Kharazmi University

Mathematical Research

Year 2025, Volume 11, Issue 3, pp. 31–41

Print ISSN: 2588-2546

Online ISSN: 2588-2554

DOI: xxxx

Approximating the edge general position number of graphs using meta-heuristic algorithms

Freydoon Rahbarnia[(1)] [1] and Zahra Hamedlabbafian[(2)]

[(1),(2)] Department of Applied Mathematics,Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

Received:1 June 2025 Accepted:15 October 2025 Published online:17 December 2025

Abstract: A subset of the edges of a graph is called an edge general position set if, for every pair of edges in the subset, none of the shortest paths between them contains any of the edges in the subset. The main objective of this study is to approximate the edge general position number of graphs using metaheuristic algorithms, including Genetic Algorithm (GA) and Simulated Annealing (SA).

Keywords: Graph, Edge General Position Set, Simulated Annealing Algorithm, Genetic Algorithm.

[1]Corresponding author

E-mail addresses: (Freydoon Rahbarnia) rahbarnia@ferdowsi.um.ac.ir, (Zahra Hamedlabbafian) hamedlabbafianzahra@gmail.com

پژوهش‌های ریاضی

سال 1404، دوره ۱۱، شماره ۳، ص ۳۱–۴۱

شاپا چاپی: ۲۵۸۸-۲۵۴۶

شاپا الکترونیکی: ۲۲۸۸-۲۵۵۴

DOI: xxxx

تقریب عدد موقعیت عمومی یالی گراف‌ها با استفاده از الگوریتم‌های فراابتکاری

فریدون رهبرنیا [(۱)] [۱] و زهرا حامدلبافیان [(۲)]

[(۱),(۲)] گروه ریاضی کاربردی، دانشکدهٔ علوم ریاضی، دانشگاه فردوسی مشهد، مشهد، ایران

تاریخ دریافت: ۱۴۰۴/۳/۱۱ تاریخ پذیرش: ۱۴۰۴/۷/۲۳ تاریخ انتشار: ۱۴۰۴/۹/۲۶

چکیده: یک زیرمجموعه از یال‌های گراف، مجموعه موقعیت عمومی یالی نامیده می‌شود اگر برای هر جفت یال در آن، تمامی کوتاه‌ترین مسیرهای بین آن‌ها شامل هیچ یک از یال‌های زیرمجموعه نباشد. هدف اصلی این پژوهش، تقریب عدد موقعیت عممومی یالی گراف‌ها با بهره‌گیری از الگوریتم‌های فراابتکاری، شامل الگوریتم ژنتیک و تبرید شبیه‌سازی شده، می‌باشد.

واژه‌هاي كليدي: گراف، مجموعه موقعیت عمومی یالی، الگوریتم تبرید شبیه‌سازی‌شده، الگوریتم ژنتیک

۱ مقدمه

گراف‌ها به‌عنوان ساختارهای ریاضی قدرتمند، نقش مهمی در مدل‌سازی سیستم‌های پیچیده در حوزه‌های مختلفی از جمله شیمی، زیست‌شناسی، شبکه‌های ارتباطی و علوم رایانه ایفا می‌کنند. یکی از مسائل مهم در نظریه گراف، مسئله موقعیت عمومی [۲] است که برای نخستین بار توسط مانوئل [۳] و کلاوژار [۴] معرفی شد و به بررسی زیرمجموعه‌ای از رئوس، X، می‌پردازد که هر کوتاه‌ترین مسیری در گراف شامل حداکثر ۲ رأس در X باشد [۵]. این مسئله در سال‌های اخیر گسترش یافته و نسخه‌های مختلفی از آن برای رئوس، یال‌ها و ساختارهای خاص گرافی مطرح شده‌اند. در نسخه یالی این مسئله، که با عنوان مسئله موقعیت عمومی یالی [۵] شناخته می‌شود، هدف یافتن زیرمجموعه‌ای

[۱] نویسنده مسئول مقاله

(Freydoon Rahbarnia) rahbarnia@ferdowsi.um.ac.ir, (Zahra Hamedlabbafian) rahbarnia@ferdowsi.um.ac.ir

