فصلنامه علمی- پژوهشی
فنّاوری اطلاعات و ارتباطات ایران

سال دوم، شماره‌های۳و۴، بهار و تابستان ۱۳۸۹
صص: ۱- ۷

تعبیه‌ی هندسی درخت درنقاط داخل یک چندضلعی با حداقل تعداد خم

اکرم سپهری▪* **علیرضا باقری****

*کارشناسی ارشد، دانشکده‌ی مهندسی برق، رایانه و فن‌آوری اطلاعات، دانشگاه آزاد اسلامی قزوین

**مربی، دانشکده‌ی مهندسی کامپیوتروفن‌آوری اطلاعات، دانشگاه صنعتی امیرکبیر

چکیده

دراین مقاله در نظر داریم تا یک درخت با N گره را روی N نقطه داخل یک چند ضلعی با n رأس تعبیه کنیم این تعبیه باید به گونه‌ای باشد که تعداد خم‌های درخت حاصل حداقل شود. ایده‌ی اصلی الگوریتم جدید مدل کردن مسئله به صورت مسئله‌ی تطبیق‌دهی گراف‌ها واستفاده از الگوریتم‌های تطبیق‌دهی گراف است که منجر به بررسی مسئله‌ی فاصله‌ی پیوندی و مسیر با حداقل تعداد لینک می شود، سپس با به کار بردن مفهوم تصحیح خطا ویافتن یک تابع هزینه‌ی مناسب و استفاده از روش تجزیه‌ی گراف‌ها، تطبیق‌دهی گراف‌ها را با حداقل هزینه برای به حداقل رساندن تعداد خم انجام می‌دهیم و الگوریتم دارای پیچیدگی محاسباتی O(N2n+N4) است.

کلیدواژگان: تعبیه‌ی هندسی، تعبیه‌ی درخت در مجموعه نقاط، به حداقل رساندن خم، تطبیق‌دهی گراف.

۱- مقدمه

محاسبه‌ی تعبیه‌ی یک ساختار بر روی مجموعه‌ای از نقاط و همچنین مسئله‌ی تصمیم‌گیری درباره‌ی این‌که آیا یک ساختار می‌تواند در مجموعه نقاط در روی یک صفحه تعبیه شود به طوری که برخی ویژگی‌ها ی خاص را داشته باشد موضوع مورد بحث دربسیاری زمینه‌هااز جمله ترسیم گراف[۱] است که نتایج به دست آمده در این موارد محدود است ودر [۱۳،۱۲،۹،۲] لیستی از مقالات دیده می شود. در[۶] مسئله ی محاسبه‌ی تعبیه‌ی صعودی مستقیم الخط از یک گراف غیردوری جهت دار[۲] G در مجموعه نقطه‌ی S بررسی شده است. مسئله‌ی تعبیه‌ی مستقیم الخط گراف در مجموعه نقاط، هم از دیدگاه الگوریتمیک وهم از دیدگاه محاسباتی بررسی شده است. در[۱۱] اثبات شده است که یک تعبیه‌ی مستقیم الخط از یک گراف درهر مجموعه از نقاط وجود دارداگر و تنها اگر آن گراف، یک گراف outer-planar باشد. از دیدگاه الگوریتمیک یک الگوریتم زمان O (Nlog3N) در [۴] برای ساخت تعبیه‌ی مستقیم الخط گراف‌های Outer-planar و درخت‌ها در مجموعه نقاط ارائه شده است. مسئله‌ی تصمیم‌گیری درباره‌ی این‌که یک تعبیه‌ی مستقیم الخط از یک گراف مسطح در یک مجموعه نقطه وجود دارد یک مسئله‌ی NP کامل است[۷]. ساختار مورد نظر ما درخت است، پیچیدگی مسئله‌ی تعبیه‌ی یک درخت آزاد در یک چند ضلعی،NP کامل است ودر[۳] اثبات شده است. در[۵] مسئله‌ی تعبیه‌ی درخت مورد بررسی قرار گرفته است یک درخت و یک مجموعه نقطه به عنوان ورودی مسئله هستند وتعبیه‌ی مستقیم الخطی از درخت در این مجموعه نقاط در زمان O (NlogN) ارائه شده است. این الگوریتم برای تعبیه‌ی درخت از قسمت‌هایی ازپوش محدب استفاده می‌کند. در [۱] مسئله‌ی تعبیه‌ی دو قسمتی درخت در یک مجموعه نقطه که به دو

▪ نویسنده عهده‌دار مکاتبات (akram_sepehri@yahoo.com ai)

