مقاله پژوهشی

رویکردی جدید برای شمارش یا بهینه‌سازی مثلث‌بندی مجموعه نقاط در صفحه مبتنی بر *MIS*

علی نوراله و زهرا رضایت

چکیده: مثلث‌بندی مجموعه نقاط S در صفحه، برابر با تعبیه مسطح یک گراف مسطح مستقیم‌الخط بیشین (با بیشترین یال) روی مجموعه نقاط است به طوری که مجموعه رئوس گراف دقیقاً همان مجموعه نقاط داده شده باشد. دو مسئله مهم در این زمینه مورد تحقیق است. الف) به چند طریق می‌توان مجموعه نقاط S را مثلث‌بندی کرد ب) کدام مثلث‌بندی بر اساس ویژگی خاصی بهینه است. مسئله اول یک مسئله باز است و به جز در شرایط خاص که دارای رابطه بسته می‌باشد تا به حال الگوریتمی با زمان چندجمله‌ای برای آن در حالت کلی ارائه نشده است. مسئله دوم نیز در حالتی که هدف پیداکردن مثلث‌بندی که مجموع طول یال‌های آن کمترین باشد یک مسئله NP-HARD است (MWT)، لذا تحقیقات در راستای ارائه الگوریتم‌های مکاشفه‌ای، فرامکاشفه‌ای یا تقریبی برای این دو حالت انجام شده است.

در این مقاله روشی ارائه شده که در آن با تولید گراف تقاطع حاصل از تمامی پاره‌خط‌های حاصل از تمامی زوج نقاط S تولید می‌شود و سپس الگوریتم‌هایی برای تولید همه مجموعه‌های مستقل بیشین (MIS) گراف تقاطع و همچنین روشی برای شمارش تعداد این مجموعه‌ها ارائه می‌شود. این رویکرد تولید گراف تقاطع و تبدیل مسئله مثلث‌بندی به مسئله مجموعه مستقل بیشین نگاهی جدید به مسئله مثلث‌بندی در هر دو حالت الف و ب محسوب می‌شود و از آنجا که ارائه الگوریتم برای مسئله الف یا ب به خاطر ذات هندسی‌بودن آن دشوار است لذا با رویکرد مطرح‌شده در این مقاله، تمامی الگوریتم‌هایی که تا به حال برای مسئله MIS مطرح شده است را می‌توان برای حل مسئله مثلث‌بندی در هر دو حالت الف یا ب به کار برد. تکنیک تبدیل مسئله مثلث‌بندی به مسئله MIS رویکردی است که تا به حال روشی مبتنی بر آن برای حل مسایل شمارش تعداد طرق مثلث‌بندی یا مثلث‌بندی با کمترین وزن گزارش نشده است. علاوه بر این یک روش تخمینی مکاشفه‌ای برای تعیین متوسط تعداد حالات مثلث‌بندی ارائه خواهد شد که نتایج پیاده‌سازی نشان می‌دهد روی نمونه‌هایی از ورودی نزدیک به مقدار دقیق هستند.

***کلیدواژه:* مثلث‌بندی، مثلث‌بندی با کمترین وزن، مسئله شمارش، مجموعه مستقل بیشین، گراف تقاطع.**

۱- مقدمه

مثلث‌بندی مجموعه نقاط S در صفحه (دوبعدی)، برابر است با گراف مستقیم‌الخط[1] بیشین[2] با مجموعه رئوس S، به طوری که گراف حاصل مسطح در صفحه تعبیه شده باشد. بیشین‌بودن این گراف به این معنی است که اضافه‌کردن هر یال مستقیم‌الخط دیگری به این گراف موجب تقاطع شده و گراف حاصل مسطح مستقیم‌الخط نخواهد بود. به بیان دیگر مثلث‌بندی مجموعه رئوس S در فضای دوبعدی، بیشترین تعداد پاره‌خط‌هایی (یال‌هایی) است که بین رئوس S رسم می‌شوند به طوری که یال‌ها یکدیگر را قطع نکنند. تعداد یال‌ها و مثلث‌های یک مثلث‌بندی روی رئوس داده‌شده، به تعداد رئوسی که بر روی پوسته محدب مجموعه رئوس قرار دارند وابسته است. اگر n تعداد عناصر مجموعه نقاط داده‌شده S در صفحه و h تعداد نقاط مرزی پوسته محدب[3] مجموعه S باشد، تعداد مثلث‌های هر مثلث‌بندی دلخواه روی S مساوی و برابر با $۲n-h-۲$ است و تعداد یال‌های هر مثلث‌بندی برابر با مقدار $۳n-h-۳$ می‌باشد.

اگر T یک نمونه مثلث‌بندی دلخواه روی مجموعه نقاط داده‌شده S و m تعداد مثلث‌های مثلث‌بندی T باشد، تعداد وجه‌های مثلث‌بندی که با n_f نشان داده می‌شود برابر با $m+۱$ است که یکی از این وجوه وجه نامتناهی بیرونی و مابقی، وجه‌های متناهی مثلثی می‌باشند. هر مثلث شامل سه یال و وجه نامتناهی شامل h یال است. از طرفی هر یال در دو وجه مشترک است، پس تعداد یال‌های مثلث‌بندی که با n_e نمایش داده می‌شود برابر با $(۳m+h)/۲$ است. روابط مطرح‌شده از روی رابطه اصلی اویلر[4] در گراف‌های مسطح که به صورت $n-n_e+n_f=۲$ است به دست می‌آیند.

مسایل مثلث‌بندی مجموعه نقاط و روش‌های مثلث‌بندی و پیداکردن یک مثلث‌بندی با ویژگی‌های خاص، از گذشته تا کنون مورد بررسی محققان زیادی قرار گرفته است. به عنوان نمونه فرض کنید n تعداد عناصر مجموعه نقاط S داده‌شده در صفحه باشد. روش‌های مختلفی وجود دارد که می‌توان این مجموعه نقاط را مثلث‌بندی کرد. از انواع مثلث‌بندی‌ها می‌توان به مثلث‌بندی دلانی[5] که یکی از مشهورترین مثلث‌بندی‌ها است اشاره کرد که در آن هر دایره گذرنده از سه رأس هر مثلث شامل هیچ رأس دیگری نمی‌باشد (دایره خالی است). این مثلث‌بندی از دوگان[6] دیاگرام وورونی[7] در زمان $O(n\log n)$ قابل تولید است. یافتن بهترین مثلث‌بندی تحت شرایط خاص همیشه مورد توجه بوده است. مثلث‌بندی دلانی از جهات زیادی یک مثلث‌بندی بهینه است و

این مقاله در تاریخ ۲۷ شهریور ماه ۱۳۹۸ دریافت و در تاریخ ۱۷ اردیبهشت ماه ۱۳۹۹ بازنگری شد.

