Mathematical Researches

Print ISSN: 2588-2546
Online ISSN: 2588-2554

Homepage: https://jemr.khu.ac.ir/

An Interdiction Model Designed To Maintain the Optimal Spanning Tree Selected By the Follower

J. Tayyebi[1], M. **Niksirat**[2] ✉

1. Department of Industrial Engineering, Birjand University of Technology, Birjand, Iran. E-mail: javadtayyebi@birjandut.ac.ir
2. Corresponding Author, Department of Computer Sciences, Birjand University of Technology, Birjand, Iran. ✉ E-mail: niksirat@birjandut.ac.ir

Article Info

Article type:
Research Article

Article history:
Received: 27 August 2024
Received in revised form: 24 May 2025
Accepted: 9 July 2025
Published online: 4 October 2025

Keywords:
Minimum spanning tree,
Interdiction problem,
Fixed strategy,
Linear cost function.

ABSTRACT

Introduction

This paper examines a combinatorial optimization interdiction problem, called minimum spanning tree interdiction. From the point of view of game theory, this problem involves two players with different goals. The first player, called the follower, seeks to find a minimum spanning tree. On the other hand, the other player, who is called the leader, increases the cost of arcs, taking into account the budget and the bound constraints, which will make the value of the follower's objective function worse as much as possible, with the hope that the follower will stop doing more. In this article, a special case of the problem is considered in which the initial optimal tree remains optimal even with the change of weights. This assumption guarantees that the leader can be sure that the follower has no motivation to change his strategy, because a better strategy is not available.

Material and Methods

In order to efficiently solve this problem with linear costs, a new model is proposed, which includes a large number of constraints. For this reason, a solution approach based on the row (constraint) generation method is proposed for the problem. Then a numerical example and calculation results on random samples are presented to evaluate the performance of this method.

Results and discussion

The proposed model addresses this problem by utilizing a row generation approach to enhance solving efficiency. Computational results demonstrate its acceptable performance under various conditions, highlighting its potential for practical applications. The computational results indicate that the proposed algorithm computes the optimal solution to the problem within a finite number of iterations and a reasonable execution time.

Conclusion

The following conclusions are obtained from this research

- The initial optimal tree remains unchanged, ensuring the follower's strategy persists
- Row-generation algorithm developed for efficient linear-cost solution
- Achieves optimal solutions in limited iterations; tested with random instances
- Shows practical potential for various real-world scenarios

How to cite: Tayyebi, J., & Niksirat, M. (2025). An interdiction model designed to maintain the optimal spanning tree selected by the follower. *Mathematical Researches*, **11** (1), 99 – 120.

 Publisher: Kharazmi University

شاپا چاپی: ۲۵۴۶-۲۵۸۸
شاپا الکترونیکی: ۲۵۵۴-۲۵۸۸

پژوهش‌های ریاضی

Homepage: https://mmr.khu.ac.ir/

مسئله ممانعت از درخت پوشای کمینه با هدف ثابت نگه داشتن استراتژی پیرو

جواد طیبی[۱]، ملیحه نیک‌سیرت[۲] ✉

۱. گروه مهندسی صنایع، دانشکده مهندسی کامپیوتر و صنایع، دانشگاه صنعتی بیرجند، بیرجند، ایران. رایانامه: Javadtayyebi@birjandut.ac.ir

۲. نویسنده مسئول، گروه علوم کامپیوتر، دانشکده مهندسی کامپیوتر و صنایع، دانشگاه صنعتی بیرجند، بیرجند، ایران. رایانامه: niksirat@birjandut.ac.ir

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

تاریخ دریافت: ۱۴۰۳/۶/۶
تاریخ بازنگری: ۱۴۰۴/۳/۳
تاریخ پذیرش: ۱۴۰۴/۴/۱۸
تاریخ انتشار: ۱۴۰۴/۷/۱۲

واژه‌های کلیدی:
درخت پوشای کمینه،
مسئله ممانعت،
استراتژی ثابت،
تابع هزینه خطی.

چکیده

این مقاله یک مسئله ممانعت در بهینه‌سازی ترکیبیاتی، به نام ممانعت از درخت پوشای کمینه (MSTI) را بررسی می‌کند. از دیدگاه نظریه بازی‌ها، این مسئله شامل دو بازیکن با اهداف متفاوت است. بازیکن اول که پیرو نامیده می‌شود به دنبال یافتن یک درخت پوشای کمینه است. از طرف دیگر، بازیکن دیگر که رهبر نامیده می‌شود، با در نظر گرفتن محدودیت‌های بودجه و کران، هزینه کمان‌ها را افزایش می‌دهد که مقدار تابع هدف پیرو را تا حد ممکن بدتر نماید؛ با این امید که پیرو دست از فعالیت بیشتر بردارد. در این مقاله حالت خاصی از مسئله در نظر گرفته شده است که در آن درخت بهینه اولیه حتی با تغییر وزن‌ها بهینه باقی می ماند. این فرض تضمین می‌کند که رهبر می‌تواند مطمئن باشد که پیرو انگیزه‌ای برای تغییر استراتژی خود ندارد، زیرا استراتژی بهتری برای وی در دسترس نیست.

به منظور حل کارای این مسئله با هزینه‌های خطی یک مدل جدید پیشنهاد می‌گردد که شامل تعداد زیادی محدودیت است. به همین دلیل یک الگوریتم حل بر مبنای روش تولید سطر (محدودیت) برای مسئله پیشنهاد می‌شود. سپس یک مثال عددی و نتایج محاسباتی بر نمونه‌های تصادفی برای ارزیابی عملکرد این روش ارائه می‌گردد. نتایج محاسباتی نشان می‌دهد که الگوریتم ارایه شده در تعداد تکرار متناهی و در زمان اجرای منطقی جواب بهینه مسئله را محاسبه می‌کند.

استناد: طیبی، جواد؛ و نیک‌سیرت، ملیحه؛ (۱۴۰۴). مسئله ممانعت از درخت پوشای کمینه با هدف ثابت نگه داشتن استراتژی پیرو. *پژوهش‌های ریاضی*، **۱۱** (۱)، ۹۹ – ۱۲۰.

مقدمه

در مسائل بهینه‌سازی شبکه، معمولاً یک تصمیم‌گیرنده به عنوان بازیگر اصلی وجود دارد که هدفش بهینه‌سازی جریان در شبکه است. این تصمیم‌گیرنده که به عنوان «پیرو» شناخته می‌شود، تلاش می‌کند با بهینه‌سازی منابع و مسیرها، کارایی شبکه را به حداکثر برساند [3].

برای هر مسئله بهینه‌سازی شبکه، یک مسئله ممانعت می‌تواند تعریف شود که جنبه‌های پیچیده و جذابی از تصمیم‌گیری را به نمایش می‌گذارد. در مسائل ممانعت، دو تصمیم‌گیرنده با اهدافی متناقض در تقابل هستند. تصمیم‌گیرنده اول، که همان پیرو است، به دنبال بهینه‌سازی عاملی در شبکه است، در حالی که تصمیم‌گیرنده دوم، که به عنوان «رهبر» شناخته می‌شود، هدفش ایجاد اختلال در شبکه و ممانعت از پیرو است. این تضاد اهداف باعث می‌شود که مسائل ممانعت به‌طور خاص و متمایز از مسائل بهینه‌سازی شبکه بررسی شوند.