1. Graph drawing
2. Directed acyclic graph

رنگ آبی و قرمز هستند بررسی شده است. دراین مقاله در نظر داریم تا یک درخت با N گره را روی N نقطه درون یک چند ضلعی با n رأس تعبیه کنیم این تعبیه باید به گونه‌ای باشد که تعداد خم‌های درخت حاصل حداقل شود و نیازی نیست که این تعبیه مسطح باشد. مسئله مطرح شده در این مقاله نسبتا جدید است و تحقیقات کمی روی آن انجام شده است. این مسئله با مسئله‌ی ترسیم گراف روی مجموعه‌ی نقاط با داشتن قسمتی از ترسیم مرتبط می‌شود که در [8] بیان شده است. در چنین تعبیه‌ای حد بالاو حد پایین برای تعداد خم دریال درخت ارائه شده است. در این مقاله یک الگوریتم جدید مبتنی بر مسئله‌ی تطبیق‌دهی گراف‌ها برای مسئله‌ی تعبیه‌ی درخت در مجموعه‌ی نقاط داخل چند ضلعی ارائه می دهیم. در این روش با اعمال یک الگوریتم که در[۱۴] ارائه شده است بر روی تمام نقاط داخل چند ضلعی با پیدا کردن فاصله ی پیوندی بین نقاط ودر نتیجه تعداد خم بین نقاط، یک گراف کامل وزن‌دار به عنوان گراف مرجع داریم که وزن نشان‌دهنده‌ی تعداد خم است. سپس با به کار بردن ایده‌ی مطرح شده در [۱۰] برای تطبیق‌دهی از روش تجزیه‌ی گراف‌ها استفاده می‌کنیم و تطبیق‌دهی بین گراف‌ها را در سطح زیر گراف‌های تجزیه شده انجام می‌دهیم.

در بخش۲ برخی تعاریف که برای الگوریتم پیشنهادی لازم است بیان می‌شود. در بخش ۳ به شرح الگوریتم وبیان روش تجزیه و ذکر یک مثال می‌پردازیم دربخش ۴ آنالیز پیچیدگی محاسباتی الگوریتم بررسی می شود و در بخش ۵ نتایج آزمایش ارائه شده است و در بخش ۶ نتیجه گیری ارائه می‌شود.

۲- تعاریف

حل مسئله‌ی مطرح شده نیازمند درک مفاهیم اولیه در مسائل مسیرهای با حداقل تعداد لینک و مسئله ی فاصله‌ی پیوندی و نیز مسئله‌ی تطبیق‌دهی گراف‌ها است بنابراین به مهم‌ترین مفاهیم در این بخش اشاره می‌شود.

- یک گراف $G = (V,E)$ ترکیبی از رئوس و یال‌ها است که V مجموعه‌ی رئوس است و $E \subset V \times V$ مجموعه‌ی یال‌ها در گراف است. اگر دو رأس $u,\ v \in V$ با یک یال $e \in E$ به هم متصل شده باشند می‌گوییم U,V مجاور یا همسایه هستند. گرافی که هر دو رأس مجزای آن مجاور باشند گراف کامل نامیده می‌شود.
- یک مسیردر یک گراف دنباله‌ای از رئوس است که از هر یک ازرئوسش یک یال به رأس بعدی در دنباله وجود دارد.
- یک دور مسیری است که رأس ابتدا و انتهای آن یکی است.
- یک درخت گرافی است که هر دو رأس آن دقیقا با یک مسیر به یکدیگر متصل شده باشد. به عبارت دیگر یک درخت، گراف همبندی است که دوری ندارد.
- چندضلعی ساده P در صفحه دنباله ای از n نقطه‌ی $v_1,v_2,\ldots v_n$. است که رئوس نامیده می شوند و به وسیله‌ی n پاره خط $v_1v_2,v_3v_4,\ldots v_{n-1}v_n,v_nv_1$ که یال نامیده می‌شوند مشخص می‌شود وهیچ دو یال متوالی اشتراک ندارند.
- اگر $G=(V,E)$ یک گراف مسطح با N گره و S یک مجموعه با N نقطه در یک صفحه باشد تعبیه‌ی گراف روی مجموعه نقاط که با $\Gamma(G,S)$ نشان داده می‌شود یک ترسیم مسطح از G روی نقاط S است که هر گره گراف G به یک نقطه‌ی مجزای S نگاشت شود. یال‌ها ممکن است خط مستقیم، منحنی یا چندخطی باشد.
- تطبیق‌دهی دو گراف $G_m=(V_m,E_m)$ و $G_t=(V_t,E_t)$ شامل ارتباط بین مجموعه‌ی رئوس V_m و V_t و مجموعه‌ی یال‌های E_m و E_t است. در تطبیق‌دهی گراف با ارائه‌ی دو گراف $G_m=(V_m,E_m)$ و $G_t=(V_t,E_t)$ مسأله پیدا کردن یک نگاشت $F:V_t \to V_m$ است، به طوری‌که $(u,v) \in E_t$ باشد اگر و تنها اگر $(f(u),f(v)) \in E_M$ باشد واگر این f وجود داشته باشد آن‌گاه می‌گوئیم G_t با G_m همریخت است.
- فاصله‌ی پیوندی بین دو نقطه‌یxوy حداقل تعداد پاره خط‌های مستقیم در مسیر متصل کننده‌ی xوy درون چندضلعی است. این فاصله را با $d_l(x,y)$ نشان می‌دهند. مسیر با حداقل تعداد لینک بین دو نقطه‌ی $x \in P$ و y دقیقا $d_l(x,y)$ یال دارد.