علی نوراله (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران، (email: nourollah@sru.ac.ir).

زهرا رضایت، دانشکده مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران، (email: zahrarezayat72@gmail.com).

1. Straight-Line
2. Maximal
3. Convex Hull
4. Euler
5. Delaunay Triangulation
6. Dual
7. Voronoi Diagram

توابع هدف بسیاری را به صورت همزمان بهینه می‌کند که یکی از مهم‌ترین آنها این است که کوچک‌ترین زاویه در کل مثلث‌های این مثلث‌بندی، بیشینه است. به عبارت دیگر مثلث‌های مثلث‌بندی دلانی از مثلث‌های سایر مثلث‌بندی‌ها پهن‌تر و چاق‌تر است [۱].

نوع دیگری از مثلث‌بندی، مثلث‌بندی حریصانه[1] است که بر اساس حداقل‌سازی طول یال‌های مثلث‌های ایجادشده، عمل می‌کند. این بدان معنی است که در انتخاب مثلث‌ها سعی بر آن شده تا مجموع فواصل، حداقل مقدار ممکن را داشته باشند. در ابتدا فاصله بین کلیه رئوس محاسبه شده و کمترین فواصل مشخص می‌گردند و به ترتیب رسم می‌شوند. پیچیدگی زمانی محاسبات و فضای ذخیره‌سازی در این الگوریتم $O(n^2)$ است. این روش نیاز به محاسبه آرایه بزرگی از فواصل دارد که باعث زمان‌گیربودن این روش می‌گردد [۲].

از آن زمان که محققان به کاربردهای زیرساختی و اساسی مثلث‌بندی پی بردند به دنبال آن بودند که بتوانند با بهینه‌سازی مثلث‌بندی، کیفیت کار خود را افزایش دهند.

مسئله مثلث‌بندی بهینه مجموعه رئوس در سال ۱۹۷۰ توسط دوپ و گات‌اسکاک[2] مطرح شد [۳]، که پیشنهاد آن برای ساخت مدل‌های شبکه‌ای نامنظم مثلثی از زمین با استفاده از الگوریتم حریصانه برای تقریب آن بود. شاموس و هوی[3] در سال (۱۹۷۵) استفاده از مثلث‌بندی دلانی را برای مثلث‌بندی بهینه ارائه دادند [۴] اما این مسئله توسط لوید[4] (۱۹۷۷) رد شد [۵]. در واقع کیرک‌پاتریک[5] نشان داد که وزن دو مثلث‌بندی می‌تواند با یک فاکتور خطی از هم متفاوت باشند [۶].

در سال ۱۹۷۹ توسط گری و جانسون[6] نشان داده شد که مسئله مثلث‌بندی بهینه راه حل قطعی با زمان چندجمله‌ای ندارد و آن را در لیستی از مسایل باز در کتاب خود که درباره NP-Completness بود قرار دادند [۷]. در نهایت، مالزر و روت[7] در سال ۲۰۰۸ نشان دادند که مسئله مثلث‌بندی بهینه جزو مسایل NP-Hard است [۸]. رمی و استیجر[8] در سال ۲۰۰۹ نشان دادند که می‌توان از الگوریتم‌های تقریبی برای حل این مسئله استفاده کرد [۹].

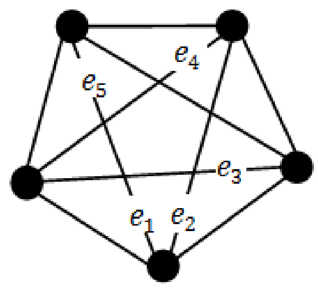
مثلث‌بندی با کمترین وزن، یکی از مسایل مهم در حوزه هندسه محاسباتی می‌باشد که تحقیقات زیادی روی آن صورت گرفته است. با توجه به عدم وجود الگوریتم با زمان چندجمله‌ای برای حل مسئله به صورت بهینه، با روش‌های شناخته‌شده، محققان سعی کردند این مسئله را در حالت‌های خاصی حل کنند. از این رو می‌توان الگوریتم‌های ارائه‌شده برای محاسبه مثلث‌بندی بهینه را به چند دسته تقسیم نمود. دسته‌ای از الگوریتم‌ها مسئله را برای حالت‌های خاصی (و نه در حالت کلی) از مجموعه نقاط محاسبه می‌کنند. بعضی دیگر با استفاده از الگوریتم‌های تقریبی[9] وزن مثلث‌بندی بهینه را تقریب می‌زنند. در [۱۰] از روش کلونی مورچه‌ها که یک الگوریتم فرامکاشفه‌ای[10] محسوب می‌شود برای مسئله مثلث‌بندی با کمترین وزن استفاده شده است. تعدادی نیز از تکنیک‌های مکاشفه‌ای[11] برای حل این مسئله استفاده می‌کنند. روش پیشنهادی در این مقاله نشان می‌دهد که با استفاده از کاهش‌پذیری مسئله مثلث‌بندی بهینه به مسئله مجموعه رئوس مستقل بیشین با طول بیشینه[12] و با وزن کمینه (که به هر رأس از گراف تقاطع وزنی معادل طول هر یال نسبت داده می‌شود) بر روی گراف تقاطع[13] حاصل از گراف کامل با n_e رأس که از روی مجموعه نقاط اولیه در صفحه قابل ساخت است، می‌توان رویکرد جدیدی برای بهینه‌سازی مثلث‌بندی مجموعه نقاط در صفحه ایجاد کرد.

مجموعه رئوس مستقل برای یک گراف، مجموعه‌ای از رئوس گراف است که هیچ یالی میان هیچ جفتی از این رئوس وجود نداشته باشد. این مجموعه مستقل یا به صورت بیشینه[14] و یا به صورت بیشین مورد نظر قرار می‌گیرد. در تعریف اول باید به گونه‌ای رئوس مستقل در گراف یافت شوند که در آن گراف، مجموعه مستقل با تعداد رئوس بیشتر را نتوان یافت. در تعریف دوم مجموعه رئوسی مستقل از گراف مورد نظر مطرح است که نتوان رأس دیگری به آن به گونه‌ای اضافه کرد که مجموعه حاصل مستقل باقی بماند. هر مجموعه مستقل بیشینه، مجموعه مستقل بیشین نیز هست، ولی مجموعه مستقل بیشین لزوماً بیشینه نیست. برای گراف G، مجموعه مستقل بیشینه بیشترین رأس‌های غیر همسایه از این گراف را در خود دارد و اندازه این مجموعه با $\alpha(G)$ نشان داده می‌شود. یافتن مجموعه مستقل بیشینه در گراف دلخواه مسئله‌ای NP-Hard است و از این رو تا به حال الگوریتمی با زمان چندجمله‌ای برای آن ارائه نشده است.