به دلیل وجود این اهداف متضاد، می‌توان مسائل ممانعت را به عنوان یک بازی مجموع-صفر تصور کرد [۱۶]. در این نوع بازی، سود یک بازیکن به طور دقیق با ضرر دیگری برابر است. اما یک ویژگی کلیدی که مسائل ممانعت را از بازی‌های ساده مجموع-صفر متمایز می‌کند، ساختار سلسله‌مراتبی انتخاب استراتژی‌ها است. به عبارت دیگر، در این بازی، تصمیم‌گیری به این صورت است که ابتدا رهبر استراتژی خود را انتخاب می‌کند و بعد از آن، پیرو با مشاهده انتخاب‌های رهبر، سعی می‌کند استراتژی بهینه خود را پیدا کند. این نوع تعاملات می‌تواند به عنوان یک مدل از بازی‌های استکلبرگ در نظریه بازی‌ها تفسیر شود [۱۷]. در بازی‌های استکلبرگ، یکی از تصمیم‌گیرندگان (رهبر) دارای اختیارات اولیه برای تعیین استراتژی است، در حالی که دیگری (پیرو) پس از مشاهده استراتژی رهبر، واکنش نشان می‌دهد و تلاش می‌کند بهترین انتخاب را در پاسخ به اقدامات رهبر انجام دهد. این الگوی تصمیم‌گیری غیرخطی و انتخاب‌های استراتژیک، به فهم بهتر پیچیدگی‌ها و چالش‌های مربوط به مسائل ممانعت و بهینه‌سازی شبکه کمک می‌کند.

در میان مسائل ممانعت در بهینه‌سازی شبکه، دسته مسائل ممانعت در بهینه‌سازی ترکیبیاتی نیز مورد توجه قرار گرفته است. این مسائل به طور گسترده در کاربردهای مختلفی مانند امنیت، مبارزه با قاچاق مواد مخدر، اهداف بشردوستانه و رسیدگی به موضوع قاچاق انسان مورد بررسی قرار می‌گیرند. این موضوع اهمیت خود را در زمینه‌های گوناگون نشان می‌دهد [۷، ۸].

در این مقاله مسئله MSTI مورد مطالعه قرار می‌گیرد. به طور کلی، دو دسته مسئله بهینه سازی ترکیبیاتی بر ساختار درختان پوشا بیشتر مورد توجه پژوهشگران است که عبارتند از:

۱) مسئله درخت پوشای کمینه[1] (MST): هدف این مسئله یافتن یک درخت پوشا در یک شبکه وزن‌دار $G = (V, A, w)$ است که مجموع وزن کمان‌های آن کمینه باشد. این مسئله در زمان $O(|A| + |V|\, log\, |V|)$ با الگوریتم‌های شناخته شده‌ی کروسکال و پریم قابل حل است [۳].

۲) مسئله درخت پوشای کمینه-بیشینه[2] (MMST): این مسئله به دنبال کمینه کردن بیشینه وزن کمان‌ها در درخت پوشا است. این مسئله در زمان $O(|A|)$ توسط یک الگوریتم بازگشتی حل می‌شود [۱۴].

مسئله MSTI برای هر دو حالت بالا قابل تعریف است [۲]. مطالعات اولیه در زمینه مسئله MSTI به دنبال یافتن یک کمان از درخت پوشای کمینه بود که حذف آن منجر به بیشترین افزایش وزن در درخت می‌شود که به عنوان حیاتی‌ترین کمان شناخته می‌شود. ثابت می‌شود که این مسئله در زمان چندجمله‌ای قابل حل است. گسترش طبیعی این مسئله یافتن k کمان از درخت پوشای کمینه است که حذف آن منجر به بیشترین افزایش در درخت پوشا می‌شود. این مسئله به عنوان k–حیاتی‌ترین کمان نامیده می‌شود و ثابت می‌شود که این مسئله به طور قوی NP-سخت است. راهکارهای زیادی برای حل کارای این مسئله از جمله الگوریتم های ترتیبی، تصادفی و موازی در طول زمان توسعه یافته‌اند [۴]. مسئله MST با کمترین لبه[3] یکی از نزیکترین مسائل به مسئله k–بیشترین کمان حیاتی است. ارتباط بین این دو مسئله در مرجع [۵] مطالعه شده است. برای حل این مسایل راهکارهای متفاوتی بر اساس الگوریتم‌های تقریبی ارائه شده است [۱۰، ۱۸]. علاوه بر روش‌های تقریبی، لیانگ و شن پیشگامان توسعه یک الگوریتم دقیق با زمان اجرای $O(n^{K+1})$ بودند که بر اساس شمارش کامل برای مسئله k–بیشترین کمان حیاتی بودند [۹]. سپس بزگان و همکاران یک الگوریتم شاخه و کران برای مسئله ارائه دادند [۴].

این مقاله یک نسخه ویژه از مسئله MSTI را با تاکید بر دو جنبه مختلف بررسی می‌کند. اول، هزینه‌های رهبر را با استفاده از توابع خطی مدل می‌کند. قابل توجه است که اکثر مطالعات روی این مسئله، تابع افزایش هزینه را به صورت یک تابع ثابت در نظر گرفته است و در حالتی که تابع افزایش هزینه، خطی باشد، فقط یک مطالعه انجام شده است [۶].

دوم، این مطالعه فرض می‌کند که درخت اولیه بهینه حتی در مواجهه با تغییرات وزن بهینه باقی می‌ماند. قابل توجه است که این فرضیه قبلاً برای مسئله MSTI در نظر گرفته نشده است و برای اولین بار در این مقاله مطالعه شده است. این فرض اعتماد را به رهبر القا می‌کند، زیرا تضمین می‌کند که با توجه به اینکه استراتژی بهتری وجود ندارد، پیرو انگیزه‌ای برای تغییر

[1] Minimum spanning tree
[2] Min-max spanning tree
[3] Minimum edge blocker spanning tree problem

استراتژی خود ندارد. در بازی استکلبرگ، رهبر به‌طور استراتژیک تلاش می‌کند تا با ایجاد تعهدی معتبر، برتری حرکت نخست خود را حفظ کرده و از تغییر استراتژی پیرو جلوگیری کند. این تعهد برای تضمین پیش‌بینی‌پذیری، ثبات و کنترل بلندمدت بر محیط رقابتی ضروری است. در ادامه مهمترین دلایل کاربرد مسئله با در نظر گرفتن فرض جدید ارائه شده است.

پیش‌بینی‌پذیری و بهینه‌سازی تصمیمات: با محدود کردن گزینه‌های استراتژیک پیرو، رهبر می‌تواند پاسخ پیرو را با دقت پیش‌بینی کند، که این امر امکان بهینه‌سازی تصمیمات بر اساس واکنش‌های مورد انتظار را فراهم می‌آورد. این کاهش عدم قطعیت استراتژیک باعث برنامه‌ریزی مؤثرتر و دستیابی به نتیجه تعادلی بهینه می‌شود.

ایجاد تعهد و بازدارندگی استراتژیک: ایجاد تعهد بر انتظارات پیرو تأثیر گذاشته و باعث هماهنگی با تعادل مطلوب رهبر می‌شود. هنگامی که حرکت رهبر تعهدی معتبر را نشان می‌دهد، پیرو از اتخاذ راهبردهای متقابل که ممکن است تصمیمات استراتژیک رهبر را تضعیف کند، منصرف می‌شود. این مکانیسم بازدارنده برای حفظ مزیت رقابتی و جلوگیری از رفتارهای تهاجمی در بازار ضروری است.

ثبات بازار و کنترل رقابت: محدود کردن انعطاف‌پذیری استراتژیک پیرو، خطرات مربوط به پویایی‌های مخرب بازار مانند جنگ قیمتی، ورود به بازار یا گسترش ظرفیت تولید را کاهش می‌دهد. در صنایعی که با نوسانات شدید و مقررات محدودکننده مواجه هستند، چنین ثبات تحمیلی به حفظ سودآوری بلندمدت و اطمینان از رفتار رقابتی قابل پیش‌بینی کمک می‌کند.

