهش بهره‌وری سیستم را به همراه دارد. در این حالت گره مرکزی پارامتر شناسایی گره‌های مخرب و سرگردان را اعمال می‌کند. هرچند در این حالت ممکن است زیرساخت‌ها کوچک شوند، اما مشارکت‌پذیری سیستم برای جبران فعالیت گره‌های حذف‌شده، خروجی سیستم را با کم‌ترین کاهش تولید همراه می‌کند. نمونه‌ای از این بررسی در شکل ۴ تا شکل ۶ نشان داده شده است.



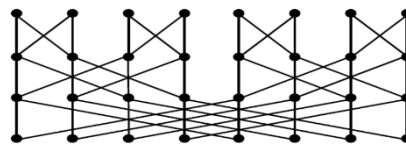
شکل ۴- ساختار کلی سیستم در زنجیره تامین.

Figure 4- Overall structure of the system in the supply chain.



شکل ۵- حذف گره های مخرب و سرگردان.

Figure 5- Removing destructive and stray nodes.

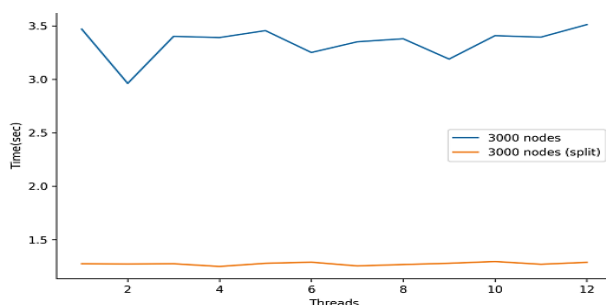


شکل ۶- مشارکت پذیری سیستم برای جبران اتصالات حذف شده.

Figure 6- System participation to compensate for deleted connections.

در این بررسی تعداد ۳۶ گره ($n = 36$) فرض شده است؛ لذا با توجه به الگوریتم شناسایی گره های مخرب، برای حداکثر ۱۲ گره ($\frac{n}{3}$) ارزیابی گره مخرب با در نظر گرفتن مقادیر ورودی و نوع شناسه های دلخواه تعیین شده، پیاده سازی شده است. با توجه به این که گره هایی با پتانسل گره مخرب بودن در مجموعه B_N قرار دارد. هر گره در این مجموعه ($b_n^i \in B_N$) با احتمال $\frac{1}{2 \times (4)} = \frac{1}{8} = 0.125$ شناسه گذاری می شود؛ لذا با وجود شرط اشاره شده در بند پنجم الگوریتم LAL ، گره هایی از مجموعه B_N حذف می شود. در این بررسی ۴ گره به عنوان گره مخرب شناسایی شده است، زیرا با توجه به الگوریتم شناسایی گره مخرب در پژوهش فریشکنخت و همکاران [49]، ارزش ورودی هر گره که با خروجی (v, x) به همه گره ها ارسال و در مجموعه S_v ذخیره شده است، باید توسط ورودی (u, y) بازخوردهای مستقلی داشته باشد که در مجموعه ای به نام S_{II} ذخیره شده است. اگر همه ورودی های (u, y) که حداقل از دو مجموعه S_{II} که شامل S_v می باشد را به عنوان عضو هایی از یک مجموعه فرضی به نام T در نظر گرفت، فقط برای گره های مشخص شده می توان کمترین ارزش ها را در نظر گرفت که به ترتیب دارای مجموعه ارزش های ورودی $\{0, 1, 0, 0\}$ است. در حالی که در سایر گره ها، ارزش های ورودی دارای اندازه حداقل ۲ هستند.

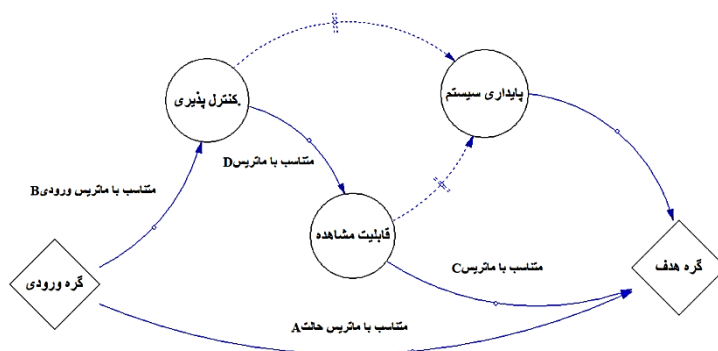
این فرآیند برای پیاده سازی برای زیرسیستم های بزرگ تر (تعداد ۳۰۰۰ گره)، به صورت شکل ۷، قابل ارایه است که نشان می دهد بعد از شناسایی گره های مخرب عملکرد یک زیرسیستم در حال هماهنگی با گره مرکزی بوده و نوسان کمی در مواجهه با چالش حذف مسیرهای ختم شده به گره های مخرب، ایجاد نموده است.



شکل ۷- عملکرد زیرسیستم پس از حذف مسیرهای مخرب.

Figure 7- Subsystem performance after removing malicious paths.

با توجه به این که DSL یا لایه پشتیبان، داده‌ها را از مجموعه $SD = \{IN_i^j - B_N\}$ فراخوان می‌کند، شناسایی و حذف گره‌های مخرب در LAL صورت گرفته است؛ لذا الگوریتم لایه سوم علی‌رغم اهمیت بالا، از یک ساختار ساده و بدون بررسی برخورداری است و به‌عنوان ارتباط‌دهنده اصلی در سطح شبکه، گره‌های ورودی تأییدشده را به سمت گره‌های هدف ارسال می‌کند. به‌طور کلی لایه سوم در محاسبات توزیع‌پذیر نیاز به ارزیابی و کنترل‌های مهم‌تری دارد، زیرا در این لایه تمامی عملکرد شبکه و زیرشبکه‌های متصل به آن ذخیره شده و به‌صورت یک پشتیبان در شبکه زنجیره‌تأمین توزیع می‌گردد. باید توجه داشت که لایه‌های اول و دوم محاسبات توزیع‌پذیر به‌صورت خودکنترلی انجام می‌شود، اما در خصوص لایه سوم این امر محقق نمی‌شود؛ زیرا گره‌های کانونی و مرکزی در این لایه دخالتی ندارند و تعاملات متعددی مانند تبادل اطلاعات بین شبکه‌های حمل و نقل، شبکه‌های بازگشت هزینه و یا شبکه‌های پشتیبان از مشتری رخ می‌دهد؛ لذا برای ترسیم الگوریتم لایه سوم در پیاده‌سازی استراتژی‌های ناب، ساختارهای تئوری کنترل مورد استفاده قرار می‌گیرد. الگوریتم‌های ارایه‌شده در این بخش با توجه به لایه سوم در محاسبات توزیع‌پذیر و متناظر با $DS = \{IN_i^j - B_N\}$ است، اما معادلات فضایی حالتی که در سیستم‌های کنترل تعریف می‌شود، به‌صورت مستقل برای هر یک از زیرسیستم‌ها ارایه می‌شود. در واقع یک شبکه زنجیره‌تأمین به‌عنوان یک معادله مرتبه n دارای n زیرسیستم از مرتبه اول است که برای سادگی، هر زیرسیستم ساختاری به شکل ۸ دارد.



شکل ۸- پیاده‌سازی پارامترهای کنترل در یک سیستم.

Figure 8- Implementing control parameters in a system.

مجموعه DS در یک تناظر با مجموعه‌ای از بردارهای $C = [c_1, c_2, \dots]_{b_i \in B}$ در ماتریس ورودی تابع حالت سیستم است.

به‌طوری‌که رابطه (۱۲) برقرار شده است.

$$[c_1] = \begin{bmatrix} in_1^i \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \dots; [c_r] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ in_r^i \end{bmatrix}, \quad \text{for all } k \neq s \rightarrow in_k^i \neq in_s^i. \quad (12)$$

یعنی شناسه هیچ دو گره‌ای یکسان نیست.