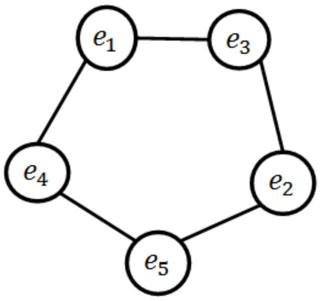
مجموعه مستقل در نظریه گراف و علوم کامپیوتر کاربردهای بسیاری دارد، مثلاً لاولر[15] در سال ۱۹۷۶ نشان داد که از مجموعه رئوس مستقل می‌توان برای رنگ‌کردن گراف با سه رنگ استفاده کرد [۱۱].

در حالت کلی و برای یک گراف دلخواه تا به حال الگوریتمی چندجمله‌ای برای یافتن مجموعه مستقل بیشینه پیدا نشده است. بنابراین محققان سعی داشتند این مسئله را یا در حالات خاص حل کنند یا مقدار آن را نسبت به جواب بهینه تقریب بزنند و یا به مقداری نزدیک به بهینه دست یابند. به عنوان مثال در سال ۱۹۶۵ موون و ماسر[16] نشان دادند که هر گراف با n رأس حداکثر $3^{n/3}$ کلیک بیشین[17] دارد [۱۲]. اگر S مجموعه رئوس مستقل بیشینه در گراف G باشد، آن گاه تعداد عناصر مجموعه مستقل بیشینه در گراف G برابر با تعداد رئوس زیرگراف کامل بیشینه[18] در گراف $\overline{G}$ (گراف مکمل[19] G) است که آن مجموعه را کلیک بیشین می‌نامند. در سال ۱۹۸۶ نیز لابی[20] یک الگوریتم موازی برای مسئله مجموعه رئوس مستقل ارائه داد [۱۳]. در مقاله پیش رو گراف تقاطع حاصل از در نظر گرفتن همه زوج رئوس اولیه مطرح می‌شود. در ریاضیات، گراف تقاطع گرافی است که تقاطع یا اشتراکی از مجموعه‌ها یا اشکال هندسی را توسط یک یال بین آن دو شکل نشان می‌دهد. به

1. Greedy Triangulation
2. Duppe & Gottschalk
3. Shamos & Hoey
4. Lloyd
5. Kirkpatrick
6. Garey & Johnson
7. Mulzer & Rote
8. Remy & Steger
9. Approximation
10. Meta-Heuristic
11. Heuristic
12. Maximum Independent Set
13. Intersection Graph
14. Maximum
15. Lawler
16. Moon & Moser
17. Maximal Clique
18. Maximum Clique
19. Complementary Graph
20. Luby



شکل ۱: گراف کامل با ۵ رأس.



شکل ۲: گراف تقاطع G_1' حاصل از گراف G_1.

عبارتی دیگر فرض کنید مجموعه S داده شده باشد و مجموعه زیرمجموعه‌های S را با C نشان دهیم. اگر به ازای هر عضو C یک رأس رسم کنیم و در صورتی که دو عضو C اشتراک داشته باشند بین رئوس متناظر با آنها یالی رسم شود، شکل حاصل گراف اشتراکی مجموع C نامیده شده و $I(C)$ نمایش داده می‌شود. از گراف تقاطع می‌توان در مسایل متفاوتی استفاده کرد. برای نمونه می‌توان در مسئله مدل‌سازی چراغ راهنمایی چندزمانه برای تقاطع خیابان‌ها استفاده کرد. هر گراف غیر جهت‌دار G ممکن است به صورت یک گراف تقاطع بیان شود. به این صورت که برای هر رأس v_i از G از مجموعه S_i یال‌هایی را که با v_i تداخل دارند در نظر بگیریم. سپس دو مجموعه اینچنینی یک تقاطع غیر تهی دارد اگر و فقط اگر رئوس مربوط در یک یال اشتراک داشته باشند. اردوس، پاول و گودمان[1] در سال ۱۹۶۶ ساختاری را ارائه می‌دهند که کارایی بیشتری دارد [۱۴]، به این معنا که تعداد کلی عناصر در مجموعه‌های ترکیب‌شده S_i کمتر است. تعداد عناصر مجموعه حداکثر $n^{2}/4$ است که n تعداد رئوس در گراف است. تا سال ۱۹۴۵ سزپیلراجن- مارکوزکی هم بر این باور بودند که همه گراف‌ها، گراف تقاطع هستند [۱۵].

در نظریه گراف‌ها، گراف خوش‌پوشش (WC)[2] گرافی غیر جهت‌دار است که در آن تمام مجموعه‌های رئوس مستقل بیشین دارای اندازه یکسان باشند. گراف‌های WC اولین بار توسط پلامر[3] بررسی شده‌اند [۱۶]. از آنجا که بین مجموعه همه مثلث‌بندی‌های روی یک مجموعه نقاط داده‌شده در صفحه و مجموعه همه مجموعه رئوس مستقل بیشین رئوس گراف تقاطع، یک نگاشت دوسویه (یک به یک و پوشا) وجود دارد، لذا هر مثلث‌بندی دقیقاً معادل یک مجموعه مستقل بیشین از رئوس گراف تقاطع است. همچنین از آنجا که همه مثلث‌بندی‌های مختلف یک مجموعه نقاط داده‌شده، دارای تعداد یال‌های یکسان با هم هستند و هر یال از هر مثلث‌بندی دقیقاً معادل یک رأس از مجموعه رئوس مستقل بیشین معادل با آن مثلث‌بندی است، بنابراین تعداد رئوس همه مجموعه‌های رئوس مستقل بیشین با هم برابر است و لذا گراف تقاطعی که در بخش الگوریتم‌های پیشنهادی از روی هر مجموعه نقاط داده‌شده ساخته می‌شود یک گراف WC است.

در این مقاله ابتدا گراف کامل حاصل از مجموعه رئوس موجود در صفحه در نظر گرفته می‌شود و سپس گراف تقاطع حاصل از برخورد یال‌های گراف کامل مشخص می‌شوند. به این صورت که گره‌های گراف تقاطع، یال‌های متقاطع در گراف کامل حاصل از مجموعه رئوس باشد و یال‌های آن به این صورت تعریف می‌شود که هر دو قطری که با هم متقاطع باشند یالی بینشان در گراف تقاطع در نظر گرفته می‌شود. در گراف تقاطع تعداد مجموعه رئوس مستقل را محاسبه می‌کنیم و سپس حالت‌های مختلف مجموعه رئوس مستقل بیشین را با توجه به گراف تقاطع پیدا می‌کنیم. با توجه به WCبودن گراف تقاطع، تعداد عناصر این مجموعه رئوس مستقل بیشین برابر با تعداد عناصر مجموعه رئوس مستقل بیشینه نیز می‌باشد. این مجموعه رئوس بیانگر یال‌هایی هستند که در گراف کامل باید باقی بمانند و سایر یال‌های متقاطع حذف شوند. به این صورت می‌توان تمام حالات مثلث‌بندی مجموعه رئوس را پیدا کرد و از بین این حالات آن حالتی که جمع مجموعه یال‌های آن کمترین باشد جواب مسئله است.