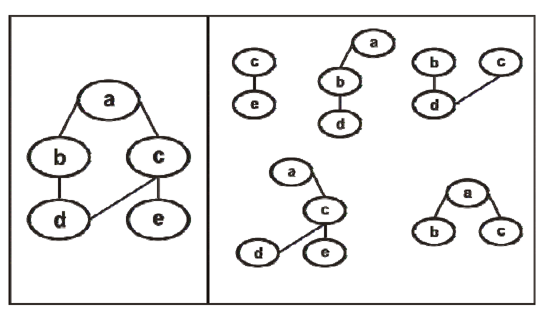
۳-شرح الگوریتم

ایده‌ی اصلی الگوریتم پیشنهادی به صورت زیر است. ما فاصله‌ی پیوندی بین هر زوج از نقاط را با اعمال الگوریتم ارائه شده در [۱۴] روی همه‌ی زوج نقاط درون چندضلعی ساده پیدا می‌کنیم. سپس ما یک گراف کامل وزن‌دار روی N نقطه می‌سازیم به طوری‌که وزن یال (u,v)، $dl(u,v)$ است. الگوریتم ارائه شده در [۱۴]، چندضلعی را در زمان $O(n)$ پیش پردازش می کند ودر زمان $O(\log n)$ به جستجوهای فاصله پیوندی پاسخ می‌دهد. سپس ما حداقل وزن تطبیق‌دهی درخت و گراف

کامل را پیدا می‌کنیم. بنابراین مسئله‌ی ما به مسئله‌ی تطبیق‌دهی گراف‌های وزن‌دار تبدیل می‌شود. برای تطبیق‌دهی ما از الگوریتم [۱۰] با اعمال برخی تغییرات استفاده می‌کنیم. در [۱۰] یک الگوریتم برای همریختی گراف‌ها ارائه شده است که به طور خلاصه شرح می‌دهیم. در این الگوریتم، گراف ورودی و مرجع به زیر گراف‌های کوچک‌تر تقسیم‌بندی می‌شود. گراف-های پایه گراف‌هایی به شکل ستاره هستند یعنی زیرگراف‌ها و همسایه هایش یک گراف پایه را می‌سازند. سپس یک ماتریس فاصله ی D بین گراف‌های پایه‌ی ورودی ساخته می‌شود که D_{ij} نشان‌دهنده‌ی فاصله‌ی بین i– امین گراف پایه‌ی گراف ورودی و j– امین گراف پایه در گراف مرجع است و به صورت زیر محاسبه می‌شود.

Distance *(U,V)* = *Wns*dist(ni, vj)* + min*(wbi, wbd)**abs *(k - p)*
\+ *min(wni, wnd)*abs(k - p)*
\+ dist *(b's,e's)* .

که *dist(ni, vj) فاصله‌ی بین برچسب های گره ni و vj* است. *p* و *k* تعدادیال‌های متصل شده به گره ریشه‌ی هر گراف پایه است که تطبیق می‌یابد و *dist (b's,e's)* حداقل وزن گراف دوقسمتی متناظر با فاصله‌ی یال‌ها است. w_{ns} هزینه‌ی جایگزینی برچسب‌های گره و w_{ni} هزینه‌ی اضافه کردن گره و w_{bd} هزینه‌ی حذف یال است. بعد از محاسبه‌ی ماتریس فاصله، یک گراف دوقسمتی وزن‌دار که معادل با ماتریس فاصله است ساخته می‌شود. گراف‌های پایه‌ی گراف ورودی یک بخش و گراف‌های پایه‌ی گراف مرجع بخش دیگر گراف دوقسمتی را تشکیل می‌دهند و حداقل وزن گراف دوقسمتی را که معادل با ماتریس فاصله است به عنوان هزینه‌ی تطبیق‌دهی گراف‌های ورودی و مرجع در نظر می‌گیریم. الگوریتم‌های زیادی برای محاسبه‌ی حداقل وزن گراف دوقسمتی وزن‌دار وجود دارد که می‌توان به [۱۵] اشاره کرد.