[2] General Position Problem

[3] Manuel

[4] Klavžar

[5] Edge General Position Problem

از یال‌هاست که هر کوتاه‌ترین مسیر شامل حداکثر ۲ یال از این مجموعه باشد. این مفهوم به‌ویژه در مدل‌سازی شبکه‌های ارتباطی و ساختارهای مولکولی کاربرد دارد و به‌دلیل پیچیدگی ترکیبیاتی، این مسئله در دسته مسائل NP-سخت قرار می‌گیرد. یک گراف همبند [۱] است اگر بین هر دو رأس r و s یک دنباله متوالی از رئوس وجود داشته باشد که در آن هر جفت رأس متوالی با یک یال به هم متصل باشند. همچنین، یک گراف مسطح [۲] است اگر بتوان آن را روی صفحه یا سطح کره رسم کرد به‌گونه‌ای که هیچ دو یالی جز در رئوس مشترکشان یکدیگر را قطع نکنند. در هر گراف $G = (V, E)$، درجه رأس v، که با $deg(v)$ نمایش داده می‌شود، برابر است با تعداد یال‌هایی که از آن رأس می‌گذرد. اگر درجه تمام رأس‌های یک گراف دقیقاً برابر با ۳ باشد، آن گراف را مکعبی [۳] می‌نامند. در این پژوهش، دو الگوریتم فراابتکاری برای تقریب عدد موقعیت عمومی یالی در گراف‌ها ارائه می‌شود. در بخش دوم مقاله، تعریف دقیق مسئله و ویژگی‌های آن بیان می‌گردد. سپس در بخش بعد، دو الگوریتم تبرید شبیه‌سازی‌شده [۴] و الگوریتم ژنتیک [۵] به‌عنوان روش‌های پیشنهادی معرفی و تشریح می‌شوند. در ادامه، به‌منظور ارزیابی کارایی این الگوریتم‌ها، پیاده‌سازی بر روی مجموعه‌ای از گراف‌های معیار و نیز خانواده‌ای از گراف‌های فولرن انجام می‌گیرد. در پایان، نتایج عددی حاصل از اجرای الگوریتم‌ها مقایسه و تحلیل می‌شوند تا عملکرد نسبی هر روش در تقریب عدد موقعیت عمومی یالی مورد بررسی قرار گیرد.

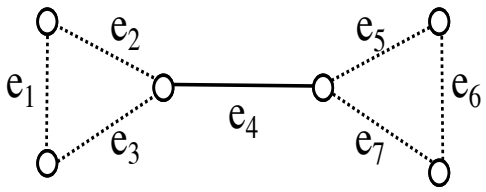
۲ تعریف مسئله موقعیت عمومی یالی

در این پژوهش، گراف $G = (V, E)$ یک گراف ساده و همبند است. دو رأس u و v مجاور [۶] هستند هرگاه یالی آن‌ها را به یکدیگر متصل کند. در این حالت، مجاورت آن‌ها با نماد $u \sim v$ نشان داده می‌شود. یک مسیر [۷] بین u و v به طول l، دنباله‌ای از رئوس متمایز گراف به‌شکل $u = u_0, u_1, \ldots, u_{l-1}, u_l = v$ است، به‌طوری که برای هر اندیس $0 \leq i < l$ شرط مجاورت $u_i \sim u_{i+1}$ برقرار باشد. فاصله [۸] میان دو رأس u و v برابر با طول کوتاه‌ترین مسیر بین آن دو رأس است، که با $d(u, v)$ نمایش داده می‌شود. مجموعه $X \subseteq E(G)$ یک مجموعه موقعیت عمومی یالی برای گراف G است، اگر هیچ کوتاه‌ترین مسیری در G بیش از دو یال از X را شامل نشود. اندازه بزرگ‌ترین مجموعه موقعیت عمومی یالی برای گراف G، عدد موقعیت عمومی یالی [۹] است که با $gp_e(G)$ نشان داده می‌شود [۶].

مثال ۱.۲. *همان‌طور که در شکل ۱ مشاهده می‌شود، $F = \{e_1, e_2, e_3, e_5, e_6, e_7\}$ یک مجموعه موقعیت عمومی یالی برای گراف G است. باتوجه به اینکه مجموعه موقعیت عمومی یالی بزرگتری وجود ندارد، بنابراین $gp_e(G) = 6$.*

مسئله موقعیت عمومی در گراف‌ها نخستین‌بار توسط مانوئل و کلاوژار در سال ۲۰۱۸ معرفی شد و از آن زمان تاکنون به‌طور گسترده‌ای مورد مطالعه قرار گرفته است. این مسئله در بسیاری از حوزه‌های علمی و فنی، به‌ویژه در مهندسی رباتیک و طراحی سامانه‌های هوشمند، کاربرد دارد. یکی از کاربردهای مهم این مسئله در ناوبری ربات‌های خودران است؛ در این مدل، ربات‌ها به‌صورت گره‌هایی در یک گراف در نظر گرفته می‌شوند و توانایی دیدن یکدیگر

[1] Connected
[2] Planar
[3] Cubic
[4] Simulated Annealing Algorithm
[5] Genetic Algorithm
[6] Adjacent
[7] Path
[8] Distance
[9] The Edge General Position Number



شکل ۱: مجموعه موقعیت عمومی یالی گراف G.