در این مقاله در بخش دوم تعاریف اولیه، در بخش سوم مراحل رویکرد جدید و همچنین یک روش مکاشفه‌ای برای تخمین تعداد مثلث‌بندی‌های امکان‌پذیر روی نقاط داده‌شده، در بخش چهارم مثال و نتایج پیاده‌سازی و در بخش پنجم نتیجه‌گیری ارائه شده است.

۲- تعاریف اولیه

در این بخش مفاهیم استفاده‌شده در این مقاله شرح داده می‌شوند.

تعریف ۱) گراف کامل گراف ساده‌ای است که در آن هر رأس به تمامی رأس‌های دیگر مجاور است. در گراف کامل با n رأس تعداد یال‌ها برابر $\binom{n}{2}$ است. شکل ۱ گراف کامل با ۵ رأس را نشان می‌دهد.

تعریف ۲) اگر G یک گراف تعبیه‌شده در صفحه با n رأس باشد، گراف تقاطع آن $G'=(V_{G'},E_{G'})$ نامیده می‌شود که در آن $V_{G'}$ یال‌های دارای تقاطع گراف G است و هر دو یال متقاطع در تعبیه مستقیم‌الخط گراف G موجب ایجاد یک یال در $E_{G'}$ خواهد شد.

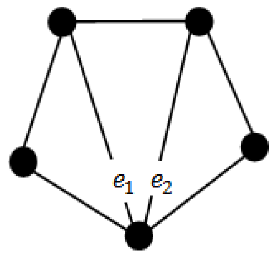
به عنوان مثال اگر گراف G_1 به صورت شکل ۱ در نظر گرفته شود، گراف شکل ۲، گراف تقاطع مربوط به G_1 را نشان می‌دهد.

تعریف ۳) مثلث‌بندی مجموعه رئوس S در فضای دوبعدی، بیشترین تعداد پاره‌خط‌هایی (یال‌هایی) است که بین رئوس S رسم می‌شوند به طوری که یال‌ها یکدیگر را قطع نکنند.

تعداد یال‌ها و مثلث‌های یک مثلث‌بندی به تعداد رئوس مستقر بر روی پوسته محدب وابسته است. اگر n تعداد رئوس مجموعه در صفحه و h تعداد رئوس پوسته محدب باشد، تعداد مثلث‌های هر مثلث‌بندی دلخواه از S بدون در نظر گرفتن وجه نامتناهی بیرونی برابر با مقدار $2n-h-2$ است. بنابراین تعداد یال‌های هر مثلث‌بندی دلخواه روی S برابر با مقدار $3n-h-3$ خواهد بود.

تعریف ۴) برای هر گراف دلخواه مانند G هر زیرمجموعه از رئوس آن که هیچ دو رأسی از آن مجموعه در گراف G مجاور نباشند، یک مجموعه مستقل از گراف G نامیده می‌شود. همچنین اگر نتوان رأس دیگری به مجموعه مستقل اضافه کرد به طوری که مستقل باقی بماند،

1. Erdos, Paul, and Goodman
2. Well-Covered
3. Plummer



شکل ۳: مجموعه نقاط مثلث‌بندی شده حاصل از حذف $\{e_3,e_4,e_5\}$ در گراف کامل G_1.

آن زیرمجموعه از رئوس برای گراف G یک مجموعه مستقل بیشین محسوب می‌شود. بزرگ‌ترین مجموعه مستقل بیشین از گراف G را مجموعه مستقل بیشینه روی گراف G می‌نامند.

در گراف شکل ۲ مجموعه رئوس $\{e_۱,e_۲\}$ یک مجموعه مستقل بیشین که بیشینه نیز می‌باشد محسوب می‌شود. مجموعه $\{e_۱\}$ نیز یک مجموعه مستقل است ولی بیشینه یا بیشین نمی‌باشد. گراف شکل ۲ چون یک گراف WC است هر مجموعه مستقل بیشین، بیشینه نیز می‌باشد. در این گراف پنج مجموعه مستقل بیشین وجود دارد که این پنج مجموعه $\{e_۲,e_۴\},\{e_۳,e_۵\},\{e_۲,e_۴\},\{e_۱,e_۲\},\{e_۱,e_۵\}$ هستند و لذا تعداد مثلث‌بندی‌های گراف شکل ۱ برابر مقدار ۵ است.

۳- الگوریتم‌های پیشنهادی

در این مقاله هدف، به دست آوردن الگوریتم و رابطه‌ای مناسب برای محاسبه تعداد حالات مجموعه رئوس مستقل بیشین در یک گراف WC است. همچنین نهایتاً این محاسبه منجر به محاسبه تعداد حالات مختلف مثلث‌بندی‌های یک مجموعه از نقاط داده‌شده در صفحه می‌شود. همچنین در صورت کم‌بودن تعداد نقاط می‌توان با تولید همه مثلث‌بندی‌های ممکن، به مثلث‌بندی بهینه مجموعه نقاط داده‌شده نیز رسید.

در ابتدا گراف کامل حاصل از n نقطه داده‌شده در نظر گرفته می‌شود و G نامیده می‌گردد. همان طور که گفته شد تعداد یال‌های گراف تقاطع $\binom{n}{۲}$ و تعداد یال‌های گراف مثلث‌بندی‌شده حاصل از همان n رأس برابر $۳n-h-۳$ است. پس باید تعداد R یال را از گراف کامل حذف کرد تا به یک مثلث‌بندی از مجموعه نقاط S رسید که R از (۱) تبعیت می‌کند