شکل۱: مثالی از تجزیه [۱۰]

۳-۱ تجزیه ی گراف [۱۰]

ایده‌ی اصلی برای تجزیه‌ی گراف‌ها این است که ساختار کوچک‌تر گراف برای فرآیند تطبیق‌دهی آسان‌تر است ودر نتیجه منجر به پیچیدگی کمتری می‌شود. گراف‌های حاصل از تجزیه، گراف‌های پایه هستند. در فر آیند تجزیه درهر گراف به تعداد گره‌ها، زیر ساختارهایی شکل می‌گیرد. به این صورت که هرزیرگراف، یک درخت یک سطحی است که شامل یک گره و همه‌ی گره‌هایی است که به این گره متصل شده است. یعنی هرگره وارتباطات آن گره با گره‌های مجاور تشکیل یک زیرگراف پایه را می‌دهد و این تجزیه را برای تمام گره‌های یک گراف انجام می‌دهیم. در شکل ۱ مثالی از تجزیه نشان داده شده است. در هر مرحله از تجزیه یک گره از روی گراف اولیه انتخاب شده است تا گره ریشه باشد و گراف پایه مطابق با ساختار اصلی گراف اولیه کامل می شود.

ما ایده‌ی اصلی در [۱۰] را به کار می‌بریم و فرآیند تطبیق‌دهی را با یک سری تغییرات، مبتنی برتجزیه‌ی گراف‌ها انجام می‌دهیم. ما ماتریس فاصله را با درنظرگرفتن بالاترین اولویت برای زیردرخت‌ها با بیشترین درجه پر می‌کنیم. برچسب سطر ها در ماتریس فاصله نشان دهنده‌ی گراف های پایه‌ی گراف ورودی وبرچسب ستون ها نشان دهنده‌ی گراف های پایه‌ی گراف مرجع است. با توجه به این نکته که وزن یال‌ها نشان‌دهنده‌ی تعداد خم است وزن یال‌ها در درخت ورودی صفراست. برای محاسبه‌ی فاصله‌ی بین دو زیر درخت در هر سطر ماتریس، حداقل اختلاف بین یال‌های زیردرخت ورودی و زیر درخت مرجع به عنوان هزینه‌ی تطبیق‌دهی در نظر گرفته می‌شود. اگر تعداد زیردرخت‌ها با بالاترین درجه بیشتر از یکی بود به صورت تصادفی از یکی از آن‌ها شروع می‌کنیم و تطبیق‌دهی را انجام می‌دهیم و به سطر انتخابی بعدی با بالاترین درجه می‌رویم و از نتایج به دست آمده در هرسطر برای پرکردن ماتریس فاصله در سطرهای بعدی استفاده می‌کنیم و در سایر موارد اگر تعداد زیر درخت‌ها با درجه‌ی یکسان، بیشتر از یک بود حداقل وزن گراف دو قسمتی به عنوان هزینه-ی تطبیق‌دهی گراف‌های پایه‌ی ورودی و مرجع در نظر گرفته می‌شود. هر زوج گراف پایه (یکی گراف پایه‌ی گراف ورودی و دیگری گراف پایه‌ی گراف مرجع) که وزن یال متصل کننده‌ی آن‌ها در محاسبه‌ی حداقل وزن گراف دوقسمتی به کار رفته است تطبیق می‌یابد، یعنی ریشه‌ی زیردرخت پایه‌ی ورودی با ریشه‌ی زیردرخت پایه‌ی مرجع تطبیق می‌یابد

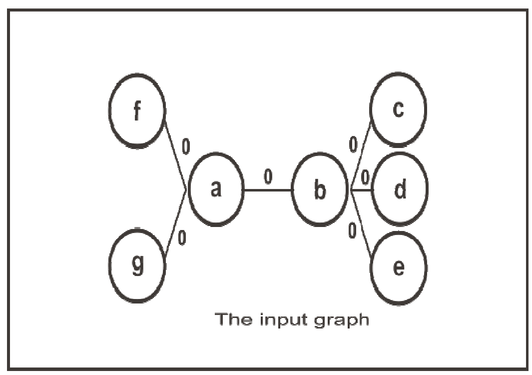
وسپس سطر وستون مربوط به آن را از ماتریس فاصله حذف می‌کنیم. ما از این نتایج برای تکمیل کردن ماتریس فاصله در سطر های بعدی استفاده می‌کنیم، یعنی باید در تخمین هزینه برای هر یک از فرزندان زیر درخت ورودی، وزن ارتباط گره فرزند با گره مرجعی که پدر با آن تطبیق یافته در نظر گرفته شود و باید به این نکته توجه کنیم که یال‌هایی را که در تخمین هزینه‌ی هر سطر استفاده می‌شوند نمی‌توان برای تخمین هزینه در سطر های بعدی به کار برد و این پروسیژر را تا تطبیق یافتن تمام گره‌های درخت ادامه می‌دهیم.