بهینه‌سازی تعادل و حفظ مزیت بلندمدت: انگیزه رهبر برای تضمین ثبات استراتژی پیرو ناشی از هدفی گسترده‌تر در کنترل پویایی‌های بازی و حفظ شرایط تعادلی مطلوب است. با شکل‌دهی به محیط استراتژیک و محدود کردن تغییرات، رهبر موقعیت برتر خود را تقویت کرده و حداکثر پرداخت ممکن را تضمین می‌کند، که به سلطه استراتژیک پایدار منجر می‌شود.

در واقع هدف اصلی این است که رهبر با اطمینان از پاسخ پیرو، کنترل بیشتری بر نتیجه نهایی داشته باشد و بتواند تصمیمات بهتری برای حداکثر کردن منافع خود بگیرد، که این همان مفهوم اصلی استراتژی‌های پیش‌بینی‌شده در بازی‌های استکلسبرگ است.

به منظور حل کارای این مسئله یک مدل جدید پیشنهاد شده است که شامل تعداد زیادی محدودیت می‌باشد. به همین دلیل یک روش حل بر مبنای رویکرد تولید سطر برای مسئله پیشنهاد شده است. سپس یک مثال عددی و نتایج محاسباتی برای ارزیابی عملکرد این رویکرد ارائه می‌شود.

ساختار مقاله به شرح زیر تنظیم شده است. در بخش بعد تعریف مسئله و مدل برنامه‌ریزی ریاضی پیشنهادی ارائه شده است. روش حل مسئله بر مبنای الگوریتم تولید سطر در بخش سوم مقاله تشریح شده است. در بخش چهارم یک مثال عددی برای تشریح عملکرد الگوریتم ارایه شده است و در بخش پنجم نتایج محاسباتی مقاله با هدف اثبات کارایی روش پیشنهادی

ارائه شده است. در بخش ششم جمع بندی و پیشنهاداتی برای تحقیقات آتی در این زمینه ارائه گردیده است.

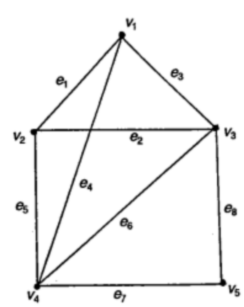
۱. مقدمات و تعریف مسئله

در این بخش ابتدا نمادها و مفاهیم مورد استفاده در ادامه مقاله تشریح می‌گردد. سپس مسئله MSTI تعریف و مدل پیشنهادی ارائه می‌شود. تعاریف مقدماتی از منبع [۱۲] گرفته شده است.

یک شبکه غیرجهت دار و همبند $G=(V,A,w)$ را در نظر بگیرید که در آن V مجموعه رئوس و A مجموعه کمان‌ها است و وزن‌های w نامنفی و قابل تغییر هستند. یک مسیر با طول k به صورت دنباله‌ای از رئوس $v_0,v_1,\dots,v_k\in V$ است به طوری که برای هر $(v_i,v_{i+1})\in A$، $0\le i\le k$. اگر همه رئوس v_i و v_j در این مسیر غیرتکراری باشند، یک مسیر ساده حاصل می‌شود. یک دور (یا دور ساده) که با C نمایش داده می‌شود، مسیر $v_0,v_1,\dots,v_k\in V$ با کمان اضافی $(v_k,v_0)\in A$ است. یک زیرگراف را همبند گوییم هرگاه بین هر دو راس آن حداقل یک مسیر وجود داشته باشد.

هر زیرگراف همبند $T=(V'\subseteq V,A'\subseteq A)$ از گراف G که هیچ دوری ندارد یک درخت نامیده می‌شود. اگر $V'=V$ آن‌گاه T یک درخت پوشای G نامیده می‌شود.

فرض کنید V_1 و V_2 دو زیرمجموعه ناتهی از V باشند به طوری که $V_1\cap V_2=\emptyset$. در این صورت مجموعه کمان‌هایی که یک انتهای آن در V_1 و انتهای دیگر آنها در V_2 باشد یک V_1V_2-برش نامیده می‌شود و با نماد $<V_1,V_2>$ نشان داده می‌شود. اگر T یک درخت پوشای G باشد، حذف هر کمان e از درخت T، درخت را به دو زیردرخت T_1 و T_2 تجزیه می‌کند. فرض کنید V_1 و V_2 به ترتیب مجموعه رئوس زیردرخت‌های T_1 و T_2 باشند. برش $<V_1,V_2>$ گراف G را به دو زیرگراف تجزیه می‌کند و مجموعه برش اساسی G مرتبط با کمان e از درخت T نامیده می‌شود. برای هر درخت پوشا از یک گراف همبند می‌توان $n-1$ برش اساسی مرتبط با کمان‌های درخت تولید نمود. برای درک بهتر مطلب گراف شکل ۱ را در نظر بگیرید:



شکل ۱: نمونه ای از یک گراف همبند [۱۱]

با در نظر گرفتن درخت پوشای $T = (\{v_1, v_2, v_3, v_4, v_5\}, \{e_1, e_2, e_6, e_8\})$ و با حذف کمان e_2 از درخت مجموعه $V_1 = \{v_1, v_2\}$ و $V_2 = \{v_3, v_4, v_5\}$ بدست می‌آید. در این صورت $< V_1, V_2 >= \{e_2, e_3, e_4, e_5\}$ مجموعه برش اساسی G مرتبط با کمان e از درخت T نامیده می‌شود.

در یک گراف وزن‌دار $G = (V, A, w)$ ، وزن درخت پوشای T برابر مجموع وزن کمان‌های آن به صورت زیر تعریف می‌شود:

$$f(T, w) = \sum_{(i,j)\in T} w_{ij} \qquad (۱)$$

اگر $\mathfrak{I}$ مجموعه تمام درخت‌های پوشای گراف $G = (V, A, w)$ باشد، هدف از حل مسئله درخت پوشای کمینه، پیدا کردن درخت پوشایی با کمترین وزن می‌باشد. بنابراین، این مسئله می‌تواند به صورت زیر بیان شود:

$$z(w) = min_{T\in\mathfrak{I}} f\,(T, w) \qquad (۲)$$

برای حل این مسئله، الگوریتم‌های زمان–چندجمله‌ای مشهوری مانند پریم و کروسکال وجود دارد. این الگوریتم‌ها بر پایه برقراری شرایط بهینگی زیر طراحی شده‌اند.

قضیه: T^* درخت پوشای کمینه از $G = (V, A, w)$ است، اگر

$$w_{ij} \le w_{kl}, \qquad \forall (i,j) \in T^*, (k,l) \in C_{ij}$$

که در آن C_{ij} برش اساسی مرتبط با کمان درختی (i,j) از T^* است.

در حالی که شرایط بهینگی برای مسئله MST ساده است و بر اساس آن روش‌های کارایی مانند الگوریتم‌های کروسکال و پریم برای حل این مسئله ارائه شده است، اکثر مدل‌های برنامه‌ریزی خطی ارائه شده برای این مسئله شامل محدودیت‌هایی است که با افزایش اندازه مسئله به طور نمایی رشد می‌کند [۳]. با این حال، چندین مدل برنامه‌ریزی خطی (مختلط) صفر و یک برای گسترش مسئله وجود دارد. آخرین مدل ارایه شده در این زمینه به صورت زیر است [۱]:

$$\min \sum_{(i,j)\in E} w_{ij}\, y_{ij} \qquad (۳)$$

$$\sum_{j:(i,j)\in E} y_{ij} = \begin{cases} 1 & i \neq s \\ 0 & i = s \end{cases} \qquad \forall i \in V \tag{۴}$$

$$v_i \geq v_j + y_{ij} + M\left(1 - y_{ij}\right) \qquad \forall (i,j) \in E \tag{۵}$$

$$v_s = 0\,, \qquad 1 \leq v_i \leq \mathrm{n} - 1 \qquad \forall i \in V - \{s\} \tag{۶}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in E \tag{۷}$$