$$R=\binom{n}{۲}-(۳n-h-۳) \qquad (۱)$$

گراف تقاطع حاصل از گراف کامل G در نظر گرفته می‌شود و آن G' نامیده می‌گردد. گره‌های گراف تقاطع $V_{G'}$ همان یال‌های متقاطع گراف کامل هستند و هر دو یال متقاطع در G دارای یک یال بین رئوس مرتبط با آن دو یال در G' هستند. یال‌های گراف تقاطع $E_{G'}$ بیانگر تلاقی بین یال‌های متقاطع می‌باشند. از بین رئوس گراف تقاطع، باید به تعداد R گره حذف شود تا مثلث‌بندی ساخته شود. این تعداد رأس را می‌توان با قضیه پوشش رأسی[1] انتخاب کرد. این قضیه بیانگر این است که حداقل تعداد گره‌ها را در گراف انتخاب کنیم که همه یال‌های گراف پوشش داده شوند. در این صورت اگر تعداد R رأس را با استفاده از قضیه پوشش رأسی از گراف تقاطع حذف کنیم، تمام یال‌های گراف تقاطع پوشش داده می‌شوند و به این صورت باعث می‌شویم حداقل تعداد یال‌های متقاطع در گراف G حذف شود و گراف به حالت مثلث‌بندی شده تبدیل شود. بنابراین اگر تعداد گره‌های گراف تقاطع را n' بنامیم، پس در این صورت تعداد گره‌هایی که نباید حذف شوند (یا تعداد قطرهای متقاطع گراف کامل که باقی می‌مانند) برابر (۲) است که در آن $|MIS|$ نشان‌دهنده تعداد عناصر مجموعه رئوس مستقل بیشین است

$$|MIS|=n'-R \qquad (۲)$$

این تعداد رئوس در گراف تقاطع بیانگر مجموعه رئوس مستقل بیشین است یعنی با تعیین این رئوس در گراف، کل گراف پوشش داده می‌شود. با توجه به آن که در کلیت هر دو روش تفاوت کمی وجود دارد، در این مقاله از راه دوم استفاده شده است یعنی باید بر روی گراف تقاطع حاصل، حالات متفاوت مجموعه رئوس مستقل بیشین به تعداد $|MIS|$ پیدا شود. از آنجایی که محاسبه این مسئله نیز همانند مسئله پوشش رأسی در زمان نمایی حل می‌شود، پس محاسبه تعداد حالات آن که خود مسئله‌ای جداگانه است نیز باید بررسی شود. حالات متفاوت مجموعه مستقل بیانگر حالات متفاوت مثلث‌بندی مجموعه رئوس نیز است، پس در اصل در این الگوریتم‌ها نهایتاً حالات متفاوت انواع مثلث‌بندی بررسی می‌شود.

در این مقاله تفاوت دو الگوریتم در محاسبه حالات متفاوت مجموعه رئوس مستقل مطرح و مقایسه می‌شود. همچنین معادلاتی برای محاسبه تقریبی تعداد حالات متفاوت مجموعه رئوس مستقل ارائه شده است.

روند کلی الگوریتم اول به این صورت است که بعد از آن که تمام حالات مجموعه رئوس مستقل محاسبه شد، قطرهای هم‌نام با گره‌های موجود در مجموعه مستقل حاصل از گراف تقاطع را در گراف کامل نگاه داشته و سایر یال‌های متقاطع حذف می‌شوند. در این صورت اگر تمام حالات متفاوت مجموعه رئوس مستقل به دست آید، تمام حالات متفاوت مثلث‌بندی مجموعه رئوس ارائه‌شده به عنوان ورودی مسئله نیز محاسبه می‌شود.

به عنوان مثال بعد از حذف $\{e_۳,e_۴,e_۵\}$ و نگه‌داشتن $\{e_۱,e_۲\}$ در شکل ۲، مثلث‌بندی شکل ۳ ایجاد می‌شود. بعد از محاسبه تمام حالات مثلث‌بندی در تمام مثلث‌بندی‌های به دست آمده، مجموع وزن یال‌های مجموعه رئوس مثلث‌بندی‌شده محاسبه می‌شود و از این بین کمترین وزن به عنوان وزن بهینه انتخاب می‌شود. همان طور که گفته شد هدف این مقاله بررسی حالات متفاوت انواع مثلث‌بندی مجموعه رئوس ارائه‌شده با استفاده از بررسی حالات مجموعه رئوس مستقل گراف تقاطع حاصل از گراف کامل مجموعه نقاط ورودی است.

در الگوریتم پیشنهادی که مبتنی بر تکنیک برگشت به عقب است، ابتدا تمام رئوس را به عنوان کاندید در نظر می‌گیریم و در هر مرحله گره‌های مجاور هر رأس انتخابی حذف می‌شود تا نهایتاً در انتهای کار رأسی باقی نماند. این روند ادامه پیدا می‌کند تا همه رئوس یک بار به عنوان رأس اول انتخاب شوند و رئوس مجاور رأس انتخابی و یال‌های مجاور به آن رئوس حذف می‌شوند و همین روند برای مابقی رئوس مجدداً به صورت بازگشتی ادامه می‌یابد. شرط بازگشت از تابع بازگشتی، انتخاب‌شدن تعداد لازم از رئوس که تشکیل یک مجموعه مستقل بیشین می‌دهند است. شبه‌کد روش پیشنهادی در شکل ۴ آمده است. قابل ذکر است که در خط ۱۶ جهت جلوگیری از انتخاب حالات تکراری، هر بار رئوس بزرگ‌تر از آخرین رأس انتخابی جاری به عنوان گزینه‌هایی برای انتخاب در نظر گرفته می‌شوند. کوچک‌تری یا بزرگ‌تری نسبتی است که با توجه به اعدادی که به گره‌های گراف تقاطع در $v,code$ به ازای همه رئوس

1. Vertex Cover

```
Algorithm AllTriangulations;
Input:
S :Set; //The given point set of size n in the plane
Output:
  TotalNum:Integer; //Total Number of All
                    //Triangulations on S
  AllTriSet:Set of Set; //Set of All Triangulations on S
Begin
 1: Construct Intersection Graph G'(V',E')
    in which V'={v1,v2,...,v'n} corresponding to S
 2: Compute h //Number of extreme points of S
 3:
    MISSize ← n' − (n choose 2) + (3n − h − 3) //Size of MIS Set of G'
 4: AllTriSet ← Ø
 5: for i ← 1 to n' do
 6:   vi.code ← i
 7: endfor
 8: SelectedVerticesSet ← Ø
 9: TotalNum ← RecMIS(G', MISSize, 0)
End of Algorithm.
```

```
Function RecMIS(G, LocalMISSize, CurrentVertexCode);
Input:
  G:Graph;// Well-Covered Graph G(V,E) where k=|V|
          // such that V={v1,v2, ..., vk}
  LocalMISSize:Integer; //Size of any MIS Set of G
  CurrentVertexCode:Integer; // Code of the Current
                             //  Selected Vertex
Output:
  AllTriSet:Set of Set; // Global Set
  ReturnValue:Integer;  // Number of all MISs for G
Begin
10:if k = 0 and LocalMISSize = 0 then
11:  Add set SelectedVerticesSet to set of sets AllTriSet
12:  return 1  // an instance of MIS is generated
13:endif
14:TempNum ← 0
15:for i←1 to k - LocalMISSize + 1 do
16:  if vi.code > CurrentVertexCode then
17:    Add vi.code to SelectedVerticesSet
18:    TempV  ← V \ ({vi} ∪ Neighbors(vi))
19:    Let induced Subgraph of G corresponding to
                                        ...TempV be G'
20:    TempNum ← TempNum + RecMIS(G',
                       ...LocalMISSize - 1, vi.code)
21:    Delete vi.code from SelectedVerticesSet
22:  endif
23:endfor
25:return TempNum
End of Function;
```

شکل ۴: شبه‌کد الگوریتم پیشنهادی.