آولا و سیلوا [50]، بیان داشتند در کنترل نظارتی، اطمینان حاصل می‌شود که خروجی اندازه‌گیری‌شده برابر (یا نزدیک) ورودی مرجع است. در رد نوسان و اغتشاش، اطمینان حاصل می‌شود که اغتشاشات روی سیستم تأثیر قابل توجهی بر خروجی اندازه‌گیری‌شده ندارد. همچنین در بهینه‌سازی، بهترین مقدار خروجی اندازه‌گیری‌شده ایجاد می‌شود؛ لذا مطابق پژوهش جوزف [51] مشاهده‌پذیری را می‌توان به‌عنوان درصدی از اندازه‌گیری‌ها در سیستم در لحظه t در نظر گرفت که در زمان‌های گسسته می‌توان با استفاده از توالی اندازه‌گیری‌ها در لحظه‌های گسسته به آن دست یافت، اما در

ساختار پیوسته این فرآیند محقق نمی شود؛ لذا یک ساختار تناظر میان توالی گسسته و توالی پیوسته توسط مشتقات اندازه گیری زمان پیوسته ایجاد می شود. یعنی یک شیفت زمانی در زمان گسسته با یک مشتق در زمان پیوسته مطابقت دارد؛ بنابراین، اجرای پارامترهای مشاهده و کنترل برای همه گره های ورودی $IN_i^j = \{in_1^j, in_2^j, \dots, in_s^j\}$ در هر زیرسیستم هزینه های مضاعفی در شبکه زنجیره تامين ایجاد می کند. به همین دلیل ممکن است تنها گره هایی بررسی شوند که در سیستم حالت های ناپایدار ایجاد می کنند که این الزام منجر به تعریف پایداری و تشخیص پذیری می شود. به صورت کلی یک سیستم خطی در صورتی پایدار است که همه حالت های ناپایدار، قابل کنترل باشند. همچنین یک سیستم خطی در صورتی قابل تشخیص است که همه حالت های ناپایدار قابل مشاهده باشند. در ادامه این اتصال با در نظر گرفتن پارامتر پایداری در تئوری کنترل بررسی می شود تا متغیرهای حالت در هر سطح از تئوری کنترل مسیر گره های ورودی به گره های هدف را پوشش دهد.

هر یک از زیرسیستم ها در سطح شبکه به صورت یک ساختار مستقل حضور دارند، اما گره های ورودی برای هر زیرسیستم به صورت یک الگوریتم پیوسته برای تبدیل شدن به گره هدف، ارزیابی می شوند. الگوریتم ELD بیان می کند، علی رغم استقلال میان هر یک از زیرسیستم ها، آن ها در یک ساختار پیوسته با گره مرکزی خود در ارتباط بوده و به دنبال بهینه نمودن استراتژی های خود در هر لحظه از عملکرد سیستم هستند. یکی از دلایل مهم در تعمیم لایه سوم محاسبات توزیع پذیر به پارامتر اول سیستم های کنترلی (قابلیت مشاهده پذیری) این است که در لایه های اول و دوم گره های ورودی فقط در سطح زیرسیستمی که در آن قرار دارند ارزیابی می شوند. در واقع در هنگام پیاده سازی الگوریتم ها در این دو لایه، هیچ ارتباطی میان زیرسیستم ها وجود ندارد، اما در هنگام انتقال گره های ورودی به لایه سوم محاسبات توزیع پذیر (DSL)، ارتباط میان زیر سیستم i th و j th برقرار شده تا فراخوان اطلاعات از هریک از پایگاه های داده، بدون تکرار انجام شود؛ لذا ممکن است در هنگام تبادل اطلاعات در ساختار پایگاه پشتیبان، گره هایی با پتانسیل گره های مخرب در سطح هر زیرسیستم ایجاد شود. در واقع برای دریافت شناسه و برچسب کمتر، گره دلخواهی در سیستم j th ($in_i^j; 1 \leq t \leq s$) ممکن است شناسه گره دیگری در سیستم i th ($in_k^i; 1 \leq k \leq s$) را انتخاب نماید و با این فرآیند به طور مشخص دو گره در سیستم j th با شناسه یکسان وجود دارد و زیرسیستم با چالش های پیاده سازی استراتژی ناب مواجه خواهد شد. با توجه به قابلیت تبدیل سیستم های کنترلی مرتبه n به n th معادله مرتبه اول در معادله حالت، فرض بدون حافظه در سیستم های کنترلی حایز اهمیت است. یعنی بازخورد خروجی بر قابلیت کنترل تاثیر نمی گذارد؛ لذا ساختار محاسبات توزیع پذیر می تواند با ایجاد پارامترهای تحمل خطا، موازی کاری و تفکیک متقارن، گره های ورودی را برای تبدیل شدن به گره هدف در سیستم های کنترلی بررسی نماید؛ لذا پیاده سازی الگوریتم های بیان شده، گره های ورودی ($in_k^i; 1 \leq k \leq s$)، قبل از بررسی در سیستم های کنترلی از ۴ ویژگی مهم برخوردار هستند:

۱. با پیاده سازی الگوریتم های PL و LAL ، گره مخربی در زیرسیستم وجود ندارد؛ لذا شناسه گره ها به صورت منحصر به فرد بوده و گره ها به صورت متوالی در صف فراخوان قرار دارند.
۲. تعداد گره های وارد شده در لایه سوم ساختار محاسبات توزیع پذیر، مرتبه ماتریس C (ماتریس خروجی در معادله حالت) است.
۳. مرتبه هر زیرسیستم متناظر با ترتیب فراخوانی آن در لایه سوم است.

به عنوان مثال اولین زیرسیستم فراخوان شده از مرتبه اول است؛ لذا تابع حالت آن به صورت بدیهی به فرم $x' = ax(t) + Bu(t)$ تعریف شده است که در آن a یک عدد ثابت (ماتریس 1×1) و ماتریس B به عنوان یک بردار سطری (ماتریس $1 \times r$) است. همچنین فرم خروجی به صورت $y = Cx(t) + 0$ است که در آن ماتریس C به عنوان یک بردار ستونی (ماتریس $q \times 1$) خواهد بود. با توجه به این که ماتریس D به ورودی سیستم اجازه می دهد تا به طور مستقیم بر خروجی سیستم تاثیر بگذارد، لذا ماتریس D یک ماتریس صفر از مرتبه $q \times r$ است ($D = [0]_{q \times r}$).

۴. در هنگام فراخوانی دومین زیرسیستم، به طور مشخص بین گره های موجود در فرم خروجی $y = Cx(t) + 0$ با گره های ورودی در فرم $x' = Ax(t) + Bu(t)$ برای زیرسیستم دوم (ماتریس A از مرتبه 2×2 و ماتریس B از مرتبه $2 \times r$)، حداقل ۱ شناسه مشترک با ویژگی های زیر وجود دارد.
 - عدم فراخوان اشتباه در لایه سوم برای گره هایی با شناسه مشترک که در ۲ زیرسیستم مجزا قرار دارند.
 - عدم جابه جایی شناسه گره ها در هر دو زیرسیستم مجزا (طبق الگوریتم لایه ورودی در محاسبات توزیع پذیر، تمایل به دریافت شناسه کمتر، وجود دارد).

در ادامه با توجه به ویژگی های بیان شده، تاثیر پارامترهای کنترل پذیری سیستم بر پیاده سازی استراتژی ناب ارایه می شود.