۳-۲- بررسی الگوریتم با یک مثال

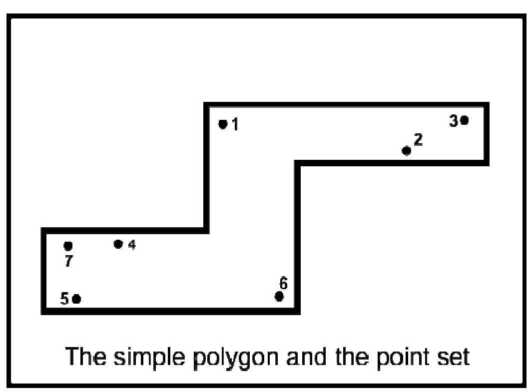
ما در این بخش الگوریتم پیشنهادی را با یک مثال بررسی می کنیم.

در نظر داریم تا درخت T با ۷ گره را روی ۷ نقطه درون یک چندضلعی ساده تعبیه کنیم. درخت، جندضلعی و مجموعه نقاط در شکل ۲و۳ نشان داده شده است.

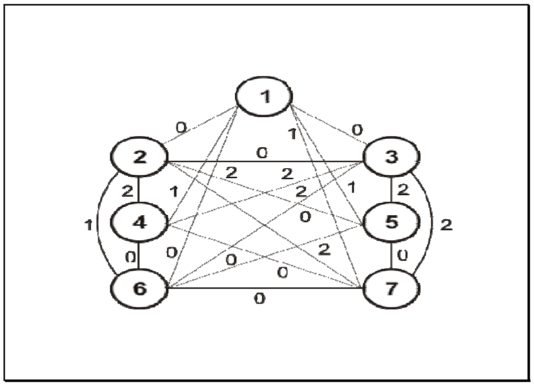
مرحله‌ی یک: با اجرای الگوریتم [۱۴] روی نقاط درون چند ضلعی و پیدا کردن فاصله‌ی پیوندی بین هر زوج از نقاط، گراف کامل وزن‌دار مانند شکل۴ حاصل می‌شود.



شکل۲: درخت ورودی



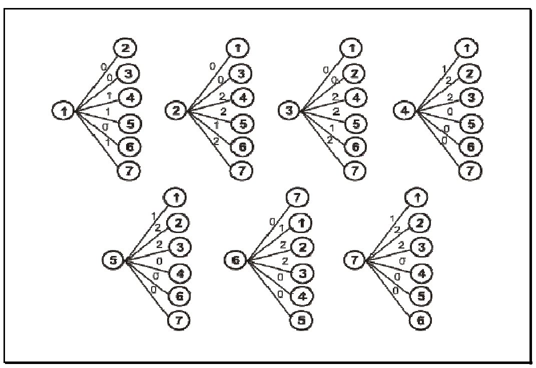
شکل۳: چندضلعی و مجموعه نقاط ورودی



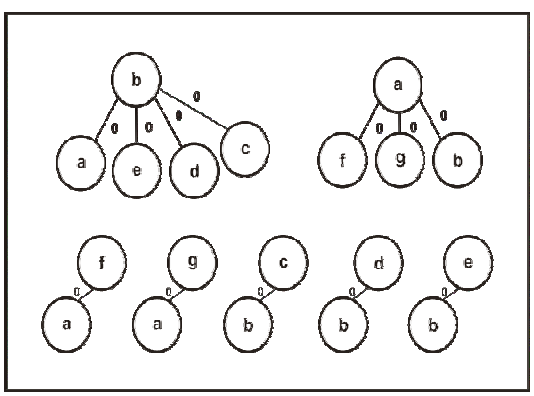
شکل۴: گراف وزن‌دار حاصل از اعمال ا لگوریتم بر روی نقاط درون چندضلعی

مرحله‌ی دو: با اعمال روش تجزیه‌ی گراف شرح داده شده در بخش ۱-۳ بر روی گراف مرجع و درخت ورودی، نتایج به صورت شکل۵ و۶ به دست آمده است.

مرحله‌ی سه و چهار: در این مرحله ماتریس فاصله بین گراف ورودی و مرجع ساخته می‌شود که معادل با فاصله‌ی بین گراف‌های پایه ورودی و مرجع است و در هر مرحله تطبیق‌دهی را انجام می‌دهیم.