به مسیرهای مستقیم (بدون مانع) بین آن‌ها بستگی دارد. اگر سه ربات در یک خط قرار گیرند، ربات وسط می‌تواند دید بین دو ربات دیگر را مسدود کند. مسئله موقعیت عمومی کمک می‌کند تا بیشترین تعداد ربات‌هایی که می‌توانند به‌طور مستقیم و بدون انسداد حسگر، یکدیگر را ببینند، تعیین شود. در واقع رویکرد این مقاله ترکیبی از تحلیل‌های نظری، اثبات‌های ریاضی، و بررسی پیچیدگی محاسباتی مسئله است [۵]. پیش از آن، در سال ۲۰۱۶، چاندران [۱] و همکارش مفهوم مجموعه‌های ژئودزیک (کوتاه‌ترین مسیر) نامستقل را در گراف‌ها مطرح کردند که ارتباط نزدیکی با مسئله موقعیت عمومی دارد و به درک بهتر ساختار مسیرها در گراف‌ها کمک می‌کند [۱۲] . پاتکوش [۲] نیز در سال ۲۰۲۰ مطالعه‌ای دقیق روی مسئله موقعیت عمومی در گراف‌های کنزر ارائه داد که گامی مهم در تعمیم مسئله به کلاس‌های خاص گراف‌ها محسوب می‌شود [؟] . تیان [۳] و شو [۴] در سال ۲۰۲۱، با بهره‌گیری از رویکرد ضرب دکارتی، عدد موقعیت عمومی را در گراف‌هایی با قطر کوچک مورد بررسی قرار دادند و نتایج کاربردی در این زمینه ارائه کردند [۹] . کورژه [۵] و وسل [۶] در سال ۲۰۲۳ به تحلیل مجموعه‌های موقعیت عمومی در خانواده‌های خاصی از گراف‌های حاصل‌ضرب دکارتی پرداختند که اهمیت آن در بررسی گراف‌های مرکب و ترکیبی است [۴] . تیان و همکارانش در سال ۲۰۲۳ با تمرکز بر گراف‌های ماکسیمال بیرونی‌صفحه، به بررسی عدد موقعیت عمومی پرداختند که به شناخت بهتر ساختارهای صفحه‌ای کمک می‌کند [۱۰] . در سال ۲۰۲۵، حامد لبافیان و همکارانش به بررسی عدد موقعیت عمومی در گراف با رویکرد الگوریتمی و تقریب آن پرداخته‌اند [۲]. از سوی دیگر، مانوئل و همکارانش در سال ۲۰۲۲ مسئله موقعیت عمومی یالی را معرفی کردند که تعمیمی مهم از مسئله کلاسیک است و تمرکز آن بر روی یال‌ها به جای رئوس می‌باشد. این مقاله نه‌تنها یک مفهوم جدید را معرفی می‌کند، بلکه با رویکردی ترکیبی از نظریه گراف، هندسه گسسته و پیچیدگی محاسباتی، به نتایج قابل توجهی در کلاس‌های مختلف گراف‌ها دست یافته است [۶]. کلاوزار و همکارش در سال ۲۰۲۳ این مفهوم را در مکعب‌های فیبوناچی و لوکاس تعمیم داده و به مطالعه ساختارهای پیچیده‌تر پرداختند [۳] در ادامه، مانوئل و همکارانش در سال ۲۰۲۴ تعمیم‌های جدیدی از مسئله موقعیت عمومی یالی ارائه کردند که چارچوب نظری مسئله را توسعه داده است [۷] و تیان و همکاران نیز در همان سال به بررسی کران‌های این مسئله در برخی گراف‌ها پرداختند [۱۱]. با توجه به اینکه مسئله موقعیت عمومی یالی در دسته مسائل NPـسخت قرار دارد، رویکردهای الگوریتمی حل آن به دو دسته تقسیم می‌شوند: طراحی الگوریتم‌های دقیق برای گراف‌های کوچک یا خاص، و استفاده از الگوریتم‌های فراابتکاری برای تقریب پاسخ در گراف‌های بزرگ. در این مقاله، رویکرد دوم با بهره‌گیری از الگوریتم‌های تبرید شبیه‌سازی‌شده و ژنتیک برای گراف‌ها دنبال شده است.

[1] Chandran
[2] Patkós
[3] Tian
[4] Xu
[5] Korže
[6] Vesel

۳ دو الگوریتم فراابتکاری برای مسئله موقعیت عمومی یالی

در این بخش، دو رویکرد فراابتکاری شامل الگوریتم تبرید شبیه‌سازی‌شده و الگوریتم ژنتیک برای حل مسئله موقعیت عمومی یالی در گراف به کار گرفته شده است. هدف، یافتن بزرگ‌ترین زیرمجموعه‌ای از یال‌هاست که در هر یک از کوتاه‌ترین مسیرهای گراف G، حداکثر دو یال از این زیرمجموعه حضور داشته باشند. در الگوریتم SA، فرایند جستجو با یک جواب اولیه تصادفی آغاز می‌شود و با استفاده از پذیرش احتمالی جواب‌های ضعیف‌تر در دماهای بالا، امکان گریز از بهینه‌های محلی فراهم می‌شود. این روش با کاهش تدریجی دما، به سمت بهبود تدریجی بهترین جواب یافت‌شده حرکت می‌کند. در مقابل، الگوریتم GA با الهام از انتخاب طبیعی و استفاده از عملگرهای ترکیب [۱] و جهش [۲] ، جمعیتی از جواب‌ها را تکامل می‌دهد تا زمانی‌که شرط توقف الگوریتم برآورده شود. در این پژوهش، از سه نوع عملگر ترکیب و دو نوع عملگر جهش برای ایجاد تعادل میان اکتشاف فضای جستجو و بهره‌برداری از بهترین جواب‌های فعلی استفاده شده است. در ادامه، به بررسی الگوریتم‌های SA و GA می‌پردازیم.