۳-۴- قابلیت کنترل‌پذیری در ارتقای کارایی زنجیره‌تأمین

کنترل‌پذیری به توانایی کنترل رفتار یک سیستم اشاره دارد؛ بنابراین، کنترل‌پذیری خروجی به‌طور کلی به این معنی است که می‌توان خروجی سیستم دینامیکی را مستقل از بردار حالت آن هدایت نمود. به‌صورت کلی سیستم کنترل‌پذیر است اگر و تنها اگر رتبه ماتریس CO در رابطه (۱۳)، کامل باشد. یعنی رتبه برابر با تعداد حالت‌های مدل فضای حالت است.

$$Con(A, B) = [B, AB, A^2B, \dots, A^{n-1}B]. \quad (13)$$

معادل با این است که $det(Con) \neq 0$. برخی از استراتژی‌های ناب مانند موجودی اضافی در سیستم و ایجاد ارتباطات غیرضروری، ناشی از عدم کنترل در سطح شبکه زنجیره‌تأمین است که برای برطرف شدن آن می‌توان مدیریت عملکرد را پیاده‌سازی نمود. این عدم کنترل‌پذیری ممکن است به گستردگی سیستم، عملکردهای چندگانه در هر زیرسیستم و یا وجود خروجی‌های مشترک در سطح برخی از زیرسیستم‌ها ارتباط داشته باشد؛ لذا برای همگرایی کنترل‌پذیری سیستم با فرآیندهای استراتژی ناب، باید چگونگی انتقال گره‌های ورودی از لایه حافظه (لایه سوم محاسبات توزیع‌پذیر) سیستم مشخص شود. رابطه (۱۳) و همچنین شکل ۸ نشان می‌دهد، کنترل‌پذیری سیستم باید در هنگام دریافت گره‌های ورودی به سیستم انجام شود. با توجه به ویژگی‌های چهارگانه بیان‌شده الگوریتم پیشنهادی زیر ارایه شده است. این الگوریتم نیز تحت الگوریتم تصادفی ارایه‌شده در ساختار سوم، به فراخوانی گره‌ها اقدام می‌کند.

الگوریتم ۴- الگوریتم فراخوان تصادفی گره‌ها.

Algorithm 4- Random node calling algorithm.

۱. الگوریتم برای زیر سیستم j th ($in_j^1; 1 \leq t \leq s$) و زیر سیستم i th ($in_i^1; 1 \leq k \leq s$) به‌صورت هم‌زمان اجرا شود
۲. تا زمانی که $in_k^1 \neq in_s^1$ مراحل زیر انجام شود
۳. الگوریتم تصادفی برای هر یک از بردارهای $b_i \in [B]$ انجام شود
۴. هر بردار b_i با شناسه معرفی‌شده در رابطه (۱۱) علامت‌گذاری می‌شود
۵. تا زمانی که $B_1 \cap B_j \neq \emptyset$
۶. اگر شناسه دریافت‌شده به مجموعه B_1 تعلق دارد، آن‌گاه؛
۷. ماتریس CO_i تشکیل شود
۸. پایان؛ اگر
۹. اگر اگر شناسه دریافت‌شده به مجموعه B_j تعلق دارد، آن‌گاه؛
۱۰. ماتریس CO_j تشکیل شود
۱۱. در غیر این صورت
۱۲. شناسه in_k^1 حذف و به مرحله ۱ منتقل شود
۱۳. پایان؛ اگر
۱۴. تا زمانی که $CO_i \cap CO_j \neq \emptyset$ فراخوان ساختار به مرحله ۱ منتقل شود
۱۵. پایان

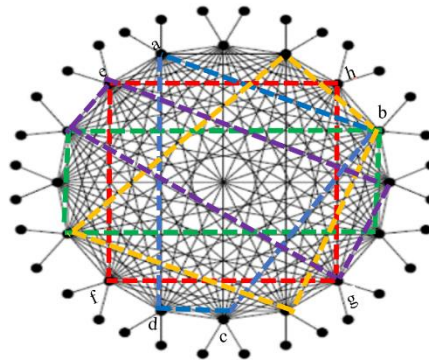
با توجه به الگوریتم، آزمون کالمن برای زیر سیستم j th ($in_j^1; 1 \leq t \leq s$) و زیر سیستم i th ($in_i^1; 1 \leq k \leq s$) به‌صورت مجزا انجام می‌شود.

ادعا می‌کنیم با پیاده‌سازی الگوریتم یادشده رابطه (۱۴) برقرار بوده و الگوریتم ضمن پیاده‌سازی، خاتمه‌پذیر است.

$$det(C_O) \neq 0, \quad \text{for all } i \neq j. \quad (14)$$

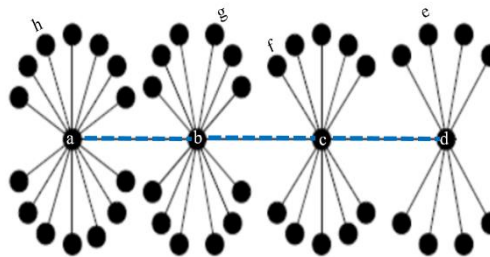
اثبات: این بررسی در ساختار خطی به‌راحتی قابل مشاهده است؛ زیرا با توجه به ویژگی اول، هر یک از زیرسیستم‌هایی که در اولین مرحله فراخوان شوند، رابطه $x' = ax(t) + Bu(t)$ برقرار است. از طرفی ماتریس $B \neq 0$ و به‌عنوان یک بردار سطری (ماتریس $1 \times q$) عمل می‌کند. همچنین با توجه به این که بردار حالت در اولین فراخوان یک عدد اسکالر است در نتیجه $det(C_O) \neq 0$ در فراخوان m th و با توجه به $x' = Ax(t) + Bu(t)$ مرتبه ماتریس C در فراخوان $m-1$ انتخاب می‌شود. با توجه به این که ماتریس B ورودی و ماتریس C خروجی تابع حالت را نشان می‌دهد و می‌دانیم که $det(C_O) \neq 0$, for all $i \neq j$ به‌طور مشخص؛ لذا به‌طور مشخص B_j یا B_i از یکی $e^{-Ax \times in_j^k} x(in_i^k) - e^{-Ax \times t_0} x(t_0)$ تعلق دارد؛

نمونه‌ای از این بررسی در شکل‌های ۹ و ۱۰ نشان داده شده است.



شکل ۹- موجودی اضافی در سیستم و ایجاد ارتباطات غیرضروری.

Figure 9- Excess inventory in the system and the creation of unnecessary connections.



شکل ۱۰- مدیریت عملکرد سیستم.

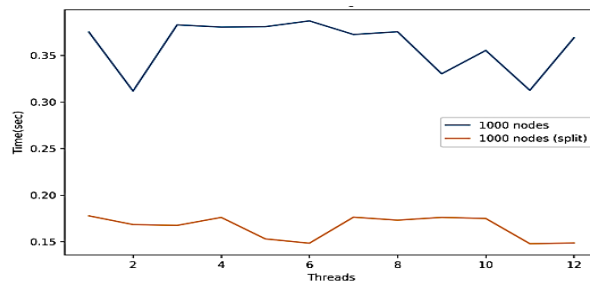
Figure 10- System performance management.

برای بررسی الگوریتم قابلیت کنترل شبکه از ارتباطات در ساختار یک سیستم با تعداد ۴۸ گره (عضو) بررسی شده است که برای سهولت در ارایه ساختار الگوریتم فرض شده است، سیستم کلی از ۱۶ زیرسیستم مجزا (هر زیرسیستم دارای ۳ عضو)، تشکیل شده است. ویژگی‌های ۴ گانه ارایه شده برای گره‌های ورودی بیان می‌کند با پیاده‌سازی الگوریتم‌های LAL و PL برنامه گره‌های مخرب فعال از سیستم حذف شده است و عضوها، فاقد شناسه نامشخص و فریب‌کارانه هستند. با توجه به این‌که اولین زیرسیستم‌های فراخوان شده از مرتبه اول هستند، یک رابطه خطی میان فراخوان‌های انجام شده وجود دارد؛ لذا طبق ویژگی چهارم، میان زیرسیستم‌ها، حداقل یک شناسه مشترک وجود دارد که با توجه حذف گره‌های مخرب و سرگردان، ارتباط میان زیرسیستم‌ها را برقرار نموده است. طبق الگوریتم مشارکت‌پذیری سیستم، پارامتر $d(in_i^j)$ به عنوان ایجاد مسیر بسته میان زیرسیستم‌ها از اندازه ۴ انتخاب می‌شود $|d(in_i^j)| = 4$ که در شکل ۸، مجموعه‌ای از مسیرهای ممکن مشخص شده است. بند اول الگوریتم، مسیرهای $abcda$ ($1 \leq t \leq 4$) و $efghe$ ($1 \leq k \leq 4$) را به صورت مجزا فراخوان می‌کند. با در نظر گرفتن شرط $in_k^i \neq in_k^j$ مسیرهای مورد انتظار ۴ گانه‌ای به صورت زیر، ترسیم می‌شود.