قرار داده شده بررسی می‌شود، لذا ترتیب انتخاب رئوس به گونه‌ای است که $code$ همه رئوس اضافه‌شده به مجموعه $SelectedVerticesSet$ تشکیل یک دنباله صعودی را می‌دهند.

تحلیل الگوریتم: از آنجا که انتخاب هر رأس از گراف تقاطع و قراردادن آن در مجموعه $SelectedVerticesSet$ و در پی آن حذف رئوس مجاور به آن، به معنی انتخاب یک پاره‌خط واصل بین دو نقطه از مجموعه نقاط ورودی و حذف کلیه پاره خطهای متقاطع با آن است. این کار در گراف تقاطع که یک گراف WC است منجر به تولید یک گراف WC با اندازه کوچک‌تر خواهد شد و هر انتخاب رأس از گراف تقاطع قطعاً منجر به مجموعه رئوس انتخاب‌شده غیر تکراری خواهد شد، لذا پیچیدگی زمانی الگوریتم به تعداد مجموعه‌های MIS که در خروجی تولید می‌شوند وابسته است و برای هر مجموعه به اندازه طول آن پردازش نیاز است. از این رو اگر تعداد مجموعه‌های مستقل بیشین در گراف تقاطع که همان تعداد مثلث‌بندی‌های مجموعه نقاط ورودی است را با t نشان دهیم، مرتبه زمانی الگوریتم برابر $\mathrm{O}(t.|MIS|)$ است. بر اساس (۱) و (۲) می‌توان نتیجه گرفت که مرتبه زمانی تابع RecMIS برابر $\mathrm{O}(t.n^{۲})$ است و از آنجا که مرتبه زمانی این تابع دارای پارامتری است که به اندازه خروجی بستگی دارد، لذا این تابع یک تابع حساس به خروجی[۱] است. خط شماره ۱ از الگوریتم که در آن گراف تقاطع ساخته می‌شود به تعداد تقاطع پاره‌خطهای واصل بین همه زوج نقاط یال دارد و از آنجا که در بدترین حالت هر ۴ نقطه یک تقاطع خواهند داشت لذا $\binom{n}{۴}$ یال خواهد داشت و ساخت گراف تقاطع $\mathrm{O}(n^{۴})$ زمان نیاز دارد. در نهایت کل الگوریتم دارای مرتبه زمانی $\mathrm{O}(n^{۴}+t.n^{۲})$ می‌باشد.

با توجه به آن که تعداد مثلث‌بندی‌های یک مجموعه از نقاط دلخواه در صفحه ممکن است به صورت نمایی بر حسب تعداد نقاط ورودی باشد، لذا در الگوریتم ارائه‌شده برای محاسبه حالات مختلف مجموعه رئوس مستقل، با افزایش تعداد رئوس گراف تقاطع زمان حل مسئله به صورت نمایی افزایش می‌یابد. از این رو روشی برای تقریب محاسبه تعداد مثلث‌بندی‌ها با زمان چندجمله‌ای ارائه می‌کنیم. برای محاسبه تخمینی تعداد حالات مختلف مجموعه مستقل بیشین در گراف تقاطع که معادل تعداد طرق مثلث‌بندی مجموعه نقاط اولیه است، روشی که در ادامه مطرح می‌شود می‌تواند روشی مکاشفه‌ای محسوب شود. از آنجا که تعداد عناصر مجموعه مستقل بیشین در گراف تقاطع $|MIS|$ است لذا به همین تعداد مرحله از بین رئوس باقیمانده گراف تقاطع یک رأس انتخاب می‌شود. اگر متوسط درجه هر یک از رئوس گراف تقاطع و تعداد رئوس گراف تقاطع در مرحله kام به ترتیب با d_k و n_k نشان داده شود، روابط بازگشتی (۳) و (۴) می‌توانند نشان‌دهنده نحوه محاسبه d_k و n_k باشند که در آن m و n' به ترتیب تعداد یال‌های گراف تقاطع و تعداد رئوس گراف تقاطع می‌باشند که در آن x_k متوسط تعداد یال‌های بین مجموعه رئوس حذف‌شده و مجموعه رئوس باقیمانده در گراف تقاطع (یعنی مجموعه $V_۱$ و مجموعه $V_۲$) می‌باشد که (۵) نحوه محاسبه آن را بیان می‌کند. همچنین (۶) متوسط تعداد حالات مجموعه مستقل بیشین را که با $\overline{|MIS|}$ نشان داده می‌شود مشخص می‌کند

$$n_k=\begin{cases} n' & ,\ k=۱ \\ n_{k-۱}-(d_{k-۱}+۱) & ,\ k>۱ \end{cases} \tag{۳}$$

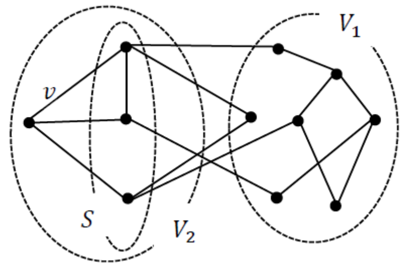
$$d_k=\begin{cases} \dfrac{۲m}{n_۱} & ,\ k=۱ \\ \dfrac{n_{k-۱}d_{k-۱}-d_{k-۱}(d_{k-۱}+۱)-x_{k-۱}}{n_k} & ,\ k>۱ \end{cases} \tag{۴}$$

$$x_k=\frac{\sum_{j=۰}^{\binom{d_k}{۲}} j}{\binom{d_k}{۲}}=\frac{۲+d_k(d_k+۱)}{۴} \tag{۵}$$

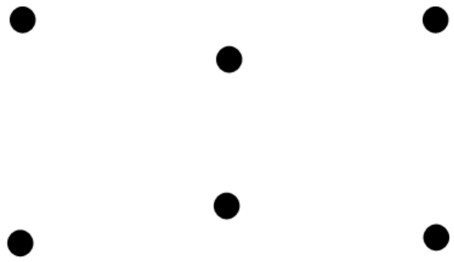
$$\overline{|MIS|}=\left\lceil \frac{\prod_{k=۱}^{|MIS|} n_k}{|MIS|!} \right\rceil \tag{۶}$$