۱.۳ الگوریتم تبرید شبیه‌سازی‌شده

با استفاده از الگوریتم تبرید شبیه‌سازی‌شده، برای هر گراف G به دنبال یافتن بزرگ‌ترین مجموعه موقعیت عمومی یالی مانند X هستیم. هر نمونه جواب در الگوریتم به صورت یک بردار باینری (لیست صفر و یک) به طول تعداد یال‌ها می‌باشد که هر عنصر آن متناظر با یک یال بوده و مقدار یک نشان‌دهنده حضور آن یال در مجموعه X است. الگوریتم با در نظر گرفتن لیستی به اندازه تعداد یال‌های گراف و اختصاص تصادفی صفر یا یک به هر یال، وجود یا عدم وجود آن در مجموعه موقعیت عمومی یال‌ها را مشخص کرده و در واقع یک پاسخ تصادفی تولید می‌کند. برازندگی آن بر اساس تعداد یال‌های انتخاب‌شده، با استفاده از الگوریتم ۲ محاسبه می‌گردد. در هر تکرار، همسایگان جواب فعلی با تغییر وضعیت یک یا دو یال به‌صورت تصادفی تولید می‌شوند. سپس برازندگی هر همسایه محاسبه و بهترین آن‌ها با عنوان N_{best} انتخاب می‌شود. اگر مقدار برازندگی N_{best} بیشتر از جواب فعلی باشد، جایگزین آن می‌شود. در غیر این صورت، ممکن است با احتمالی که بر اساس اختلاف برازندگی و دمای فعلی تعیین می‌شود، همچنان پذیرفته شود. در الگوریتم تبرید شبیه‌سازی‌شده، زمانی که همسایه‌ی انتخاب‌شده N_{best} دارای برازندگی کمتر از جواب فعلی S باشد، برای جلوگیری از گیر افتادن در بهینه‌های محلی، مکانیزمی احتمالی تعریف می‌شود که گاهی اجازه‌ی پذیرش این جواب‌های ضعیف‌تر را بدهد. اختلاف برازندگی به‌صورت زیر تعریف می‌شود:

$$\Delta F = fitness(N_{best}) - fitness(S)$$

در این حالت $\Delta F < ۰$. بنابراین یک عدد تصادفی u بین صفر و یک تولید می‌کنیم. اگر ، آنگاه جواب N_{best} جایگزین S می‌شود. در غیر این‌صورت S بدون تغییر باقی می‌ماند. واضح است که با کاهش دما، احتمال پذیرش این جواب‌ها کمتر می‌شود و الگوریتم به سمت همگرایی حرکت می‌کند. دمای سیستم به‌صورت تدریجی و طبق یک برنامه کاهش دما، کاهش می‌یابد تا احتمال پذیرش نمونه جواب‌های ضعیف‌تر با گذشت زمان کاهش یابد. این پذیرش احتمالی باعث می‌شود الگوریتم بتواند در مراحل اولیه از بهینه‌های محلی فرار کند. با کاهش دما، الگوریتم به‌تدریج رفتاری بهره‌بردارانه‌تر اتخاذ کرده و تمرکز خود را بر بهبود بهترین جواب‌های یافته‌شده معطوف می‌سازد. بنابراین پس از هر دوره سرمایش ($cooling\_time$ تکرار) ، دما طبق رابطه‌ی زیر کاهش می‌یابد:

$$T = T \times \rho, ۰ < \rho < ۱$$

[1] Crossover
[2] Mutation

هر چه مقدار نرخ کاهش دما (ρ) نزدیک به ۱ باشد، موجب کاهش آهسته و اکتشاف بیشتر در فضای جواب می‌شود، در حالی که مقادیر کوچک‌تر باعث سرد شدن سریع و ورود زودتر به فاز استثمار می‌گردد. پارامتر $cooling\_time$ نیز فاصله‌ی بین کاهش‌های دما را مشخص می‌کند. در پایان بهترین جواب مشاهده‌شده در طول اجرای الگوریتم به‌عنوان خروجی بازگردانده می‌شود.

الگوریتم ۱ الگوریتم تبرید شبیه‌سازی‌شده (SA) برای مسئله موقعیت عمومی یالی

ورودی: گراف بدون‌جهت $G = (V, E)$، حداکثر تعداد تکرار N، دمای اولیه T، نرخ کاهش دما ρ، دوره‌ی سرمایش $cooling\_time$