در این فرمول‌بندی ابتدا گراف به یک گراف جهت‌دار تبدیل می‌گردد. برای این منظور هر کمال با دو یال رفت و برگشت جایگزین می‌شود و مجموعه E نشان دهنده‌ی مجموعه یال‌های گراف است. متغیر صفر و یک y_{ij} نشان دهنده‌ی این است که آیا کمان (i,j) به درخت پوشا متعلق است یا خیر. ایده اصلی پشت مدل پیشنهادی بر اساس قیاس بین رودخانه‌هایی است که چندین منبع و یک چاهک دارند و درخت‌هایی که چندین شاخه دارند و یک تنه. در مدل پیشنهادی، یک گره دلخواه $s \in V$ به عنوان گره چاهک علامت‌گذاری می‌شود. سایر گره‌ها یا نمایانگر منابع آب هستند یا نقاط تلاقی میانی. در هر دو حالت، آب تنها از یک گره در سطح بالاتر به سطح پایین‌تر جریان می‌یابد. متغیرهای تصمیم v_i سطح آب در گره i را نشان می‌دهد. محدودیت (۵) از تولید دور در درخت جلوگیری می‌کند. در واقع این محدودیت تضمین می‌کند که اگر $y_{ij} = 1$ سطح گره i به اندازه یک واحد از سطح گره j بیشتر است. همچنین محدودیت (۴) تضمین می‌کند مجموع کمان‌های ورودی انتخابی به هر راس (به جز راس s) باید برابر یک باشد.

۱,۱ تعریف مسئله

در این مقاله سناریویی در نظر گرفته می‌شود که در آن رهبر به دنبال بیشینه‌سازی هزینه درخت پوشای کمینه به وسیله افزایش وزن کمان‌ها است. فرض کنید w'_{ij} وزن‌های افزایش یافته باشد که توسط رهبر مشخص شده است. در این صورت فضای جواب‌های شدنی رهبر به شکل زیر تعریف می‌شود:

$$W = \{w' \in \mathbb{R}^{|A|}: w_{ij} \leq w'_{ij} \leq w^u_{ij} \quad \forall (i,j) \in A, \sum_{(i,j)\in A} r_{ij}(w'_{ij} - w_{ij}) \leq R\} \tag{۸}$$

که در آن r_{ij} هزینه‌ای است که رهبر برای یک واحد افزایش وزن کمان (i,j) متحمل می‌شود. همچنین R کل بودجه موجود رهبر است. این رابطه دو محدودیت را برای افزایش هزینه کمان‌ها توسط رهبر اعمال می‌کند. محدودیت اول، محدودیت کران است که میزان حداکثر افزایش هزینه کمان (i,j) را محدود به کران بالای w^u_{ij} می‌کند. محدودیت دوم محدودیت بودجه است که می‌بایست هزینه افزایش همه کمان‌ها محدود به مقدار بودجه R باشد. توجه به این نکته ضروری

است که $r_{ij}(.)$ یک تابع از اعداد حقیقی مثبت ($\mathbb{R}^+$) به اعداد حقیقی مثبت است که در مورد آن با جزئیات بیشتر بحث خواهیم کرد.

این مسئله، مسئله MSTI نامیده شده و در قالب یک مسئله ترکیبیاتی می‌توان آن را به صورت فشرده زیر نمایش داد [۱۷].

$$\max_{w' \in W} \quad \min_{T \in \mathfrak{J}} \quad f\,(T, w') \tag{۹}$$

قابل توجه است که این مسئله یک مسئله دوسطحی است که در آن در سطح اول رهبر بردار وزن $w' \in W$ را انتخاب می‌کند. سپس پیرو، وزن‌های جدید را مشاهده کرده و درخت پوشای کمینه را با وزن‌های جدید انتخاب می‌کند [۱۳].

مسئله MSTI را می‌توان در حالت‌های مختلف بررسی کرد. این مسئله را می‌توان به عنوان یک بازی سلسله مراتبی دو نفره (بازی استکلبرگ) طبقه‌بندی کرد. علاوه بر این، این بازی می‌تواند به عنوان یک بازی مجموع-صفر نیز در نظر گرفته شود زیرا تابع $f(T, w')$ توسط یک بازیکن کمینه و همین تابع توسط بازیکن دیگر بیشینه می‌گردد. این مسئله در دو حالت هزینه ثابت و هزینه خطی قابل بررسی است. در مواجهه با هزینه‌های ثابت، تابع $r_{ij}(.)$ به صورت زیر تعریف می‌شود:

$$r_{ij}(w'_{ij} - w_{ij}) = \begin{cases} c_{ij} & w'_{ij} \neq w_{ij} \\ 0 & w'_{ij} = w_{ij} \end{cases} \tag{۱۰}$$

و در حالت خطی تابع به صورت

$$r_{ij}\big(w'_{ij} - w_{ij}\big) = m_{ij}(w'_{ij} - w_{ij})$$

خواهد بود که در این روابط c_{ij} بیانگر هزینه ثابت تحمیل شده بر افزایش وزن کمان (i, j) و m_{ij} بیانگر میزان هزینه افزایش وزن این کمان به اندازه یک واحد است. در این مقاله تمرکز بر هزینه‌های خطی است زیرا همانطور که خواهید دید در رویکرد مورد استفاده در صورت خطی بودن هزینه‌ها، مسئله در هر تکرار نیز خطی بوده و در نتیجه می‌توان آن را با کمک روش‌های مشهوری چون سیمپلکس در زمان منطقی حل نمود.

مسئله MSTI با هزینه‌های ثابت به عنوان مسئله بیشترین کمان‌های حیاتی[۱] نیز شناخته می‌شود و در پژوهش‌های متعددی این مسئله مورد توجه قرار گرفته است. حالتی که هزینه‌ها به صورت خطی مدل سازی شوند، به ندرت در ادبیات موجود مسئله مطالعه شده است. تا جایی که نویسندگان اطلاع دارند، مسئله MSTI با هزینه‌های خطی فقط در [۶] بررسی

[1] Most vital arcs problem

شده است که در آن یک الگوریتم با زمان $O\left(n^3m^2\, log\left(\frac{n^2}{m}\right)\right)$ که در آن m و n به ترتیب تعداد رئوس و تعداد کمان‌های گراف هستند، برای مسئله پیشنهاد شده است.

این مقاله مسئله MSTI را با هزینه‌های خطی و با اعمال فرض مهم زیر بررسی می‌کند.

فرض ۱،۱. اگر T^* درخت پوشای کمینه بر حسب وزن‌های اولیه w باشد، آنگاه این درخت در جواب بهینه‌ی مسئله MSTI نیز بهینه باقی می‌ماند.

بر طبق این فرض، اگر پیرو بخواهد استراتژی خود را تغییر دهد، احتمالاً با یک افزایش غیرمنتظره و ناخواسته در هزینه‌ها یا تأثیرات منفی حاصل از این تغییر مواجه خواهد شد. بنابراین، پیرو به این نتیجه می‌رسد که تغییر استراتژی نه تنها مزیتی برای وی ندارد، بلکه مشکلات و هزینه‌های بیشتری را برایش به دنبال خواهد داشت.

از این رو، رهبر به یک راه‌حل بهینه دست می‌یابد: او می‌تواند اطمینان حاصل کند که پیرو در یک حاشیه ایمنی باقی می‌ماند و استراتژی خود را تغییر نمی‌دهند؛ بنابراین با این رویکرد رهبر می‌تواند از بروز هزینه‌های پنهان ناشی از تغییرات نامطلوب و ناخواسته در پاسخ‌دهی پیرو به استراتژی‌ها جلوگیری کند. این رویکرد نه تنها ثبات را حفظ می‌کند، بلکه رهبر را قادر می‌سازد تا بر روی اهداف و راهبردهای کلان خود تمرکز کند بدون اینکه نگران تغییرات ناگهانی و غیرمنتظره در رفتار پیرو باشد.