قابل ذکر است که اگر در مرحله $k-۱$ام گراف تقاطع به صورت شکل ۵ فرض شود که در آن متوسط درجه رئوس $d_{k-۱}$ و تعداد رئوس $n_{k-۱}$

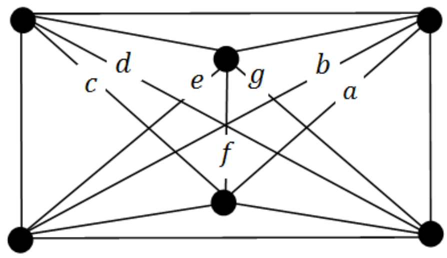
1. Output Sensitive



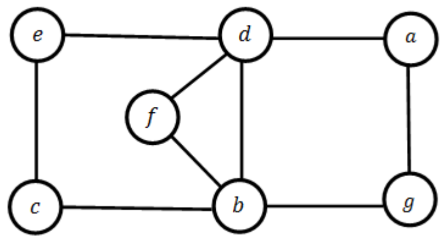
شکل ۵: انتخاب رأس v و حذف مجموعه V_2.



شکل ۶: مجموعه‌ای شامل ۶ نقطه در فضای دوبعدی.



شکل ۷: گراف کامل حاصل از اتصال همه زوج نقاط در شکل ۶

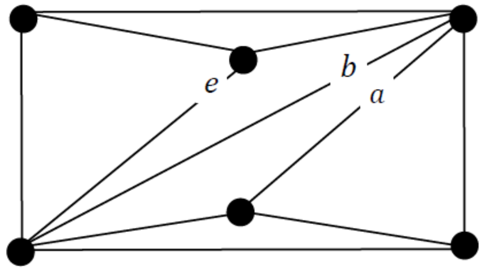


شکل ۸: گراف تقاطع حاصل از گراف شکل ۷.

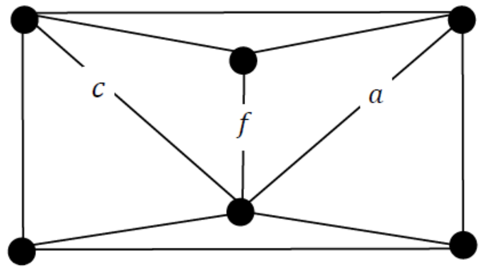
است، n_{k-1} حق انتخاب وجود دارد و با انتخاب رأسی مانند v از رئوس فعلی مجموعه رئوس V_2 که شامل رأس v و مجموعه رئوس S است حذف شده و لذا مقدار $d_{k-1}+1$ از تعداد رئوس که n_{k-1} است، کم می‌شود. همچنین از مجموع درجات رئوس که $n_{k-1}d_{k-1}$ است باید مقدار $d_{k-1}(d_{k-1}+1)$ که مجموع درجات مجموعه V_2 است به انضمام تعداد یال‌های بین مجموعه V_1 و مجموعه V_2 که x_{k-1} می‌باشد نیز کم شود که منجر به (۴) خواهد شد.

۴- نتایج الگوریتم و تعداد متوسط حالات مثلث‌بندی

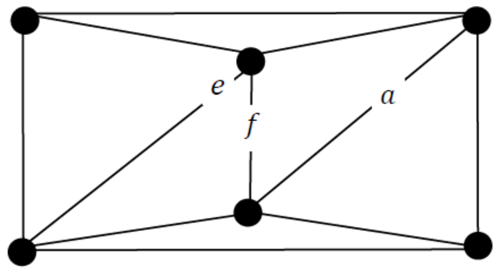
در شکل ۶ مجموعه‌ای از ۶ نقطه در صفحه نشان داده شده که برای این مجموعه، نقاط گراف کامل حاصل از اتصال همه زوج نقاط آن در شکل ۷ آمده است. یال‌های برچسب‌گذاری‌شده در شکل ۷ یال‌هایی هستند که حداقل یک تقاطع در آنها وجود دارد و لذا گراف تقاطع آن به شکل ۸ است. با محاسبه تمامی MISهای این گراف به ۶ مجموعه که تعداد عناصر همه آنها برابر با ۳ است خواهیم رسید که این MISها به صورت مجموعه‌ای از مجموعه‌های



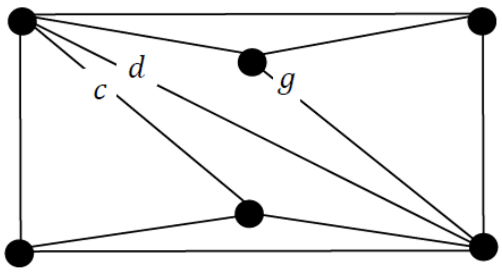
شکل ۹: مثلث‌بندی متناظر با $\{a,b,e\}$.



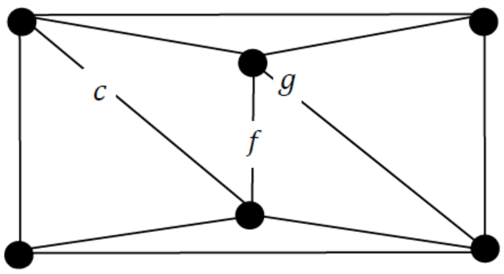
شکل ۱۰: مثلث‌بندی متناظر با $\{a,c,f\}$.



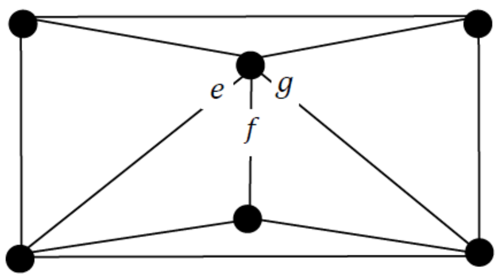
شکل ۱۱: مثلث‌بندی متناظر با $\{a,e,f\}$.



شکل ۱۲: مثلث‌بندی متناظر با $\{c,d,g\}$.



شکل ۱۳: مثلث‌بندی متناظر با $\{c,f,g\}$.



شکل ۱۴: مثلث‌بندی متناظر با $\{e,f,g\}$.

جدول ۱: نتایج پیاده‌سازی.

تعداد نقاط	تعداد رئوس گراف تقاطع	تعداد یال‌های گراف تقاطع	تعداد حالات دقیق انواع مثلث‌بندی	تعداد حالات محاسبه‌شده از (۶)
۵	۵	۵	۵	۵
۶	۹	۱۱	۱۴	۱۴
۷	۱۴	۳۵	۴۲	۵۰
۸	۲۰	۷۰	۱۳۲	۱۹۶
۱۰	۳۵	۲۱۰	۱۴۳۰	۱۳۵۰
۱۳	۶۵	۷۱۵	۵۸۷۸۶	۵۹۴۱۰
۱۵	۹۰	۱۳۶۵	۷۴۲۹۰۰	۷۴۴۳۷۷

$\{\{a,b,e\},\{a,c,f\},\{a,e,f\},\{c,d,g\},\{c,f,g\},\{e,f,g\}\}$ قابل تولید هستند. هر کدام از این MISها معادل با یک مثلث‌بندی یکتا است که به ترتیب در شکل‌های ۹ تا ۱۴ نشان داده شده‌اند.