خروجی: یک مجموعه موقعیت عمومی یالی به‌صورت بردار دودویی از یال‌ها

۱: یک جواب اولیه تصادفی S به صورت بردار باینری به طول $|E|$ تولید کن.
۲: مقدار برازندگی S را محاسبه کن و آن‌را به عنوان بهترین جواب اولیه S_{best} ذخیره کن.
۳: **برای هر** $i = ۱, \ldots, N$ **انجام بده**
۴: همسایگان S را با تغییر وضعیت یک یا دو یال تصادفی تولید کن.
۵: برای هر همسایه مقدار برازندگی را محاسبه کن.
۶: همسایه‌ای با بیشترین مقدار برازندگی را به‌عنوان N_{best} انتخاب کن.
۷: **اگر** $fitness(N_{best}) \geq fitness(S)$ **آنگاه**
۸: N_{best} را جایگزینS کن.
۹: **وگرنه**
۱۰: یک عدد تصادفی u در بازه‌ی $(۰, ۱)$ تولید کن.
۱۱: $\Delta F = \text{fitness}(N_{\text{best}}) - \text{fitness}(S)$
۱۲: **اگر** $e^{\Delta F/T} \geq u$ **آنگاه**
۱۳: N_{best} را جایگزینS کن.
۱۴: **پایان اگر**
۱۵: **پایان اگر**
۱۶: **اگر** $fitness(S) > fitness(S_{best})$ **آنگاه**
۱۷: S را جایگزینS_{best} کن.
۱۸: **پایان اگر**
۱۹: **اگر** $i \bmod cooling\_time = ۰$ **آنگاه**
۲۰: $T \leftarrow T \times \rho$(در واقع، در هر مضرب از دوره‌ی سرمایش، دما به صورت $T \leftarrow T \times \rho$ به‌روزرسانی شود.)
۲۱: **پایان اگر**
۲۲: **پایان برای هر**
۲۳: **return** S_{best} را به‌عنوان بهترین جواب مسئله با مقدار برازندگی آن بازگردان

۲.۳ تابع برازندگی

این الگوریتم مقدار برازندگی ($fitness$) هر نمونه جواب را با ارزیابی ساختار و جریمه‌های موجود در گراف محاسبه می‌کند. ابتدا تعداد یال‌های انتخاب شده (فعال) مشخص می‌شود. سپس، برای هر جفت از رئوس، تمامی کوتاه‌ترین مسیرهای بین آن‌ها را بررسی می‌کند و درصورتی‌که مسیری شامل حداقل سه یال فعال (نامطلوب) باشد، جریمه‌ای اعمال می‌گردد. هر مسیر نامطلوب، با یک ضریب ثابت M ضرب شده و به‌عنوان جریمه در نظر گرفته می‌شود. در نهایت، مقدار تناسب از تفاضل تعداد یال‌های فعال(sum) و حاصل‌ضرب مجموع جریمه‌ها در ضریب M به دست می‌آید:

$$fitness = sum - (penalty \times M)$$

که در آن M ضریب جریمه بزرگ، برای جلوگیری از وجود مسیرهای نامطلوب است. با توجه به اینکه گراف مورد نظر ساده، بدون وزن و ثابت است، و الگوریتم در چندین مرحله نیازمند بررسی کوتاه‌ترین مسیر بین جفت رأس‌ها می‌باشد، استفاده از پیش‌پردازش جهت محاسبه و ذخیره‌سازی ماتریس فاصله‌ها (با استفاده از الگوریتم BFS برای هر رأس) منجر به کاهش چشم‌گیر در زمان اجرای گام‌های بعدی الگوریتم خواهد شد. پیچیدگی زمانی این پیش‌پردازش برابر با $O(n(n+m))$ است که در گراف‌های کم‌تراکم نیز قابل قبول بوده و با توجه به تکرار استفاده از داده‌ها، کاملاً مقرون‌به‌صرفه است (n تعداد رئوس و m تعداد یال‌های گراف است).

الگوریتم ۲ تابع برازندگی

ورودی: گراف بدون‌جهت $G = (V, E)$، نمونه جواب به‌صورت بردار باینری به طول $|E|$ که یال‌های فعال را نشان می‌دهد (۱ نشان‌دهنده یال فعال)، مقدار ضریب جریمه M
خروجی: مقدار برازندگی ($fitness$) مربوط به نمونه جواب

```
مقدار sum را برابر با تعداد یال‌های فعال قرار بده.
جریمه (penalty) را برابر با ۰ مقداردهی اولیه کن.
برای هر هر جفت رأس u و v در گراف انجام بده
    مجموعه P را به‌صورت تمام کوتاه‌ترین مسیرهای بین u و v درنظر بگیر.
    برای هر هر مسیر p در P انجام بده
        اگر تعداد یال‌های فعال در مسیر p برابر یا بیشتر از ۳ بود آنگاه
            مقدار penalty را یک واحد افزایش بده.
        پایان اگر
    پایان برای هر
پایان برای هر
مقدار تابع ارزیابی جواب را به‌صورت زیر محاسبه و بازگردان:
```