$$\begin{cases} 1. |d(a)| = 0 \rightarrow |d(ef)| = 1 \text{ or } |d(ab)| = 1 \rightarrow |d(e)| = 0, \\ 2. |d(ab)| = 1 \rightarrow |d(efg)| = 2 \text{ or } |d(abc)| = 2 \rightarrow |d(ef)| = 1, \\ 3. |d(abc)| = 2 \rightarrow |d(efgh)| = 3 \text{ or } |d(abcd)| = 3 \rightarrow |d(efg)| = 2, \\ 4. |d(abcd)| = 3 \rightarrow |d(efghe)| = 4 \text{ or } |d(abcda)| = 4 \rightarrow |d(efgh)| = 3. \end{cases}$$

با توجه به الگوریتم انتخاب تصادفی، مسیرهای مشخص شده ارزیابی می‌شود. طبق رابطه (۱۵)، با توجه به این‌که مسیرهای بسته مشخص شده در ساختار هر یک از زیرسیستم‌ها، از قبل فراخوان شده است، دورهای تکراری حذف می‌شود؛ لذا طول مسیر از ۴ به ۳ کاهش می‌یابد. از طرفی تا زمانی که اشتراک میان هر یک از بردارهای حالت ایجاد شده از زیرسیستم‌ها وجود دارد، مسیر مشخص $abcd$ می‌تواند هر یک از مراحل چهارگانه بالا را به صورت مجزا فراخوان و شناسه‌گذاری نماید. اگر هر یک از عضوهای a, b, c, d گره‌های مرکزی و کانونی باشند، عضوهای e, f, g, h مسیرهای منحصر به فردی مانند ah, bg, cf, de را برای گره مرکزی ایجاد می‌کند. از طرفی تمام مسیرهایی که شناسه عضوهای a, b, c, d و شناسه عضوهای e, f, g, h را فراخوان کرده است به عنوان یک مسیر تکراری حذف شده تا به همه عضوها یک مسیر منحصر به فرد با شناسه یکتا نسبت داده شود. همچنین

این بررسی برای یک سیستم بزرگ (۱۰۰۰ گره) به صورت شکل ۱۱ نمایش داده می‌شود که در آن سطح پاسخگویی، با توجه به ساختار ارایه‌شده تا حد امکان کاهش یافته است.



شکل ۱۱- عملکرد زیرسیستم پس از اعمال مدیریت عملکرد.

Figure 11- Subsystem performance after applying performance management.

۴-۴- قابلیت مشاهده‌پذیری در ارتقای کارایی زنجیره‌تأمین

مشاهده‌پذیری بیان می‌کند، در زیرسیستمی که در حال کنترل است، چه فرآیندهایی در حال رخ دادن بوده و چه خروجی‌هایی ایجاد می‌شود؛ لذا آگاهی از تمامی فرآیندها، به عنوان مشاهده‌پذیری مطرح شده است. برخی از استراتژی‌های ناب مانند پردازش بیش از اندازه در سیستم به شناخت وضعیت سیستم در هر لحظه وابسته است که برای برطرف شدن آن می‌توان یک تعادل پردازش ایجاد نمود؛ لذا برای همگرایی مشاهده‌پذیری سیستم با فرآیندهای استراتژی ناب، باید انتقال گره‌های ورودی به گره‌های هدف انجام و خروجی سیستم مشخص شود. به منظور پیاده‌سازی استراتژی‌های ناب در پارامتر مشاهده‌پذیری سیستم کنترل، از آزمون کالمن در رابطه (۱۵) استفاده شده است. اگر $det(Q_0) \neq 0$ ، مشاهده‌پذیری سیستم وجود دارد.

$$Obs(A, C) = [C^T, A^T C^T, (A^2)^T C^T, \dots, (A^{n-1})^T C^T]^T. \quad (15)$$

پس از پیاده‌سازی پارامتر کنترل‌پذیری عملاً گره‌های ورودی توسط لایه حافظه و ذخیره، قابل مشاهده هستند؛ لذا برچسب هر یک از گره‌های ورودی با توجه به میزان عملکرد آن‌ها در سطح سیستم ارایه می‌شود. برای انتقال گره‌های ورودی به گره هدف در پارامتر مشاهده‌پذیری باید تابع تبدیل مناسبی ایجاد شود تا گره‌های مرجع $in_f^h(s)$ در لایه سوم محاسبات توزیع‌پذیر، مجموعه $out_f^h(s)$ را از مجموعه گره‌های $in_s^i(s)$ و $in_k^i(s)$ تولید نماید. در واقع به صورت کلی همه ورودی‌ها (بعد از اعمال کنترل‌پذیری سیستم) از ماتریس ورودی B دریافت و فراخوان شده است. از طریق پیاده‌سازی الگوریتم کنترل‌پذیری سیستم بر روی مجموعه بردارهای $[B]$ مقدارهای مرجع $in_f^h(s)$ به طور متناوب بررسی می‌شود. در واقع کنترل‌پذیری با شناسایی تعداد پردازش‌های غیر مفید (پنهان شدن گره‌های مخرب) اعمال و خطای ε به عنوان تفاوت میان تعداد گره‌های مرجع و تعداد گره‌های خروجی $(|in_f^h(s) - out_f^h(s)|)$ تعیین می‌شود تا ورودی‌های $in_s^i(s)$ را به مجموعه گره‌های پردازش‌شده $out_f^h(s)$ تغییر دهد. فرآیندهای $in_k^i(s)$ و $in_s^i(s)$ قبل نسبت داده شدن به فرآیند $out_f^h(s)$ باید به عنوان ترکیب خطی از $in_f^h(s)$ معرفی شوند؛ لذا مشاهده‌پذیری گره‌های هدف $out_f^h(s)$ با رابطه (۱۶) پشتیبانی می‌شود.

$$out_f^h(s) = C(sI - in_s^i)^{-1} in_k^i + in_f^h. \quad (16)$$

در واقع به ازای هر تابع حالت دلخواهی مانند $\hat{x}(t)$ ، فرآیندهای زمانی متوالی می‌تواند بدون تغییر $in_f^h(t)$ را به $out_f^h(t)$ تبدیل نماید و $\hat{x}(t+1)$ را به صورت رابطه (۱۷) ایجاد نماید.

$$x(t+1) = in_f^h(t)x(t) + in_f^h(t)(in_s^i(t+1) - out_f^h(t) \times in_f^h(t) \times x(t)). \quad (17)$$

رابطه‌های (۱۶) و (۱۷)، مبنای الگوریتم مشاهده‌پذیری در استراتژی ناب خواهد بود که در ادامه تشریح می‌شود. این الگوریتم نیز تحت الگوریتم تصادفی ارایه‌شده، به فراخوانی گره‌ها اقدام می‌کند.

الگوریتم ۵- الگوریتم مشاهده پذیری در استراتژی ناب.
Algorithm 5- Observability algorithm in lean strategy.

۱. به صورت تصادفی گره f در زیرسیستم hth به عنوان اولین گره، شناسه (۱) داده شود
۲. تا زمانی که $in_k^i(t) \neq \emptyset$ ، فراخوان $in_f^h(t)$ از گره های ورودی $in_k^i(t)$ انجام شود
۳. تا زمانی که $in_s^i(t) \neq \emptyset$ ، فراخوان $in_f^h(t)$ از گره های ورودی $in_s^i(t)$ انجام شود
۴. تا زمانی که $in_f^h(t) \neq \emptyset$ ، فراخوان $out_f^h(t)$ انجام شود
۵. $out_f^h(t)$ تحت تناظر $in_f^h + C(sI - in_s^i)^{-1}in_k^i$ به $out_f^h(s)$ تبدیل شود
۶. تبدیل $x(t)$ تحت تناظر $t \rightarrow t+1$ به $in_s^i(t)(in_s^i(t+1) - out_f^h(t) \times in_f^h(t) \times x(t)) + in_f^h(t)x(t)$ تبدیل شود
۷. پایان

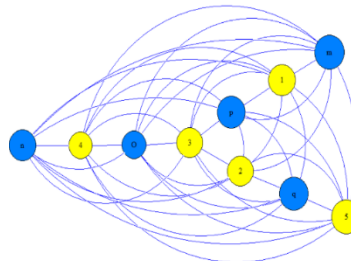
با توجه به الگوریتم، آزمون کالمن متناظر با رابطه (۱۸)، است.

$$Obs(in_f^h(t), out_f^h(t)) = [(out_f^h(t))^T, (in_f^h(t))^T(out_f^h(t))^T, \dots, ((in_f^h(t))^{n-1})^T(out_f^h(t))^T]^T. \quad (18)$$

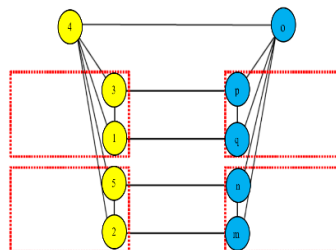
در ادامه ادعا می کنیم با برقراری شرط $det(Obs) \neq 0$ ، الگوریتم ارایه شده خاتمه پذیر است.