شکل۵: زیر درخت های حا صل ازتجزیه‌ی گراف مرجع



شکل ۶: زیردرخت ها ی حاصل از تجزیه‌ی درخت ورودی

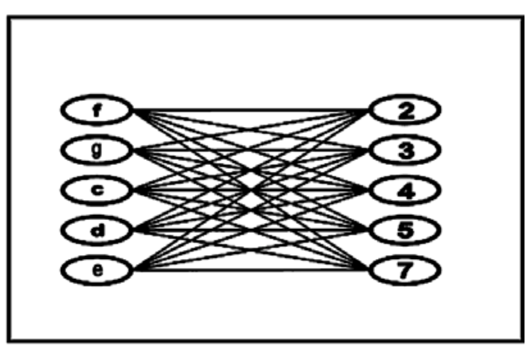
جدول ۱: ماتریس فاصله

DISTANCE MATRIX							
REFRENCE / INPUT							
a b c d e	1	3	3	1	1	0 1457	1
+6-1457 a f g b	0 623	×	×	4	4	×	4
+1 a f	×	0	0	1	1	×	1
+1 a g	×	0	0	1	1	×	1
+6 b c	×	1	1	0	0	×	0
+6 b d	×	1	1	0	0	×	0
+6 b e	×	1	1	0	0	×	0

همان‌طورکه ملاحظه می‌کنید ماتریس فاصله تکمیل می‌شود و تطبیق‌دهی انجام می‌شود. چون تعداد زیردرخت‌ها با درجه‌ی یک، بیشتر از یکی است گراف دو قسمتی مطابق شکل ۷ تشکیل شده است.

با اعمال الگوریتم پیشنهادی نتایج تطبیق‌دهی به صورت زیر به دست آمده است و تعداد خم، مطابق با شکل ۸ برابر صفر است که جواب بهینه است.

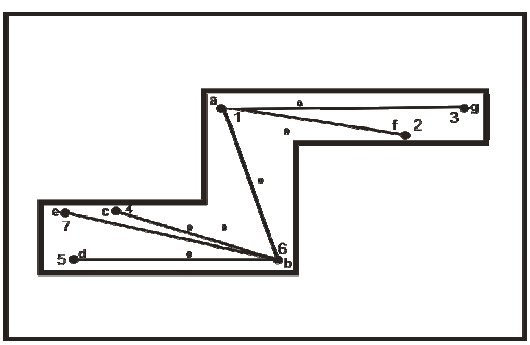
b → 6   a →1   →2   g→3   c→4   d→5   e→7



شکل ۷: گراف دوقسمتی برای زیردرخت‌های پایه با درجه‌ی یک

۴- آنالیز پیچیدگی محاسباتی الگوریتم پیشنهادی

مسئله‌ی تعبیه‌ی درخت در یک چندضلعی مسئله‌ی NP -کامل است. در این بخش به بررسی پیچیدگی محاسباتی الگوریتم پیشنهادی می‌پردازیم. فرض کنید که یک درخت با N گره و N نقطه درون یک چندضلعی با n رأس داریم، اولین مرحله در الگوریتم این است که باید تعداد خم بین هر نقطه درون چندضلعی را با سایر نقاط محاسبه کنیم که با اعمال الگوریتم [۱۴] این مرحله پیچیدگی محاسباتی $O(Nn+N^2 \log n)$ دارد.



شکل ۸: نتیجه‌ی حاصل از الگوریتم با صفر خم

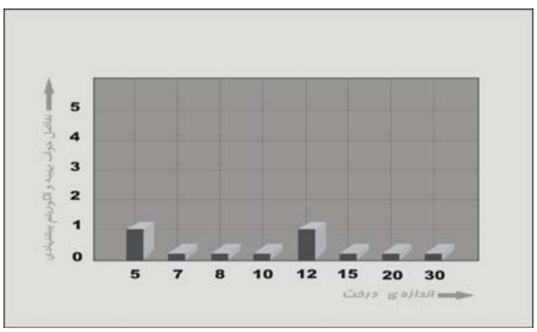
دومین مرحله، تجزیه‌ی درخت ورودی و گراف مرجع است که پیچیدگی محاسباتی این مرحله $O(N^2)$ است. مرحله‌ی سوم تکمیل ماتریس فاصله و ساخت گراف دوقسمتی وزن‌دارو انجام تطبیق‌دهی است که این مرحله دارای پیچیدگی $O(N^4)$ است. پس کل الگوریتم دارای پیچیدگی محاسباتی $O(N^2n+N^4)$ است.