به طور خلاصه، رهبر با این تدبیر به یک وضعیت متعارض دست می‌یابد که در آن اگر چه سعی در ثابت نگه‌داشتن استراتژی پیرو ممکن است هزینه اضافی بر وی تحمیل کند، اما در عین حال ثبات و پیش‌بینی‌پذیری در واکنش پیرو را نیز تضمین کرده و از هزینه‌های ناشی از تغییر استراتژی جلوگیری می‌کند.

با اعمال این فرض و استفاده از فرمول‌بندی ارایه شده در (۳)-(۷) می‌توان مسئله را به شکل زیر فرمول‌بندی نمود:

$$\max \sum_{(i,j)\in T^*} w_{ij} + x_{ij} \tag{۱۱}$$

$$\sum_{(i,j)\in T^*} w_{ij} + x_{ij} \le z \tag{۱۲}$$

$$\sum_{(i,j)\in A} r_{ij}(x_{ij}) \le R \tag{۱۳}$$

$$0 \le x_{ij} \le w_{ij}^u - w_{ij} \qquad \forall (i,j) \in A \tag{۱۴}$$

$$z = \min\{ \sum_{(i,j)\in E} (w_{ij} + x_{ij})\, y_{ij} \tag{۱۵}$$

$$\sum_{j:(i,j)\in E} y_{ij} = \begin{cases} 1 & i \neq s \\ 0 & i = s \end{cases} \qquad \forall i \in V \tag{۱۶}$$

$$v_i \geq v_j + y_{ij} + M\big(1 - y_{ij}\big) \qquad \forall (i,j) \in E \tag{۱۷}$$

$$v_s = 0\,, \qquad 1 \leq v_i \leq \mathrm{n} - 1 \qquad \forall i \in V - \{s\} \tag{۱۸}$$

$$y_{ij} \in \{0,1\} \quad \forall (i,j) \in E\} \tag{۱۹}$$

که در آن $x_{ij} = w'_{ij} - w_{ij}$. متغیر صفر و یک y_{ij} نشان دهنده‌ی این است که آیا کمان (i,j) به درخت پوشا متعلق است یا خیر. مشابه با مسئله (۹) این فرمول‌بندی نیز دو سطحی بوده که در سطح رهبر (تابع هدف (۱۱)) هدف ماکزیمم سازی وزن درخت T^* با افزایش وزن هر یال از مقدار w_{ij} به مقدار $w_{ij} + x_{ij}$ با محدودیت‌های کران و بودجه است. محدودیت (۱۲) تضمین می‌کند که وزن درخت T^* بیشتر از وزن درخت پوشای کمینه حاصل از مسئله سطح دوم با وزن‌های یالی $w_{ij} + x_{ij}$ نخواهد بود. سطح دوم همان مسئله (۳)-(۷) است که وزن‌های یالی آن به $w_{ij} + x_{ij}$ تغییر کرده است. متاسفانه به دلیل دو سطحی بودن و همچنین وجود متغیرهای صفر و یک، نمی‌توان به کمک نرم‌افزارهای حل کننده مسایل بهینه‌سازی این فرمول‌بندی را حل نمود.

۲. مدل سازی مسئله و الگوریتم تولید سطر

در این بخش، یک مدل ریاضی برای مسئله MSTI با ثابت نگه داشتن استراتژی پیرو پیشنهاد می‌شود. با توجه به اینکه تعداد محدودیت‌های مدل پیشنهادی زیاد است یک الگوریتم تولید سطر برای حل مسئله با هزینه‌های خطی ارائه می‌گردد.

با توجه به فرض ۱,۱، می‌توان استراتژی پیرو را ثابت و برابر T^* در نظر گرفت و به دنبال بدتر کردن این جواب تا حد ممکن بود. در عین حال سایر درختان پوشا نیز نباید با افزایش وزن این درخت وزنی کمتر از T^* داشته باشند زیرا در این صورت نیز فرض ۱,۱ برقرار نخواهد بود. بنابراین با تحمیل فرض ۱,۱ بر مسئله MSTI، می‌توان انتخاب استراتژی توسط پیرو را حذف کرده و محدودیت‌هایی را اضافه کرد که T^* همواره کمینه باقی بماند. بر این اساس مدل زیر برای این مسئله حاصل می‌شود:

$$max\, f\,(T^*, w') \tag{۲۰}$$

$$f(T^*, w') \leq f(T, w') \qquad T \in \mathfrak{J} \tag{۲۱}$$

$$w' \in W \tag{۲۲}$$

فرم گسترش یافته این مدل به شرح زیر است:

$$\max \sum_{(i,j)\in T^*} x_{ij} \tag{۲۳}$$

$$\sum_{(i,j)\in T^*} r_{ij}(x_{ij}) \leq R \tag{۲۴}$$

$$\sum_{(i,j)\in T^*} x_{ij} - \sum_{(i,j)\in T} x_{ij} \leq \sum_{(i,j)\in T} w_{ij} - \sum_{(i,j)\in T^*} w_{ij} \quad T \in \mathfrak{J} \tag{۲۵}$$

$$0 \leq x_{ij} \leq w'_{ij} - w_{ij} \tag{۲۶}$$

با توجه به اینکه مدل جدید ارائه شده برای مسئله MSTI شامل تعداد زیادی محدودیت می‌باشد، این محدودیت‌ها را می‌توان با استفاده از الگوریتم تولید سطر به مسئله اضافه نمود. بنابراین برای حل مسئله، الگوریتم تولید سطر به شرح زیر پیشنهاد می‌گردد.

الگوریتم ۱: الگوریتم تولید سطر برای حل مسئله MSTI

گام ۱: فرم گسترش یافته مدل مسئله MSTI را بدون مجموعه محدودیت‌های دوم در نظر بگیرید.

گام ۲: مسئله نتیجه شده از گام اول را با روش سیمپلکس حل کنید تا جواب بهینه $\left(x_{ij}\right)^*$ بدست آید.

گام ۳: درخت پوشای کمینه T را با وزن‌های $w_{ij} + \left(x_{ij}\right)^*$ بدست آورید.

گام ۴: اگر وزن درخت‌های T^* و T نسبت به وزن‌های $w_{ij} + \left(x_{ij}\right)^*$ با هم برابر باشد متوقف شوید. در غیر این صورت محدودیت متناظر با درخت T را به مسئله اضافه کرده و به گام ۲ بروید.

مزیت عمده این رویکرد در آن است که مسئله اولیه یک مسئله برنامه‌ریزی خطی بوده و در هر تکرار نیز یک محدودیت خطی به مسئله افزوده می‌شود در نتیجه مسئله حاصل از هر تکرار یک مسئله برنامه‌ریزی خطی بوده و می‌توان برای حل آن از رویکردهای زمان چند جمله‌ای از جمله الگوریتم های نقطه درونی و همچنین روش مشهور سیمپلکس استفاده نمود.

رویکرد اتخاذ شده در الگوریتم ۱ بر پایه روش تولید سطر یا به‌عبارتی تولید محدودیت است، که شباهت زیادی با تکنیک تجزیه بندرز دارد. در این روش، پاسخ اولیه‌ای که ارائه می‌شود الزاماً یک پاسخ شدنی نیست، اما در عین حال خاصیت بهینگی را دارا می‌باشد. فرآیند حل مسئله به‌گونه‌ای است که در هر مرحله یک محدودیت جدید به مدل برنامه‌ریزی خطی افزوده می‌شود، که باعث حرکت تدریجی مدل به سمت پاسخ شدنی می‌گردد، بدون اینکه ویژگی بهینگی از بین برود. این روند باعث کاهش تدریجی مقدار تابع هدف در تکرارهای مختلف می‌شود، چراکه اضافه شدن هر محدودیت جدید موجب بهبود مدل و

کاهش مقدار تابع هدف می‌گردد. ازاین‌رو، با افزایش تعداد تکرارها در فرآیند محاسبات، کاهش وزن درخت T^* مشاهده می‌شود. به‌منظور تحلیل میزان تأثیر این استراتژی، یک درخت کمینه جدید بر اساس وزن‌های اصلاح‌شده $w_{ij} + \left(x_{ij}\right)^*$ محاسبه می‌شود. این درخت کمینه، که با نام T شناخته می‌شود، بیانگر وضعیت درخت کمینه پس از اعمال تغییرات وزنی است. اگر وزن این درخت به‌اندازه وزن T^*برسد، فرآیند حل مسئله به نقطه توقف می‌رسد؛ زیرا در این مرحله موفق شده‌ایم درخت T^*را با حفظ ویژگی بهینگی، تا حد امکان گسترش دهیم. این امر نشان‌دهنده کارآمدی روش تولید سطر در هدایت مدل به سمت یک پاسخ شدنی و بهینه است، در حالی که بهبودهای تدریجی در مقدار تابع هدف را نیز تضمین می‌کند.