اثبات: این ادعا با توجه به رابطه (۱۵)، در هنگام تعیین مقادیر ویژه ساختار $out_f^h(s)$ به سادگی قابل اثبات است؛ زیرا با توجه به الگوریتم ارایه شده، فرآیند تابع تبدیل یک تناظر یک به یک میان متغیرهای t و s ایجاد می کند. از طرفی مجموعه های $in_f^h(t)$ و $out_f^h(t)$ از الگوریتم ارایه شده در کنترل پذیری پشتیبانی شده است و در این ساختار نیز باید رابطه $e^{-Ax}in_f^h(t) - e^{-Ax}x(t_0) - T(s)$ به ساختار $T(s)$ تعلق داشته باشد تا در الگوریتم مشاهده پذیری مرحله سوم و مرحله چهارم ادامه یابد. همچنین تحت تبدیل $t \rightarrow t+1$ ماتریس هایی با مقادیر ویژه مخالف صفر ایجاد می شود که مرتبه ماتریس Obs را با مرتبه ماتریس های کلی A و B برابر نشان می دهد؛ لذا این ماتریس به عنوان بزرگترین ماتریس با دترمینان مخالف صفر است.

نمونه ای از این بررسی در شکل های ۱۲ و ۱۳ نشان داده شده است. ضایعات ناشی از پردازش بیش از حد، به این دلیل اتفاق می افتد که ارزش افزوده بیشتری نسبت به چیزی که مشتری نهایی نیاز دارد به محصول داده می شود. در واقع ارتباط میان مجموعه گره هایی از دو زیرسیستم مجزا به صورت غیر ضروری اتفاق می افتد. در این حالت پارامتر تعادل پردازش در محاسبات توزیع پذیر اعمال شده و قابلیت ارتجاعی برای سیستم اعمال می شود.



شکل ۱۲- پردازش بیش از اندازه در سیستم.
Figure 12- Overprocessing in the system.



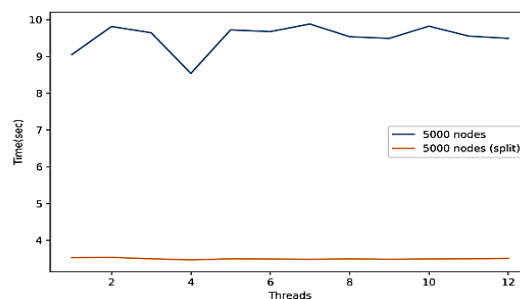
شکل ۱۳- تعادل پردازش با تعیین گره کانونی.
Figure 13- Processing balancing with focal node determination.

گره‌های بررسی شده با توجه به برقراری شرایط الگوریتم‌های LAL ، PL و اجرای الگوریتم کنترل‌پذیری، مبنای تعیین گره کانونی اولیه در مجموعه‌ای که قابلیت مشاهده‌پذیری دارد، این است که گره موردنظر باید نسبت به سایر گره‌ها، ارتباط و مسیرهای بیشتری داشته باشد. به عنوان یک نمونه، مجموعه‌های $in_k^i(s) = \{1,2,3,4,5\}$ برای تعیین مجموعه $out_f^h(s)$ در نظر گرفته شده است. با توجه به این که $|in_s^i(s)| = 5$ ، در مجموعه $in_s^i(s)$ گره "0" و در مجموعه $in_k^i(s)$ گره ۴ به عنوان گره مرکزی تعیین شده است. برای تعیین مجموعه $out_f^h(s)$ به فراخوان مجموعه $in_f^h(t)$ نیاز است که با توجه به الگوریتم کنترل‌پذیری اعمال شود. با فراخوان مستقل این دو مجموعه، مسیرهای تکراری، حذف و زیرسیستم‌های مجزایی به اندازه ۲ برای $in_f^h(t)$ ایجاد می‌شود. در واقع گره مرکزی از مجموعه $in_s^i(s) = \{m,n,o,p\}$ با مسیری به اندازه ۲ به مجموعه $in_k^i(s) = \{1,2,3,4,5\}$ متصل می‌شود (و بر عکس)؛ بنابراین، شناسه تعیین مجموعه $out_f^h(t)$ برای هر دو مجموعه با اعمال خطای ε $|in_f^h(t) - out_f^h(t)| = 0$ به عبارت دیگر:

$$\begin{cases} |op3| = |oq1| = |on5| = |om2| = 2 \\ |43p| = |41q| = |45n| = |42m| = 2 \end{cases} \Rightarrow |d(4o)| = 1.$$

لذا اندازه مسیر گره‌های مرکزی در دو مجموعه $in_s^i(s)$ و $in_k^i(s)$ یک می‌باشد و تحت یک تناظر $1 \rightarrow 1$ ، $out_f^h(s)$ به $out_f^h(t)$ تبدیل شده است.

با اعمال شرایط و فرآیندهای پیوسته و با افزایش سطح گره‌ها در یک زیرسیستم، قابلیت مشاهده‌پذیری سیستم می‌تواند بسیاری از نوسانات منجر به کاهش کارآمدی زنجیره تامین را تا حد امکان کاهش داده و مسیر تحقق استراتژی‌های ناب را با توجه به لایه سوم محاسبات توزیع پذیر، فراهم نماید.



شکل ۱۴- عملکرد زیر سیستم پس از اعمال مدیریت عملکرد.

Figure 14- Subsystem performance after applying performance management.

۵- نتیجه گیری

شبکه زنجیره تامین کارآمد که از استراتژی‌های ناب استفاده می‌کند، می‌تواند با جریان بهینه اطلاعات، تصمیم‌گیری سریع‌تر و دقیق‌تری اعمال کند. پیاده‌سازی استراتژی‌های ناب در کنار یک شبکه زنجیره تامین کارآمد، می‌تواند موجودی‌ها را به حداقل برساند. در واقع استفاده از مدل‌های پیش‌بینی تقاضا و سیستم‌های اطلاعاتی می‌تواند کمک کند تا موجودی‌ها بر اساس نیاز واقعی بازار و پیش‌بینی دقیق تقاضا بهینه‌سازی شوند و با استراتژی تولید به موقع^۱ مواد و محصولات فقط در زمانی که نیاز به آن‌ها است، تامین شوند. این امر باعث کاهش موجودی‌های اضافی و بهبود گردش مواد در زنجیره تامین می‌شود.

زنجیره تامین مدرن به شدت به ارتباطات سریع و هم‌زمان بین تمام اعضای آن وابسته است. با استفاده از محاسبات توزیع پذیر، اطلاعات به صورت هم‌زمان در سراسر شبکه زنجیره تامین به اشتراک گذاشته می‌شود. این ارتباطات لحظه‌ای و به هنگام باعث می‌شود اجزای ساختاری در خط تولید و ذی‌نفعان، هم‌زمان از تغییرات در تقاضا، موجودی‌ها یا وضعیت تولید مطلع شوند.