$$fitness = sum - (penalty \times M)$$

۳.۳ الگوریتم ژنتیک

الگوریتم‌های ژنتیک (GAs) از روش‌های بهینه‌سازی فراابتکاری هستند که از فرایند انتخاب طبیعی الهام گرفته‌اند. فرایند الگوریتم با یک جمعیت اولیه از نمونه جواب‌های تصادفی آغاز می‌شود. در این مرحله، جمعیتی به اندازه np به‌صورت تصادفی ایجاد می‌شود. هر فرد در جمعیت به شکل یک بردار باینری از صفر و یک نمایش داده می‌شود که طول آن برابر با تعداد یال‌های گراف است. مقدار «۱» نشان‌دهنده حضور یال در مجموعه موقعیت‌ها و مقدار «۰» نشان‌دهنده عدم حضور آن است. سپس در هر نسل، جواب‌های بهتر بر اساس مقدار برازندگی انتخاب می‌شوند. عملیات ترکیب و جهش روی جواب‌های منتخب اعمال می‌شود تا نسل جدیدی از جواب‌ها تولید شود. این روند به صورت تکراری ادامه پیدا می‌کند تا شرایط توقف برقرار شود، مانند رسیدن به تعداد نسل مشخص یا دستیابی به جواب قابل قبول. در این پژوهش، سه استراتژی ترکیب برای هر زوج والد به کار گرفته شده است تا فرزندان جدید حاصل گردد: روش OR هر یالی که در حداقل یکی از والدین فعال باشد، در فرزند نیز فعال تلقی می‌شود؛ روش AND تنها یال‌هایی را نگه می‌دارد که در هر دو والد فعال هستند؛ و روش $Half$ که نیمی از یال‌ها را از یک والد و نیمه دیگر را از والد دیگر به ارث می‌برد. به‌منظور حفظ تنوع در جمعیت، عمل جهش به‌طور هم‌زمان بر روی دو فرد اعمال می‌شود: بهترین فرد موجود در جمعیت و یک فرد دیگر که به‌صورت تصادفی انتخاب می‌گردد. در هر یک از این دو فرد، دو یال به‌طور تصادفی برگزیده شده و وضعیت آن‌ها تغییر می‌یابد؛ به این معنا که اگر مقدار یال برابر با «۱» (فعال) باشد، به «۰» (غیرفعال) تبدیل می‌شود و بالعکس. برای مثال، فرض کنید فردی شامل بردار

[۱, ۰, ۱, ۱, ۰] باشد. اگر یال‌های دوم و چهارم به‌صورت تصادفی انتخاب شوند، پس از تغییر وضعیت، بردار جدید به شکل [۱, ۱, ۱, ۰, ۰] در خواهد آمد. این روش تعادل بین اکتشاف و بهره‌برداری را در فرایند جستجو تضمین می‌کند. الگوریتم تا رسیدن به حداکثر تعداد تکرار که به‌عنوان شرط توقف عمل می‌کند، ادامه می‌یابد.

الگوریتم ۳ الگوریتم ژنتیک (GA) برای مسئله موقعیت عمومی یالی

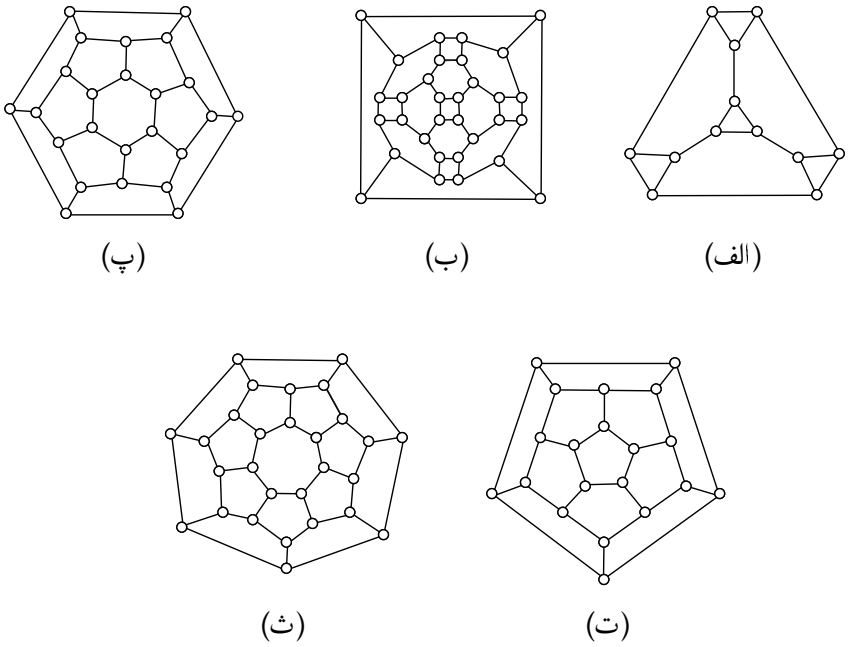
ورودی: یک گراف بدون‌جهت $G = (V, E)$، اندازه جمعیت np، حداکثر تعداد تکرار N
خروجی: یک مجموعه موقعیت عمومی یالی به شکل یک بردار دودویی از یال‌ها

۱: تولید جمعیت اولیه به اندازه np.
۲: **برای هر** هر نسل از ۱ تا N **انجام بده**
۳: محاسبه برازندگی ($fitness$) برای تمام افراد جمعیت با استفاده از الگوریتم ۲.
۴: اعمال سه استراتژی ترکیب بر روی دو فرد برتر به‌عنوان والدین و افزودن فرزند تولیدشده به جمعیت.
۵: اعمال دو نوع جهش و افزودن فرد تولیدشده به جمعیت.
۶: حذف افراد با کمترین مقدار برازندگی برای حفظ اندازه جمعیت np.
۷: **پایان برای هر**
۸: بازگرداندن بهترین فرد به همراه مقدار برازندگی آن.