۳. مثال عددی

در این بخش، به منظور نمایش عملکرد و صحت الگوریتم ۱، یک مثال عددی ارائه می‌شود. این مثال با فرض هزینه‌های خطی برای کمان‌ها شبیه‌سازی شده است. برای پیاده‌سازی الگوریتم و به ویژه رسم نمودار گرافیکی، از زبان برنامه‌نویسی پایتون و کتابخانه‌ی محبوب Matplotlib استفاده شده است.

شکل ۲ گراف متناظر با مثال عددی را به تصویر می‌کشد. این گراف شامل ۱۰ رأس با برچسب‌های ۰ تا ۹ و ۲۸ کمان است. در جدول ۱ مشخصات هر کمان ارائه شده است و همچنین پارامترهای هر کمان (i,j)شامل وزن اولیه (w_{ij})، حداکثر وزن مجاز (w_{ij}^{u}) و هزینه افزایش وزن به ازای هر واحد (m_{ij}) می‌باشد.

در شکل ۲، کمان‌ها به دو دسته قرمز و سیاه تقسیم‌بندی شده‌اند. کمان‌های قرمز رنگ نشان‌دهنده‌ی کمان‌های درختی هستند که ابتدا بر اساس وزن‌های اولیه، درخت پوشای کمینه را تشکیل می‌دهند.

جدول ۲ روند اجرای الگوریتم ۱ را برای این مثال به تفصیل شرح می‌دهد. الگوریتم پس از ۱۰ تکرار به جواب بهینه می‌رسد. به عبارت دیگر، تنها در تکرار آخر به مساله محدودیتی اضافه می‌شود. در این جدول، درخت‌های کمینه حاصل از هر تکرار که با T نمایش داده شده‌اند و همچنین وزن درخت بهینه‌ی فعلی (T^*) و وزن درخت حاصل از هر تکرار گزارش شده است. یادآوری می‌کنیم که برای تعیین درخت کمینه Tاز الگوریتم‌هایی نظیر کراسکال و پریم استفاده کرد که در اینجا از الگوریتم کراسکال استفاده شده است. همانطور که مشاهده می‌شود، وزن درخت T^* در تکرار اول ۶۳ و در آخرین تکرار به ۵۰ کاهش می‌یابد. دلیل این کاهش آن است که در تکرار اول، تمام منابع صرف افزایش وزن کمان‌های درخت اولیه شده است. اما در تکرارهای بعدی، با اضافه شدن درخت‌های جدید، این منابع به صورت مساوی بین تمام درخت‌ها توزیع می‌شود تا جایی که در نهایت در تکرار آخر، وزن دو درخت برابر شده و الگوریتم متوقف می‌شود. جدول ۳ میزان افزایش وزن هر کمان در هر تکرار را به صورت دقیق نمایش می‌دهد.

نتایج حاصل از اجرای الگوریتم بر روی مثال عددی نشان‌دهنده‌ی کارایی و دقت بالای آن در یافتن جواب بهینه مسئله است. با توجه به روند کاهش وزن درخت T^* در هر تکرار و همگرایی الگوریتم به سمت جواب بهینه، می‌توان به عملکرد

مناسب الگوریتم در حل مسائل مشابه اطمینان داشت.

جدول ۱. پارامترهای مربوط به مثال عددی

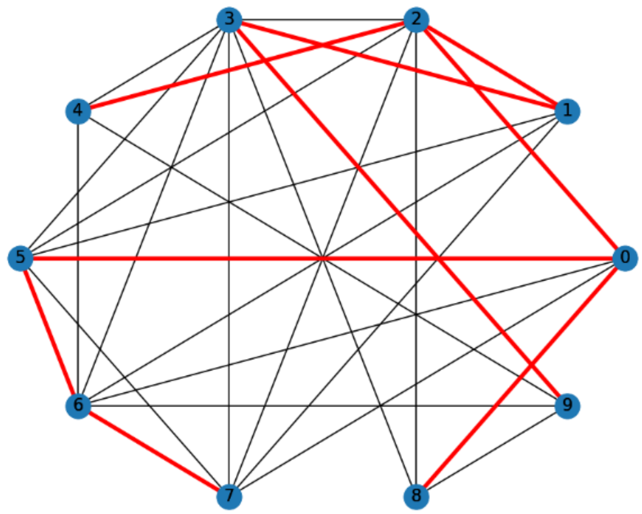
کمان	وزن اولیه	حداکثر وزن	هزینه واحد
(0,2)	2	11	2
(0,5)	4	11	10
(0,6)	2	5	6
(0,7)	9	12	1
(0,8)	5	7	1
(1,2)	2	5	10
(1,3)	5	7	8
(1,5)	6	15	5
(1,6)	2	5	9
(1,7)	1	8	4
(2,3)	5	12	1
(2,4)	1	9	10
(2,5)	6	8	3
(2,7)	7	14	1
(2,8)	5	11	9
(3,4)	4	9	7
(3,5)	9	16	9
(3,6)	4	8	6
(3,7)	9	11	2
(3,8)	10	11	9
(3,9)	2	7	2
(4,6)	2	10	5
(4,9)	6	13	1
(5,6)	1	3	3
(5,7)	4	5	10
(6,7)	1	4	7
(6,9)	7	9	3
(8,9)	9	11	10

جدول ۲. اجرای الگوریتم ۱ برای مثال عددی

شماره تکرار	وزن درخت T^*	وزن درخت T	درخت کمینه تشخیص داده شده T
1	63	32	(0, 5), (1, 2), (1, 6), (2, 8), (3, 6), (4, 6), (4, 9), (5, 6), (5, 7)
2	63	48	(0, 6), (1, 2), (1, 3), (1, 6), (2, 8), (2, 4), (3, 9), (5, 6), (6, 7)
3	56	45	(0, 5), (0, 8), (1, 7), (1, 2), (2, 3), (2, 4), (3, 9), (5, 6), (6, 7)
4	56	49	(0, 5), (0, 8), (1, 2), (1, 6), (1, 3), (2, 4), (3, 9), (5, 6), (6, 7)
5	56	48	(0, 6), (0, 8), (1, 7), (1, 2), (1, 3), (2, 4), (3, 9), (5, 6), (6, 7)
6	55	50	(0, 2), (0, 6), (0, 8), (1, 2), (1, 3), (2, 4), (3, 9), (5, 6), (6, 7)
7	55	47	(0, 6), (0, 8), (1, 2), (1, 6), (1, 3), (3, 4), (3, 9), (5, 6), (6, 7)
8	51	48	(0, 2), (0, 5), (0, 8), (1, 7), (1, 3), (3, 9), (3, 4), (5, 6), (6, 7)
9	51	49	(0, 2), (0, 6), (0, 8), (1, 7), (1, 3), (3, 9), (3, 4), (5, 6), (6, 7)
10	**50**	**50**	(0, 2), (0, 5), (0, 8), (1, 2), (1, 3), (2, 4), (3, 9), (5, 6), (6, 7)