محاسبات توزیع پذیر با به کارگیری منابع پردازشی در محل‌های مختلف، امکان پردازش داده‌ها در زمان واقعی و به طور هم‌زمان را فراهم می‌آورد و باعث بهبود کارایی، کاهش هزینه‌ها، بهینه‌سازی فرآیندها و افزایش پایداری و انعطاف‌پذیری زنجیره تامین می‌شود. این رویکرد به ویژه در شرایطی که پیچیدگی‌ها و تغییرات محیطی در زنجیره‌های تامین بیشتر شده‌اند، اهمیت ویژه‌ای دارد و سازمان‌ها را قادر می‌سازد که با سرعت و دقت بیشتری

¹ Just-in-time

به تغییرات واکنش نشان دهند. رویکرد محاسبات توزیع پذیر در یک زنجیره تامین با توجه به مقیاس پذیری و توانایی دریافت و پردازش داده ها و همچنین امکان ارسال بدون تاخیر داده برای ایجاد تصمیم گیری های هوشمند، یکی از گزینه های موثر در ایجاد استراتژی های ناب در سیستم است.

محاسبات توزیع پذیر منطقه ای (در هر زیرسیستم)، پردازش داده ها را به جای ارسال به یک سیستم مرکزی به صورت محلی انجام می دهد؛ لذا حسگرها هوشمندتر می شوند و قدرت محاسباتی کافی برای شناسایی و پردازش داده ها در هنگام دریافت و پاسخ مناسب را دارند. رویکرد محاسبات توزیع شده در زنجیره تامین، دسترسی به حجم زیادی از داده ها را با ذخیره سازی در گره مرکزی زیرسیستم ها، فراهم می کند؛ لذا دسترسی آسان تر و شفاف تر به داده ها، انعطاف پذیری و پاسخ گویی بیشتر را امکان پذیر می کند و در دراز مدت هزینه های مرتبط با مدیریت و نگهداری را کاهش می دهد. همچنین با اشتراک گذاری و اتصال داده های تمام زیرسیستم ها به یکدیگر، برنامه ریزی و مدیریت بهینه را فراهم می کند. در واقع امکان ادغام و هماهنگی داده ها در هر یک از زیرسیستم ها، عملکرد فرآیندهای مخرب را در کم ترین زمان، مدیریت و تغییرات مخرب در شبکه زنجیره تامین خود تا حد زیادی خنثی می کند.

برای جلوگیری از بروز مشکلاتی مانند تداخل یا تعارض در داده ها، از الگوریتم های هم زمان سازی استفاده می شود. این الگوریتم ها به شبکه اجازه می دهند که تغییرات در یک بخش از زنجیره تامین به سرعت به سایر بخش ها منتقل شده و هم زمان به روزرسانی های لازم در سیستم انجام شوند. داده های زنجیره تامین در مکان های مختلف (تامین کنندگان، انبارها، توزیع کنندگان و مشتریان) ذخیره می شوند و با استفاده از محاسبات توزیع پذیر، می توان این داده ها را به طور هم زمان در چندین گره پردازش کرده و به تحلیل های سریع تر و بهینه تری رسید.

الگوریتم های ارایه شده در هر یک از لایه های دستیابی به اهداف برای ارتقای کارآمدی زنجیره تامین ناب مبتنی بر شبکه محاسباتی توزیع شده، از مزایای پردازش موازی و بهینه سازی منابع استفاده می کند. این الگوریتم با ترکیب مدل های پیش بینی تقاضا، برنامه ریزی خطی برای بهینه سازی موجودی، هماهنگ سازی داده ها و یادگیری تقویتی می تواند به طور مداوم کارایی زنجیره تامین را بهبود دهد و هدررفت ها را کاهش دهد. نتایج بررسی نشان داد، پارامتر کنترل پذیری که با فراخوان گره های ورودی از لایه سوم فرآیند محاسبات توزیع پذیر انجام می شود، می تواند در حذف و کاهش قابل ملاحظه اتلاف ها در سطح شبکه زنجیره تامین موثر باشد به گونه ای که گره های ورودی در هر یک از زیرسیستم های ایجاد شده در سطح شبکه زنجیره تامین با پیاده سازی الگوریتم پیشنهادی، از هر گونه عملکرد نادرست برای ایجاد اتلاف هایی مانند پردازش بیش از اندازه در سیستم، تولید زباله قبل از مصرف و همچنین ضایعات ناشی از نقص، جلوگیری می کند و یا در صورت وجود این اتلاف ها، زیرسیستم های کوچک تری برای حذف گره های مخرب جدید که از اتصال زیرسیستم ها به یکدیگر (برای اشتراک گذاری حافظه پنهان) ایجاد شده است، تولید می کند. به صورت کلی الگوریتم پیشنهادی در پارامتر کنترل پذیری، هم راستا با الگوریتم *PL* در ساختار محاسبات توزیع پذیر عمل می کند. در ادامه الگوریتم مشاهده پذیری در یک ساختار جدید و با توجه به گره های ورودی حاصل از اتصال زیرسیستم ها به یکدیگر، معرفی و عملکرد آن تشریح شد. بررسی الگوریتم مشاهده پذیری نشان داد، تحت تابع تبدیل ارایه شده، می توان از اتلاف های مهمی در چارچوب حمل و نقل و جابه جایی های غیر ضروری و انتظار در محل تولید جلوگیری نمود. در واقع هنگامی که گره های ورودی به صورت متوالی و پیوسته از اشتراک گذاری لایه سوم زیرسیستم های مشترک در بخش حافظه پنهان، فراخوان می شوند. به طور کامل از چارچوب و هنجارهای ارایه شده در سیستم پیروی می کند و تحت یک تابع تبدیل، مبتنی بر همان گره های ورودی که در طول زمان به یک سیستم خطی تغییرناپذیر متصل می شود، هیچ عملکرد نامطلوبی به گره های هدف منتقل نمی شود. در واقع، تمامی گره هایی ورودی به لایه سوم، بدون حذف به گره های هدف تبدیل شده و وظایف ایجاد شده در هر زیرسیستم بدون وقفه و کاهش ابعاد انجام می شود. هنگامی که ساختار محاسبات توزیع پذیر، مبتنی بر استراتژی ناب در سطح در شبکه زنجیره تامین کارآمد شده پیاده سازی گردد، فعالیت های هم زمانی متعددی می تواند در سطح سیستم انجام شود به گونه ای که استقلال آن ها از هزینه های اضافی برای تعیین وابستگی یا عدم وابستگی جلوگیری می کند.

در این پژوهش با بهره مندی از ساختار و لایه های ۳ گانه محاسبات توزیع پذیر، چالش های پیش رو در مسیر ناب بودن یک استراتژی در زنجیره تامین کارآمد، مورد بررسی قرار گرفت. بررسی ها نشان داد، مبتنی بر لایه اول محاسبات توزیع پذیر، گره های ورودی در سیستم باید با دریافت یک شناسه مستقل، نوع عملکرد و نحوه فعالیت خود را در سطح شبکه ای که به آن تعلق دارند، بیان نمایند. لذا گره ای که تمایلی به تعریف عملکرد خود در ساختار یک شبکه نداشته باشند، پتانسیل تبدیل شدن به یک گره مخرب و سرگردان را دارد. به صورت نسبی می توان گفت لایه اول در محاسبات توزیع پذیر می تواند مقیاس پذیری سیستم را کنترل نماید و از جابه جایی بیهوده فرآیندها قبل و بعد از تولید جلوگیری نماید. در لایه دوم محاسبات