۴ مقایسه الگوریتم تبرید شبیه‌سازی‌شده و الگوریتم ژنتیک

به منظور ارزیابی دقت و کارایی الگوریتم‌ها، آن‌ها را با استفاده از زبان برنامه‌نویسی پایتون نسخه ۳٫۱۱٫۵ پیاده‌سازی کرده و بر روی سیستم ۶۴ بیتی مجهز به پردازنده $Intel(R)Core(TM)i۵-۴۳۰۰M$ با فرکانس ۲٫۶۰ گیگاهرتز اجرا نمودیم. سپس الگوریتم‌ها را بر روی چند کلاس گراف معیار شناخته‌شده به‌کار بردیم. داده‌های گرافی و مقادیر عدد موقعیت عمومی یالی مربوطه از مقاله مرجع [۶] استخراج شده است که به صورت مستقیم به حل مسئله موقعیت عمومی یالی در دسته‌ای از گراف‌ها پرداخته است. در این موارد الگوریتم نتایج دقیقی ارائه داده است. با توجه به اهمیت گراف‌های فولرن به‌عنوان مدل‌های مناسب برای نمایش مولکول‌های کربنی (نظیر $C_{۶۰}$ که به نام باکمینسترفولرن نیز شناخته می‌شود) و بررسی ویژگی‌های شیمیایی آن‌ها، این گراف‌ها به‌عنوان بستر اصلی برای پیاده‌سازی و ارزیابی الگوریتم‌ها در این پژوهش انتخاب شده‌اند. فولرن‌ها گراف‌هایی همبند، مسطح و مکعبی هستند که تمام وجه‌های آن‌ها پنج‌ضلعی یا شش‌ضلعی‌اند و به‌دلیل ساختار متقارن و منظم خود، محیطی مناسب برای تحلیل‌های ترکیبی، شیمیایی و بهینه‌سازی فراهم می‌کنند. نتایج این آزمایش‌ها در جدول ۱ خلاصه شده‌اند. همچنین، شکل ۲ نمونه‌هایی از گراف‌های فولرن را که در پیاده‌سازی الگوریتم‌ها استفاده شده‌اند، نشان می‌دهد. بر اساس نتایج ارائه‌شده در جدول ۱ ، مقایسه بین الگوریتم ژنتیک (GA) و الگوریتم تبرید شبیه‌سازی‌شده (SA)، تفاوت‌های قابل‌توجهی در عملکرد آن‌ها را نشان می‌دهد. اگرچه هر دو الگوریتم قادر به یافتن جواب‌های با کیفیت بالا هستند، الگوریتم ژنتیک در اغلب موارد زمان اجرای بهتری از خود نشان می‌دهد. به‌طور خاص، الگوریتم ژنتیک همگرایی سریع‌تر و زمان محاسباتی کمتری دارد، که این موضوع را می‌توان به ماهیت جستجوی مبتنی بر جمعیت و کاوش هم‌زمان در فضای جواب نسبت داد. در مقابل، SA که مبتنی بر یک جواب منفرد است، معمولاً به تکرارهای بیشتری برای دستیابی به نتایج رقابتی نیاز دارد. بنابراین، از نظر کارایی زمانی، الگوریتم ژنتیک عملکرد بهتری نسبت به الگوریتم تبرید شبیه‌سازی‌شده دارد و برای کاربردهای حساس به زمان گزینه مناسب‌تری محسوب می‌شود. دستاورد اصلی این مقاله، ارائه رویکردی نوین برای حل مسئله موقعیت عمومی یالی در گراف‌ها با استفاده از الگوریتم‌های فراابتکاری است که نتایج تجربی آن کارایی و دقت بالای روش پیشنهادی را نشان می‌دهد. در جدول ۱ ، ستون G نوع گراف را نشان می‌دهد. در ستون $|V|$، تعداد رئوس گراف آورده شده

است. ستون gp_e عدد موقعیت عمومی یالی ۴ گراف اول را نشان می‌دهد که از منابع [۶] استخراج شده است. ستون‌های $SA.approximation$ و$GA.approximation$ به‌ترتیب مقادیر تقریبی به‌دست‌آمده توسط الگوریتم تبرید شبیه‌سازی‌شده (SA) و الگوریتم ژنتیک (GA) هستند. ستون‌های $SA.time$ و $GA.time$ زمان اجرای هر الگوریتم را (برحسب ثانیه) نمایش می‌دهند. مقادیر پارامترهای استفاده‌شده در اجرای الگوریتم‌ها در زیر هر زمان ذکر شده‌اند و بر اساس ساختار الگوریتم‌های ۱ و ۳ تعریف شده‌اند. نتایج ارائه‌شده در جدول ۱ نشان می‌دهد که الگوریتم‌های پیشنهادی در گراف‌هایی با اندازه کوچک عملکرد مناسبی دارند، اما با افزایش تعداد رأس‌ها، کارایی آن‌ها کاهش می‌یابد.