جدول۳. مقادیر x_{ij} در تکرارهای مختلف الگوریتم ۱ برای مثال عددی

شماره تکرار	(0,2)	(0,5)	(0,6)	(0,7)	(0,8)	(1,2)	(1,3)	(1,5)	(1,6)	(1,7)	(2,3)	(2,4)	(2,5)	(2,7)	(2,8)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(3,9)	(4,6)	(4,9)	(5,6)	(5,7)	(6,7)	(6,9)	(8,9)
1	9	0	3	0	2	0	0	0	0	7	0	8	0	0	0	5	0	0	0	0	5	0	0	2	0	3	0	0
2	9	5	3	0	2	3	0	0	3	7	0	8	0	0	0	5	0	4	0	0	5	8	7	2	1	3	0	0
3	9	0	3	0	0	3	2	0	3	2	0	8	0	0	6	5	0	4	0	0	5	8	7	2	1	3	0	0
4	5	0	3	0	0	3	2	0	3	6	7	8	0	0	6	5	0	4	0	0	5	8	7	2	1	3	0	0
5	9	5	3	0	2	3	2	0	3	0	7	8	0	0	6	5	0	4	0	0	5	8	7	2	1	3	0	0
6	1	4	3	0	2	3	2	0	3	7	7	8	0	0	6	5	0	4	0	0	5	8	7	2	1	3	0	0
7	6	4	3	0	2	3	2	0	3	7	7	8	0	0	6	0	0	4	0	0	5	8	7	2	1	3	0	0
8	2	0	3	0	2	3	2	0	3	3	7	8	0	0	6	4	0	4	0	0	5	8	7	2	1	3	0	0
9	2	3	3	0	2	3	2	0	3	3	7	8	0	0	6	4	0	4	0	0	5	8	7	2	1	3	0	0
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



شکل ۲: گراف متناظر با مثال عددی

۴. نتایج محاسباتی بر نمونه‌های تصادفی

در این بخش، نتایج محاسباتی مختلفی را برای ارزیابی عملکرد الگوریتم پیشنهادی ارائه می‌دهیم. تمامی آزمایش‌ها بر روی یک لپ‌تاپ با سیستم‌عامل ویندوز ۱۱ و پردازنده Intel Core i7-8550U با فرکانس ۱٫۸۰ GHz و ۸ گیگابایت RAM انجام شده‌اند. برای پیاده‌سازی الگوریتم‌ها، از پایتون نسخه ۳٫۱۱٫۵ به همراه سه کتابخانه اصلی Matplotlib نسخه ۱٫۳٫۱، Pulp نسخه ۱٫۸٫۰ و NetworkX نسخه ۱٫۸٫۱ استفاده شده است. کتابخانه‌های NetworkX و Matplotlib به ترتیب برای مدیریت و ترسیم شبکه‌ها به کار می‌روند، در حالی که Pulp یکی از کتابخانه‌های معروف و رایگان برای حل مسائل بهینه‌سازی است. تمامی نتایج از میانگین مقادیر مربوط به ده شبکه با اندازه و چگالی یکسان استخراج شده است.

برای ایجاد گراف‌های تصادفی، از یک دسته خاص از گراف‌های تصادفی به نام گراف‌های دوجمله‌ای استفاده می‌کنیم. این گراف‌ها با تعیین دو پارامتر n و $p \in [0,1]$ تولید می‌شوند، که در آن n تعداد گره‌ها و p احتمال وجود هر یال (i,j) در گراف می‌باشد. بنابراین، هرچه مقدار p از صفر به یک افزایش یابد، گراف متراکم‌تر می‌شود تا جایی که به ازای مقدار یک، گراف به یک گراف کامل تبدیل می‌گردد.

برای تولید یک نمونه تصادفی از مسئله، ابتدا یک گراف بدون جهت دوجمله‌ای را ایجاد می‌کنیم. سپس، برای تولید

پارامترها با مقادیر صحیح، از توزیع احتمال یکنواخت به صورت زیر استفاده می کنیم:

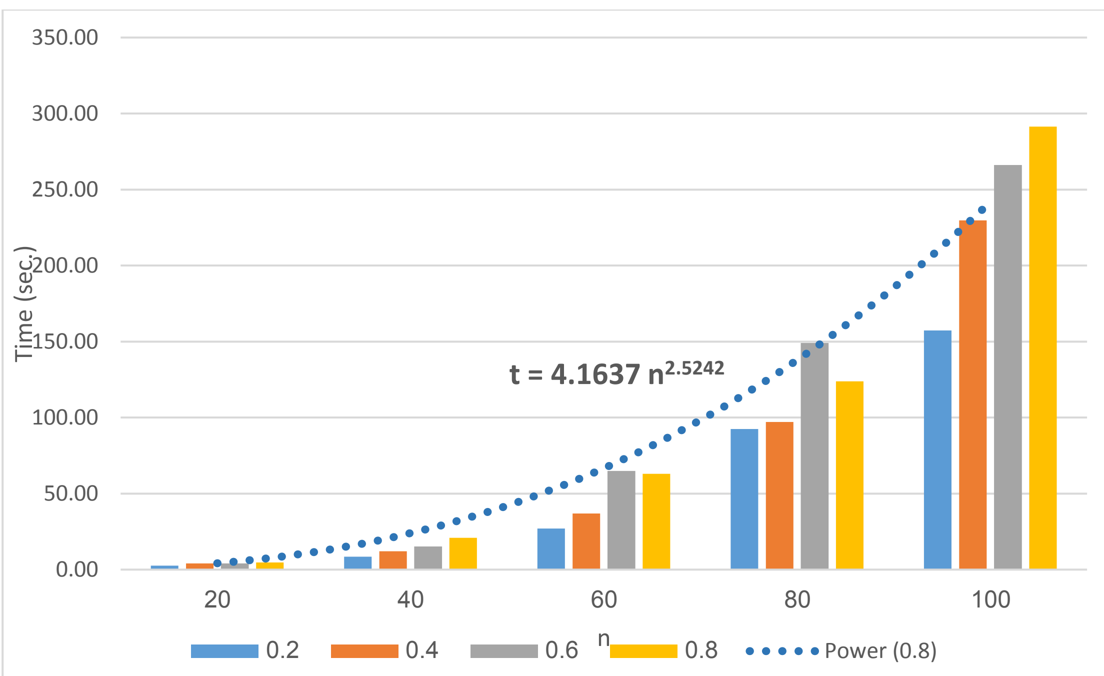
$$w_{ij} \sim U(1,n) \qquad \forall(i,j) \in A,$$

$$m_{ij} \sim U(1,n) \qquad \forall(i,j) \in A,$$

$$w_{ij}^{u} \sim w_{ij} + U(1,n) \qquad \forall(i,j) \in A,$$

$$R \sim U(1,n^3) \qquad \forall(i,j) \in A,$$

جدول ۴ نتایج عددی به دست آمده از اجراها را بر اساس تغییرات تعداد رئوس از ۲۰ تا ۱۰۰ و تغییرات چگالی از ۰,۲ تا ۰,۸ نمایش می‌دهد. همان‌طور که نتایج نشان می‌دهند، با افزایش تعداد رئوس و چگالی، تعداد تکرارها و زمان اجرای الگوریتم نیز افزایش می‌یابد. در این زمینه، **شکل ۳** نمودار تغییرات زمان اجرا را براساس تغییر در اندازه مسئله ارائه می‌کند. به‌ویژه، خط روند نمای چند جمله‌ای نشان‌دهنده وجود یک رابطه درجه ۲,۵ بین تعداد رئوس و زمان اجرا است.