توزیع‌پذیر گره‌هایی که ظرفیت تبدیل شدن به گره مخرب دارند، به طور دقیق شناسایی و برای دریافت مجدد شناسه به لایه اول فراخوان می‌شود. در صورتی که این گره به دریافت شناسه مستقل پاسخ ندهد، عملاً از مجموعه‌ای که به آن تعلق دارد، حذف می‌شود. اما با توجه به پارامتر مستقل بودن عملکرد هر شبکه با شبکه‌های دیگر، خللی در عملکرد سایر سیستم رخ نمی‌دهد و سایر عضوهای همان شبکه عمل گره‌های مخرب را پوشش می‌دهند. یکی از مهم‌ترین مزایای این الگوریتم، حذف زباله‌ها قبل از تولید و همچنین کاهش محسوس ضایعات ناشی از نقص در سیستم است که با مشارکت‌پذیری سیار گره‌ها در هر زیرسیستم همراه است. لایه سوم در ساختار محاسبات توزیع‌پذیر، به عنوان مهم‌ترین فرآیند هر یک از شبکه‌های متصل به شبکه مادر محسوب می‌شود. در این لایه هیچ گره مخربی در سطح زیرسیستم‌ها وجود ندارد. لذا با توجه به ایجاد تعادل در پردازش زیرسیستم‌ها، تولیدی بیش از نیاز در یک زیرسیستم انجام نمی‌شود و تبع آن، ضایعات ناشی از پردازش بیش از اندازه تولید نخواهد شد. اما در هنگام اتصال زیرسیستم‌ها به یکدیگر ممکن است عملکرد برخی از گره‌ها با شناسه واحد گره رهبر سیستم در تضاد باشد. پارامتر کنترل‌پذیری که با فراخوان گره‌های ورودی از لایه سوم انجام می‌شود، می‌تواند در حذف و کاهش قابل ملاحظه اتلاف‌ها در سطح شبکه زنجیره‌تأمین موثر باشد به گونه‌ای که گره‌های ورودی در هر یک از زیرسیستم‌های ایجادشده، از هرگونه عملکرد نادرست، جلوگیری و یا در صورت وجود این اتلاف‌ها، ساختار زیرسیستم‌های کوچک‌تری برای حذف گره‌های مخرب جدید که از اتصال زیرسیستم‌ها به یکدیگر (برای اشتراک‌گذاری حافظه پنهان) ایجاد شده است، تعریف می‌کند. در ادامه الگوریتم مشاهده‌پذیری نشان داد، تحت تابع تبدیل ارایه‌شده، می‌توان از اتلاف‌های مهمی در چارچوب حمل‌ونقل عملکرد آن تشریح شد. بررسی الگوریتم مشاهده‌پذیری نشان داد، تحت تابع تبدیل ارایه‌شده، می‌توان از اتلاف‌های مهمی در چارچوب حمل‌ونقل و جابه‌جایی‌های غیرضروری و انتظار در محل تولید جلوگیری نمود. در واقع هنگامی که گره‌های ورودی به صورت متوالی و پیوسته از اشتراک‌گذاری لایه سوم زیرسیستم‌های مشترک در بخش حافظه پنهان، فراخوان می‌شوند. به طور کامل از چارچوب و هنجارهای ارایه‌شده در سیستم پیروی می‌کند و تحت یک تابع تبدیل، مبتنی بر همان گره‌های ورودی که در طول زمان به یک سیستم خطی تغییرناپذیر متصل می‌شود، هیچ عملکرد نامطلوبی به گره‌های هدف منتقل نمی‌شود. در واقع، تمامی گره‌های ورودی به لایه سوم، بدون حذف به گره‌های هدف تبدیل شده و وظایف ایجادشده در هر زیرسیستم بدون وقفه و کاهش ابعاد انجام می‌شود.

منابع

- [1] Wamba, S. F., Dubey, R., Gunasekaran, A., & Akter, S. (2020). The performance effects of big data analytics and supply chain ambidexterity: The moderating effect of environmental dynamism. *International journal of production economics*, 222, 107498. <https://doi.org/10.1016/j.ijpe.2019.09.019>
- [2] Bortolini, M., Calabrese, F., Galizia, F. G., & Mora, C. (2022). A three-objective optimization model for mid-term sustainable supply chain network design. *Computers & industrial engineering*, 168, 108131. <https://doi.org/10.1016/j.cie.2022.108131>
- [3] Hashemi-Amiri, O., Ghorbani, F., & Ji, R. (2023). Integrated supplier selection, scheduling, and routing problem for perishable product supply chain: A distributionally robust approach. *Computers & industrial engineering*, 175, 108845. <https://doi.org/10.1016/j.cie.2022.108845>
- [4] Wu, G., de Carvalho Servia, M. Á., & Mowbray, M. (2023). Distributional reinforcement learning for inventory management in multi-echelon supply chains. *Digital chemical engineering*, 6, 100073. <https://doi.org/10.1016/j.dche.2022.100073>
- [5] Lee, K. J., Tilling, K. M., Cornish, R. P., Little, R. J. A., Bell, M. L., Goetghebeur, E., ... & Carpenter, J. R. (2021). Framework for the treatment and reporting of missing data in observational studies: The treatment and reporting of missing data in observational studies framework. *Journal of clinical epidemiology*, 134, 79–88. <https://doi.org/10.1016/j.jclinepi.2021.01.008>
- [6] Dong, F., Jin, D., Zhao, X., Han, J., & Lu, W. (2022). A non-cooperative game approach to the robust control design for a class of fuzzy dynamical systems. *ISA transactions*, 125, 119–133. <https://doi.org/10.1016/j.isatra.2021.06.031>
- [7] Decardi-Nelson, B., & Liu, J. (2023). Computing control invariant sets of cascade nonlinear systems: Decomposition and distributed computing. *Computers & chemical engineering*, 171, 108142. <https://doi.org/10.1016/j.compchemeng.2023.108142>
- [8] Gomes, K. R. R., Perera, H. N., Thibbotuwawa, A., & Sunil-Chandra, N. P. (2023). Comparative analysis of lean and agile supply chain strategies for effective vaccine distribution in pandemics: A case study of COVID-19 in a densely populated developing region. *Supply chain analytics*, 3, 100022. <https://doi.org/10.1016/j.sca.2023.100022>
- [9] Negi, S. (2021). Supply chain efficiency framework to improve business performance in a competitive era. *Management research review*, 44(3), 477–508. <https://doi.org/10.1108/MRR-05-2020-0272>
- [10] Strimovskaya, A., Barykin, S., Volkova, E., Tsyplakova, E., Sinko, G., Kuzmenkova, V., & Krasilnikov, A. (2023). Multi-level conceptual model of efficiency control in supply chain management. *IFAC-papersonline*, 56(2), 1809–1814. <https://doi.org/10.1016/j.ifacol.2023.10.1894>
- [11] Fu, S., Ge, Y., Hao, Y., Peng, J., & Tian, J. (2024). Energy supply chain efficiency in the digital era: Evidence from China's listed companies. *Energy economics*, 134, 107597. <https://doi.org/10.1016/j.eneco.2024.107597>