شکل ۲: الف) (۳و۶)ـ فولرن، ب) (۴و۶)ـ فولرن، پ) (۵و۶)ـ فولرن، ت) (۷و۶)ـ فولرن، ث) (۶و۷)ـ فولرن

$GA.time$	$SA.time$	gp_e $SA.approximation$ $GA.approximation$	$\|V\|$	G
۰٫۰۱ $np = ۱۰$ $N = ۲۰$	۱٫۷۳	۱۰ ۱۰ ۱۰	۱۱	$K_{۱,۱۰}$
۰٫۰۳ $np = ۱۰$ $N = ۲۰$	۳٫۵۲	۴۵ ۴۵ ۴۵	۴۵	$K_{۱۰}$
۰٫۰۱ $np = ۱۰$ $N = ۲۰$	۲٫۴۶	۴ ۴ ۴	۴	$Cycle\ graph(۱۰)$
۰٫۸۵ $np = ۱۰$ $N = ۲۰$	۴٫۷۶	۱۲ ۱۲ ۱۲	۱۲	$L(۱۰ \times ۲)$ $(Lattice\ graph)$
۰٫۲۵ $np = ۱۰$ $N = ۵۰$	۵٫۸۶	– ۱۲ ۱۲	۱۲	$(۳,۶) - fullerene$
۳۹٫۸۰ $np = ۱۰$ $N = ۲۰۰$	۲۰٫۴۳	– ۱۶ ۱۶	۳۲	$(۴,۶) - fullerene$ $((BN)_{۱۶}\ graph)$
۴٫۶۴ $np = ۱۰$ $N = ۲۰۰$	۴٫۳۶	– ۱۳ ۱۲	۲۶	$(۵,۶) - fullerene$ $(Generalized\ Peterson)$
۷٫۰۸ $np = ۱۰$ $N = ۵۰$	۳۶٫۰۷	– ۱۴ ۱۴	۲۸	$(۵,۷) - fullerene$
۱٫۹۹ $np = ۱۰$ $N = ۵۰$	۴٫۷۴	– ۱۲ ۱۲	۲۰	$(۵,۷) - fullerene$ $(Dodecahedral)$
۲۵۵٫۳۲ $np = ۲۰$ $N = ۱۰۰۰$	۵۹۵٫۵۴	– ۱۸ ۱۸	۶۰	$C_{۶۰}$ $(Truncated\ icosahedral)$
۴۴۲٫۱۲ $np = ۲۰$ $N = ۱۵۰۰$	۴۳۰٫۷۹	– ۲۰ ۱۶	۷۰	$C_{۷۰}$ $(۷۰ - fullerene)$

جدول ۱: مقایسه عملکرد الگوریتم تبرید شبیه‌سازی‌شده و الگوریتم ژنتیک بر روی گراف‌های خاص و فولرن

مراجع

[1] Ghorbani, M., 2013. Fullerene graphs with pentagons and heptagons. *Journal of Discrete Mathematics and Its Applications*, 3(1–2), pp.33–37.

[2] Hamed-Labbafian, Z., Sabeghi, N., Tavakoli, M. and Klavžar, S., 2025. Three algorithmic approaches to the general position problem. *Bulletin of the Australian Mathematical Society*. doi:10.1017/S0004972725100178

[3] Klavžar, S. and Tan, E., 2023. Edge general position sets in Fibonacci and Lucas cubes. *Bulletin of the Malaysian Mathematical Sciences Society*, 46(4), p.120.

[4] Korže, D. and Vesel, A., 2023. General position sets in two families of Cartesian product graphs. *Mediterranean Journal of Mathematics*, 20(4), p.203.

[5] Manuel, P. and Klavžar, S., 2018. A general position problem in graph theory. *Bulletin of the Australian Mathematical Society*, 98(2), pp.177–187.

[6] Manuel, P., Prabha, R. and Klavžar, S., 2022. The edge general position problem. *Bulletin of the Malaysian Mathematical Sciences Society*, 45(6), pp.2997–3009.

[7] Manuel, P., Prabha, R. and Klavžar, S., 2022. Generalization of edge general position problem. arXiv:2207.07357.

[8] Patkós, B., 2019. On the general position problem on Kneser graphs. arXiv:1903.08056.

[9] Tian, J. and Xu, K., 2021. The general position number of Cartesian products involving a factor with small diameter. *Applied Mathematics and Computation*, 403, p.126206.

[10] Tian, J., Xu, K. and Chao, D., 2023. On the general position numbers of maximal outerplane graphs. *Bulletin of the Malaysian Mathematical Sciences Society*, 46(6), p.198.

[11] Tian, J., Klavžar, S. and Tan, E., 2024. Extremal edge general position sets in some graphs. *Graphs and Combinatorics*, 40(2), p.40.

[12] Parthasarathy, G.J., 2016. The geodesic irredundant sets in graphs. *Mathematical Combinatorics (International Book Series)*, 4, p.135.