شکل ۳: نمودار مقایسه زمان اجرا بر نمونه های تصادفی دوجمله‌ای

جدول ۴: نتایج عددی بر نمونه‌های تصادفی دوجمله‌ای

تعداد رئوس N	احتمال وجود یال p	تعداد یال‌ها M	تعداد متغیرها	تعداد محدودیت ها (تعداد تکرارها)	زمان (ثانیه)
20	0/2	42	39/6	15/2	2/61
	0/4	84	75/4	29/4	3/97
	0/6	126	117/2	33/4	4/05
	0/8	168	156/2	38/2	4/64
40	0/2	164	149/8	68/6	8/40
	0/4	328	300	93/6	12/13
	0/6	492	471/4	113/6	15/29
	0/8	656	621/8	144/6	20/99
60	0/2	366	364/6	177	27/08
	0/4	732	703	227/6	36/99
	0/6	1098	1053	343/2	64/86
	0/8	1464	1409/8	328/2	63/08
80	0/2	648	645/4	448/8	92/45
	0/4	1296	1262/2	447/4	97/19
	0/6	1944	1894	605/4	149/02
	0/8	2592	2524/8	518	123/91
100	0/2	1010	1000/6	616/8	157/37
	0/4	2020	2005/8	775/4	229/65
	0/6	3030	2972/6	825/4	266/11
	0/8	4040	3976/6	862	291/47

نمودار ارائه‌شده در شکل ۳ و نتایج محاسباتی در جدول ۴ نشان می‌دهد که افزایش تعداد رئوس در گراف تأثیر مستقیمی بر زمان اجرای الگوریتم دارد. این افزایش به گونه‌ای است که روند تغییرات زمان اجرا به طور تقریبی یک تابع درجه سوم نسبت به تعداد رئوس است. این امر نشان می‌دهد که با رشد اندازه گراف، زمان پردازش افزایش می‌یابد، به‌ویژه در مواردی که تعداد رئوس به مقدار زیادی برسد. علاوه بر تأثیر تعداد رئوس، چگالی یال‌های گراف نیز نقش قابل توجهی در تغییر زمان اجرا دارد. با افزایش چگالی یال‌ها، زمان اجرای الگوریتم افزایش می‌یابد، اما تأثیر این افزایش نسبت به تغییر تعداد رئوس کم‌تر محسوس است.

۵. نتیجه گیری و جمع‌بندی

این مقاله به بررسی مسئله MSTI در چارچوب نظریه بازی‌ها پرداخته و تعاملات بین دو بازیکن با اهداف متضاد را تحلیل می‌کند. در این زمینه، پیرو به دنبال بهینه‌سازی درخت پوشای خود بوده و رهبر با افزایش وزن کمان‌ها، تلاش می‌کند تا به گونه‌ای رفتار پیرو را دچار چالش کند. مسئله در صورتی که رهبر بخواهد درخت بهینه اولیه به وضعیت بهینه باقی بماند، مدل‌سازی شده است.

الگوریتم جدیدی که برای حل این مسئله با در نظر گرفتن هزینه‌های خطی پیشنهاد شده است، مبتنی بر یک رویکرد تولید سطر است که توانسته به بهبود کارایی حل این مسئله کمک کند. نتایج محاسباتی ارائه‌شده در این مطالعه نشان‌دهنده عملکرد مطلوب روش پیشنهادی در شرایط مختلف است و قابلیت‌های بالقوه آن را برای کاربردهای عملی تأیید می‌کند. بررسی‌های عددی بیانگر آن است که الگوریتم ارائه‌شده قادر است در یک تعداد تکرار محدود و طی یک زمان اجرای منطقی (تقریبا با مرتبه زمانی درجه سوم)، پاسخ بهینه مسئله را محاسبه نماید.

برای پژوهش‌های آتی، پیشنهاد می‌شود که تحقیقات بیشتری در زمینه گسترش این مدل با در نظر گرفتن شرایط مختلف نظامی، اقتصادی و اجتماعی انجام شود. همچنین، بررسی سناریوهای مختلف با تغییر پارامترهای موثر، مانند تعداد بازیکنان و ساختار هزینه‌ها، می‌تواند به فهم عمیق‌تری از این مسئله کمک کند. توسعه الگوریتم‌های بهینه‌سازی جدید برای حل مسائل مشابه با پیچیدگی‌های بیشتر نیز می‌تواند از دیگر زمینه‌های تحقیقاتی مفید باشد. نهایتاً، پیاده‌سازی مدل‌های پیشنهادی در دنیای واقعی و ارزیابی نتایج آنها می‌تواند منجر به دریافت بینش‌های کاربردی و عملی بیشتری در این حوزه شود.

قدردانی

نویسندگان مقاله مراتب قدردانی خود را از داوران محترم اعلام می‌دارند. بی‌شک نقطه نظرات ارزشمندشان در بهبود کیفیت مقاله نقش بسزایی داشته است.

References

1. T. F. Abdelmaguid, An efficient mixed integer linear programming model for the minimum spanning tree problem, Mathematics, **6** (2018), 183.

2. A. Abdolahzadeh, M. Aman, , J. Tayyebi, Bottleneck spanning tree interdiction problem with fixed and linear costs, Bull. Transilv. Univ. Bras. III: Math. Comput. Sci., **3** (2023) 185-198.

3. R. K. Ahuja, T. L. Magnanti and J. B. Orlin, Network flows: Theory, applications and algorithms. Englewood Cliffs, New Jersey, USA: Prentice-Hall, 1993.

4. C. Bazgan, S. Toubaline and D. Vanderpooten, Efficient determination of the k most vital edges for the minimum spanning tree problem, Comput. Oper. Res., **39** (2012), 2888-2898.

5. C. Bazgan, S. Toubaline, and D. Vanderpooten, Critical edges/nodes for the minimum spanning tree problem: Complexity and approximation, J. Comb. Optim., **26** (2013), 178-189.

6. G. N. Frederickson and R. Solis-Oba, Increasing the weight of minimum spanning trees, J. Algorithms, **33** (1999), 244-266.

7. N. Ghorbani-Renani, A. D. González and K. Barker, A decomposition approach for solving tri-level defender-attacker-defender problems, Comput. Ind. Eng., **153** (2021), 107085.

8. M. P. Johnson, A. Gutfraind, and K. Ahmadizadeh, Evader interdiction: Algorithms, complexity and collateral damage, Ann. Oper. Res., **222** (2014), 341-359.

9. W. Liang and X. Shen, Finding the k most vital edges in the minimum spanning tree problem, Parallel Comput., **23** (1997), 1889-1907.

10. A. Linhares and C. Swamy, Improved algorithms for MST and metric-TSP interdiction, arXiv preprint arXiv:1706.00034, 2017.

11. A. Liu, Graph Theory, in: S. M. A. R. T. Circle Minicourses, Springer Texts in Education, Springer, Cham, 2018.

12. F. R. P. Mary, S. Mohanaselvi, and S. Broumi, A solution approach to minimum spanning tree problem under Fermatean fuzzy environment, Bull. Electr. Eng. Inf., 12 (2023), 1738-1746.

13. L. Salazar-Zendeja, Models and algorithms for the minimum spanning tree interdiction problem (Doctoral dissertation), Centrale Lille Inst., 2022.

14. A. R. Sepasian and E. Monabbati, Upgrading min–max spanning tree problem under various cost functions, Theor. Comput. Sci., **704** (2017), 87-91.

15. J. C. Smith and Y. Song, A survey of network interdiction models and algorithms, Eur. J. Oper. Res., **283** (2020), 797-811.

16. J. Tayyebi, A. Mitra, and J. Sefair, The continuous maximum capacity path interdiction problem, Eur. J. Oper. Res., **305** (2023), 38-52.

17. N. Wei, J. L. Walteros, and F. M. Pajouh, Integer programming formulations for minimum spanning tree interdiction, INFORMS J. Comput., **33** (2021), 1461-1480.

18. R. Zenklusen, An O(1)-approximation for minimum spanning tree interdiction, In 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, (2015), 709-728.