از آنجا که برای محاسبه دقیق تعداد حالات مثلث‌بندی یک مجموعه نقاط دلخواه، الگوریتمی با زمان چندجمله‌ای وجود ندارد لذا می‌توان از روش مکاشفه‌ای مطرح‌شده طبق (۳) تا (۶) که در بخش قبلی ارائه شد استفاده کرد. با پیاده‌سازی الگوریتم مورد نظر به نتایج مناسبی که به مقدار دقیق نزدیک هستند رسیدیم که این نتایج در جدول ۱ ارائه شده‌اند.

۵- نتیجه‌گیری

هدف این مقاله طراحی الگوریتمی برای محاسبه تعداد طرق مثلث‌بندی روی مجموعه نقاط داده‌شده در ورودی و همچنین ارائه روشی برای محاسبه تخمینی این مقدار بود. در طراحی این دو الگوریتم از رویکردی جدید که تا به حال گزارشی از این رویکرد در پژوهش‌ها وجود نداشته است استفاده شد. این رویکرد موجب شد که بتوان مسئله‌های با دیدگاه مثلث‌بندی را با نگاه تبدیل مسئله به یک گراف و محاسبه مجموعه رئوس مستقل بیشین آن گراف نگریست. هر دو الگوریتم مطرح‌شده نیز از این رویکرد استفاده کردند، الگوریتم اول که دقیقاً تعداد طرق مثلث‌بندی را تولید می‌کند و نتایج پیاده‌سازی در الگوریتم دوم نشان‌دهنده جواب‌های نزدیک به مقدار واقعی است.

مراجع

[1] L. P. Chew, "Constrained delaunay triangulations," *Algorithmica*, vol. 4, no. 1, pp. 97-108,. 1986.

[2] D. Shojaee, H. Helali, and A. A. Alesheikh, "Triangulation for surface modeling," in *Proc. 9th Symp. on 3D Analysis of Human Movement,* poster session, Valenciennes, France, 28-30 Jun. 2006.

[3] R. D. Duppe and H. H. Gottschalk, "Automatische interpolation von isolinien bei willkürlichen stützpunkten," *Allgemeine Vermessungsnachrichten*, vol. 77, no. 10, pp. 423-426, 1970.

[4] M. I. Shamos and D. Hoey, "Closest-point problems," in *Proc. 16th IEEE Annual Symp. On Foundations of Computer Science,* pp. 151-162, USA, 13-15 Oct. 1975.

[5] E. L. Lloyd, "On triangulations of a set of points in the plane," in *Proc. 18th IEEE Annual Symp. on Foundations of Computer Science,* pp. 228-240, Providence, RI, USA, 31 Oct.-2 Nov. 1977.

[6] D. G. Kirkpatrick, "A note on delaunay and optimal triangulations," *Information Processing Letters*, vol. 10, no. 3, pp. 127-128, Apr. 1980.

[7] M. R. Garey, Computers and Intractability: A Guide to the Theory of NP-Completeness, 1978.

[8] W. Mulzer and G. Rote, "Minimum-weight triangulation is NP-hard," *J. of the ACM*, vol. 55, no. 2, pp. 1-29, 2008.

[9] J. Remy and A. Steger, "A quasi-polynomial time approximation scheme for minimum weight triangulation," *J. of the ACM,* vol. 56, no. 3, Article No.: 15, May 2009.

[10] M. Jahani, B. Sadeghi Bigham, and A. Askari, "An ant colony algorithm for the minimum weight triangulation," in *Proc. IEEE Int. Conf. on Computational Science and Its Applications,* vol. 1, pp. 81-85, Fukuoka, Japan, 23-26 Mar. 2010.

[11] E. L. Lawler, "A note on the complexity of the chromatic number problem," *Inf. Proc. Lett.*, vol. 5, pp. 66-67, 1976.

[12] J. W. Moon and L. Moser, "On cliques in graphs," *Israel J. of Mathematics*, vol. 3, no. 1, pp. 23-28, 1965.

[13] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM J. on Computing*, vol. 15, no. 4, pp. 1036-1053, Nov. 1986.

[14] P. Erdos, A. W. Goodman, and L. Posa, "The representation of a graph by set intersections," *Canadian J. of Mathematics*, vol. 18, pp. 106-112, 1966.

[15] E. S. Marczewski, "Sur deux propriétés des classes d'ensembles," *Fund. Math*, vol. 33, pp. 303-307, 1945. (in French)

[16] M. D. Plummer, "Some covering concepts in graphs," *J. of Combinatorial Theory*, vol. 8, no. 1, pp. 91-98, Jan. 1970.

علی نوراله مدرک کارشناسی ارشد و دکتری خود را به ترتیب در سال‌های ۱۳۷۷ و ۱۳۸۷ در رشته مهندسی کامپیوتر (نرم‌افزار) از دانشگاه صنعتی شریف و دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران) دریافت نموده است. ایشان در کلیه مقاطع تحصیلی (کارشناسی، کارشناسی ارشد و دکتری) دارای رتبه اول و ممتاز بوده است. زمینه تحقیقاتی ایشان الگوریتم‌های هندسه محاسباتی و حرکت روبات‌ها می‌باشد. ایشان در حال حاضر عضو هیأت علمی دانشگاه تربیت دبیر شهید رجایی و دارای رتبه استادیاری می‌باشد و دروسی نظیر اصول طراحی کامپایلر، نظریه زبان‌ها و ماشین‌ها، طراحی و پیاده سازی زبان‌های برنامه‌سازی، طراحی و تحلیل الگوریتم‌ها و ساختمان داده‌ها را در دوره کارشناسی و دروسی نظیر الگوریتم‌های هندسه محاسباتی، الگوریتم‌های موازی پیشرفته، الگوریتم‌های گراف و اصول رمزنگاری را در دوره کارشناسی ارشد تدریس می‌نماید.

زهرا رضایت مدرک کارشناسی و کارشناسی ارشد خود، در رشته مهندسی نرم افزار را به ترتیب در سال‌های ۱۳۹۵ و ۱۳۹۸ از دانشگاه تربیت دبیر شهید رجایی اخذ کرده است. وی در هر دو مقطع موفق به کسب رتبه اول شده است. زمینه‌های تحقیقاتی مورد علاقه ایشان الگوریتم‌های هندسه محاسباتی و مدل‌سازی فرمال است.