۵-نتیجه‌ی آزمایش

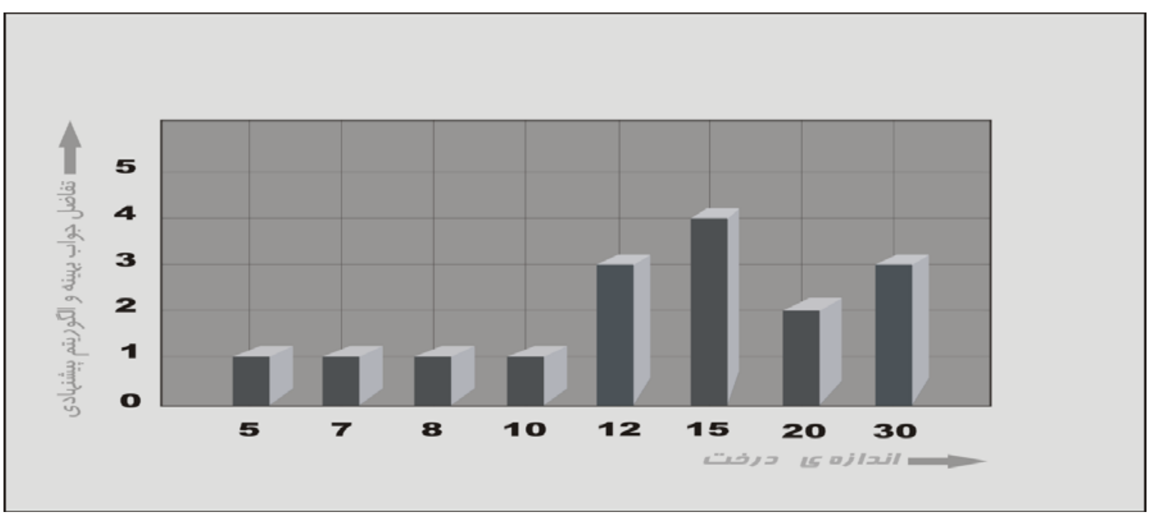
در این بخش الگوریتم پیشنهاد شده با درخت‌های ورودی با تعداد گره های متفاوت و انواع چندضلعی آزمایش شده است. با توجه به این که در الگوریتم، زیر درخت‌ها با بیشترین درجه اولویت بالاتری دارد نتایج آزمایشات نشان می‌دهد که افزایش اندازه‌ی درخت احتمال به دست آوردن جواب بهینه را کاهش نمی‌دهد و در مواردی که تعداد زیردرخت‌ها با بیشتربن درجه در درخت ورودی یکتا باشد الگوریتم بسیار مؤثر بوده و جواب‌های بهینه و یا نزدیک به بهینه ارائه می‌کند و با توجه به انتخاب تصادفی یک زیر درخت با بیشترین درجه اگر تعداد زیردرخت‌ها با بیشتربن درجه یکتا نباشد احتمال یافتن جواب بهینه در اولین انتخاب کم می‌شود و در برخی موارد تعداد خم، بیشتر از تعداد خم بهینه است، اما با توجه به این نکته که الگوریتم در هر مرحله روابط همسایگی را در نظر می‌گیرد و سعی در انتخاب همسایه‌ها باکم‌ترین تعداد خم برای هر زیر درخت دارد میانگین تعداد خم در هر یال کم است و جواب ارائه شده قابل قبول است. نتایج آزمایشات در شکل های ۹، ۱۰و ۱۱ آمده است.