- [12] Kim, C. K., Lee, C., Kim, D., Cha, H., & Cheong, T. (2023). Enhancing supply chain efficiency: A two-stage model for evaluating multiple sourcing and extra procurement strategy optimization. *Sustainability*, 15(22). <https://doi.org/10.3390/su152216122>
- [13] Madhani, P. M. (2020). Enhancing supply chain efficiency and effectiveness with lean six sigma approach. *International journal of project management and productivity assessment (IJPMPA)*, 8(1), 40–65. <https://doi.org/10.4018/IJPMPA.2020010103>
- [14] Daneshvar, M., Hajiagha, S. H. R., Tupènaîtè, L., & Khoshkheslat, F. (2020). Effective factors of implementing efficient supply chain strategy on supply chain performance. *Technological and economic development of economy*, 26(4), 947–969. <https://doi.org/10.3846/tede.2020.12827>
- [15] Fischer, J. H., Pfeiffer, D., Hellingrath, B., Scavarda, L. F., & Martins, R. A. (2014). Robust parameter setting of supply chain flexibility measures using distributed evolutionary computing. *Procedia cirp*, 19, 75–80. <https://doi.org/10.1016/j.procir.2014.05.023>
- [16] Delporte, C., Fauconnier, H., Rajsbaum, S., & Raynal, M. (2022). Distributed computability: Relating k-immediate snapshot and x-set agreement. *Information and computation*, 285, 104815. <https://doi.org/10.1016/j.ic.2021.104815>
- [17] Liu, G., Xiao, Z., Tan, G., Li, K., & Chronopoulos, A. T. (2020). Game theory-based optimization of distributed idle computing resources in cloud environments. *Theoretical computer science*, 806, 468–488. <https://doi.org/10.1016/j.tcs.2019.08.019>
- [18] Lukoyanov, E., Kolesov, N., Gruzlikov, A., & Tolmacheva, M. (2022). Models of dynamic systems in diagnostic tasks. *Symmetry*, 14(7), 1433. <https://doi.org/10.3390/sym14071433>
- [19] Jaggard, A. D., Lutz, N., Schapira, M., & Wright, R. N. (2017). Dynamics at the boundary of game theory and distributed computing. *Association for computing machinery transactions on economics and computation*, 5(3), 1–20. <https://doi.org/10.1145/3107182>
- [20] Shoop, E., Matthews, S. J., Brown, R., & Adams, J. C. (2025). Hands-on parallel & distributed computing with Raspberry Pi devices and clusters. *Journal of parallel and distributed computing*, 196, 104996. <https://doi.org/10.1016/j.jpdc.2024.104996>
- [21] Dhamal, S., Ben-Ameur, W., Chahed, T., Altman, E., Sunny, A., & Poojary, S. (2018). *A stochastic game framework for analyzing computational investment strategies in distributed computing*. <https://doi.org/10.48550/arXiv.1809.03143>
- [22] Marden, J. R., & Shamma, J. S. (2018). Game-theoretic learning in distributed control. In *Handbook of dynamic game theory* (pp. 511–546). Cham: Springer international publishing. https://doi.org/10.1007/978-3-319-44374-4_9
- [23] Purohit, R., Chowdhary, K. R., & Purohit, S. D. (2023). *On the design and analysis of parallel and distributed algorithms*. <https://doi.org/10.48550/arXiv.2311.05857>
- [24] Sarkar, L. (2014). *Distributed systems: Basic algorithms*. University of Edinburgh.
- [25] Herlihy, M., Kozlov, D., & Rajsbaum, S. (2013). *Distributed computing through combinatorial topology*. Newnes.
- [26] Ali, M. F., & Khan, R. Z. (2015). Distributed computing: An overview. *International journal of advanced networking and applications*, 7(1), 2630. <https://www.researchgate.net/publication/280977301>
- [27] Bouneb, Z. E. A. (2022). A distributed algorithm for computing groups in IoT systems. *International journal of software science and computational intelligence (IJSSCI)*, 14(1), 1–21. <https://doi.org/10.4018/IJSSCI.300363>
- [28] Karishma, & Kumar, H. (2023). A new hybrid particle swarm optimization algorithm for optimal tasks scheduling in distributed computing system. *Intelligent systems with applications*, 18, 200219. <https://doi.org/10.1016/j.iswa.2023.200219>
- [29] Madiseti, V. K., & Panda, S. (2021). A dynamic leader election algorithm for decentralized networks. *Journal of transportation technologies*, 11(3), 404–411. <https://doi.org/10.4236/jtts.2021.113026>
- [30] Zhao, N., Hong, J., & Lau, K. H. (2023). Impact of supply chain digitalization on supply chain resilience and performance: A multi-mediation model. *International journal of production economics*, 259, 108817. <https://doi.org/10.1016/j.ijpe.2023.108817>
- [31] Liu, Y., & Jin, M. (2023). Does supply chain network centrality affect firm cost stickiness? *Finance research letters*, 58, 104459. <https://doi.org/10.1016/j.frl.2023.104459>
- [32] Saddikutti, V., Gudavalleti, P. K., & Singh, M. P. (2022). Lean and legacy supply chains for coordinated demand driven production to handle disruptions. *IFAC-papersonline*, 55(10), 2846–2851. <https://doi.org/10.1016/j.ifacol.2022.10.162>
- [33] Senthil, J., & Muthukannan, M. (2022). Development of lean construction supply chain risk management based on enhanced neural network. *Materials today: Proceedings*, 56, 1752–1757. <https://doi.org/10.1016/j.matpr.2021.10.456>
- [34] Xu, L., Mak, S., & Brintrup, A. (2021). *Will bots take over the supply chain? Revisiting agent-based supply chain automation*. <https://doi.org/10.48550/arXiv.2109.01703>
- [35] Chopra, S., & Sodhi, M. (2014). Reducing the risk of supply chain disruptions. *MIT sloan management review*, 55(3), 73–80. <https://sloanreview.mit.edu/article/reducing-the-risk-of-supply-chain-disruptions/>
- [36] Zhao, P., Yin, S., Han, X., & Li, Z. (2021). Research on lean supply chain network model based on node removal. *Physica A: Statistical mechanics and its applications*, 567, 125556. <https://doi.org/10.1016/j.physa.2020.125556>
- [37] Maqueira, J. M., Novais, L. R., & Bruque, S. (2021). Total eclipse on business performance and mass personalization: How supply chain flexibility eclipses lean production direct effect. *Supply chain management: an international journal*, 26(2), 256–278. <https://doi.org/10.1108/SCM-02-2020-0083>
- [38] Dwivedi, S. K., Amin, R., & Vollala, S. (2020). Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism. *Journal of information security and applications*, 54, 102554. <https://doi.org/10.1016/j.jisa.2020.102554>

- [39] Boskabadi, A., Mirmozaffari, M., Yazdani, R., & Farahani, A. (2022). Design of a distribution network in a multi-product, multi-period green supply chain system under demand uncertainty. *Sustainable operations and computers*, 3, 226–237. <https://doi.org/10.1016/j.susoc.2022.01.005>
- [40] Validi, S., Bhattacharya, A., & Byrne, P. J. (2014). A case analysis of a sustainable food supply chain distribution system—A multi-objective approach. *International journal of production economics*, 152, 71–87. <https://doi.org/10.1016/j.ijpe.2014.02.003>
- [41] Dhamal, S., Ben-Ameur, W., Chahed, T., Altman, E., Sunny, A., & Poojary, S. (2022). Strategic investments in distributed computing: A stochastic game perspective. *Journal of parallel and distributed computing*, 169, 317–333. <https://doi.org/10.1016/j.jpdc.2022.07.012>
- [42] Ikevuje, A. H., Anaba, D. C., & Iheanyichukwu, U. T. (2024). Optimizing supply chain operations using IoT devices and data analytics for improved efficiency. *Magna scientia advanced research and reviews*, 11(2), 70–79. <https://doi.org/10.30574/msarr.2024.11.2.0107>
- [43] Bellingeri, M., & Cassi, D. (2018). Robustness of weighted networks. *Physica A: Statistical mechanics and its applications*, 489, 47–55. <https://doi.org/10.1016/j.physa.2017.07.020>
- [44] Bracha, G. (1987). Asynchronous Byzantine agreement protocols. *Information and computation*, 75(2), 130–143. [https://doi.org/10.1016/0890-5401\(87\)90054-X](https://doi.org/10.1016/0890-5401(87)90054-X)
- [45] King, V., & Saia, J. (2013). Byzantine agreement in polynomial expected time. *Proceedings of the forty-fifth annual ACM symposium on theory of computing* (pp. 401–410). New York, NY, USA: Association for computing machinery. <https://doi.org/10.1145/2488608.2488658>
- [46] Liu, M., Song, T., Hu, J., Yang, J., & Gui, G. (2019). Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio Networks. *IEEE transactions on vehicular technology*, 68(1), 641–653. <https://doi.org/10.1109/TVT.2018.2883669>
- [47] Métivier, Y., Robson, J. M., Saheb-Djahromi, N., & Zemmari, A. (2011). An optimal bit complexity randomized distributed MIS algorithm. *Distributed computing*, 23(5), 331–340. <https://doi.org/10.1007/s00446-010-0121-5>
- [48] Schneider, J., & Wattenhofer, R. (2008). A log-star distributed maximal independent set algorithm for growth-bounded graphs. *Proceedings of the twenty-seventh acm symposium on principles of distributed computing* (pp. 35–44). New York, NY, USA: Association for computing machinery. <https://doi.org/10.1145/1400751.1400758>
- [49] Frischknecht, S., Keller, B., & Wattenhofer, R. (2013). Convergence in (social) influence networks. *Distributed computing* (pp. 433–446). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41527-2_30
- [50] Ayala, V., & Da Silva, A. (2002). Structural properties of the bounded control set of a linear system. <https://www.researchgate.net/publication/339015921>
- [51] Joseph, G. (2023). Output controllability of a linear dynamical system with sparse controls. *IEEE transactions on control of network systems*, 10(1), 147–156. <https://doi.org/10.1109/TCNS.2022.3188484>