۶- نتیجه

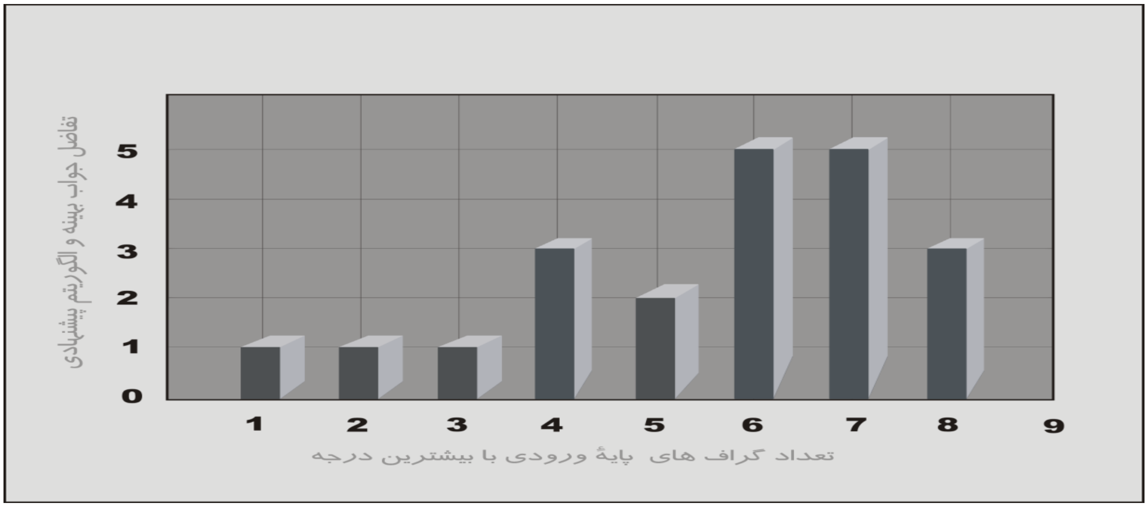
در این مقاله یک الگوریتم برای تعبیه ی درخت T با N گره روی N نقطه درون چندضلعی ساده با n رأس با حداقل تعداد خم پیشنهاد شده است. ایده‌ی اصلی الگوریتم جدید، مدل کردن مسئله به مسئله‌ی تطبیق‌دهی گراف است که منجر به بررسی مسئله‌ی فاصله پیوندی ومسیر با حداقل تعداد لینک می‌شود. الگوریتم ما برای حل مسئله از تکنیک‌های اخیر در مسئله‌ی فاصله‌ی پیوندی و مسئله تطبیق‌دهی گراف استفاده می‌کند. الگوریتم با در نظر گرفتن روابط همسایگی بین رئوس درخت و استفاده از مفهوم تصحیح خطا و پیدا کردن یک تابع هزینه‌ی مناسب یک را حل مؤثر برای مسئله‌ی مطرح شده ارائه می‌کند. الگوریتم در بیشتر موارد جواب‌های بهینه یا نزدیک به بهینه ارائه می‌کند ودارای پیچیدگی محاسباتی $O(N^2n+N^4)$ است .



شکل ۹: نتیجه‌ی اجرای الگوریتم بر روی درخت‌هایی که زیر درخت با بیشترین درجه‌ی آنها یکتا است.



شکل ۱۰: نتیجه‌ی اجرای الگوریتم بر روی درخت‌هایی که زیردرخت با بیشترین درجه‌ی آن‌ها یکتا نیست .



شکل۱۱: نمایش تعداد زیردرخت‌ها با بیشترین درجه در هر درخت.

مراجع

[1] Abellanas M, Garcia-Lopez J, Hernnandez G, Noy M, P. Ramos A. Bipartite embeddings of trees in the plane. Discrete Applied Mathematics, pp: 1-10,1999.

[2] Badent M, Giacomo E.D, Liotta G. Drawing colored graphs on colored points, in Proceedings of the Tenth Workshop on Algorithms and Data Structures, pp: 129-142,2008.

[3] Bagheri A.R, Razzazi M.R. On complexity of planar embeddability of trees inside simple polygons. Information processing letters, pp: 521-523,2010.

[4] Bose P. On embedding an outer-planar graph on a point set,computational Geometry: Theory and Applications, pp: 303–312,2002.

[5] Bose P, McAllister M, Snoeyink J. Optimal algorithms to embed trees in a point set, Journal of Graph Algorithms and Applications, pp: 1–15, 1997.

[6] Binucci C , Di Giacomo E , Didimo W , Estrella-Balderrama A, Frati F, GKobourov S, Liotta G. Upward straight-line embeddings of directed graphs into point sets, Computational Geometry, pp: 219–232, 2010.

[7] Cabello S. Planar embeddability of the vertices of a graph using a fixed point set is NP-hard, J. Graph Algorithms Appl, pp:353–366,2006.

[8] Giacomo E.D, Didimo W, Liotta G, Meijer H, Wismath S.k. Point-set embeddings of trees with given partial drawings, Computational Geometry, pp: 664-676, 2009.

[9] Giacomo E. D, Liotta G, Trotta F. On embedding a graph on two sets of points, International Journal of Foundations of Computer Science, pp: 1071–1094, 2006.

[10] El-sonbaty Y, ismail M. A. A new algorithm for subgraph optimal Isomorphism, Pattern Recognition Society, published by Elsevier Science, pp: 205-218, 1997.

[11] Gritzmann P, Mohar B, Pach J, Pollack R. Embedding a planar triangulation with vertices at specified positions. The American Mathematical Monthly 98, pp: 165–166, 1991.

[12] Kaufmann M, Wiese R. Embedding vertices at points: Few bends suffice for planar graphs, Journal of Graph Algorithms and Applications, pp: 115–129, 2002.

[13] Pach J, Wenger R. Embedding planar graphs at fixed vertex locations, Graphs and Combinatory, pp: 717–728, 2001.

[14] Suri S. A linear time algorithm for minimum link paths inside a simple polygon, Computer Vision Graphics and Image Process, pp: 99-110, 1986.

[15] Yamada S, Hasai T. An efficient algorithm for the linear assignment problem, Elect. Comm, pp: 28-36, 1990.