

تعییه‌ی هندسی درخت در نقاط 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 کامپیوتر و فن‌آوری اطلاعات، دانشگاه صنعتی امیرکبیر

چکیده

در این مقاله در نظر داریم تا یک درخت با N گره را روی N نقطه داخل یک چند ضلعی با n رأس تعییه کنیم این تعییه باید به گونه‌ای باشد که تعداد خم‌های درخت حاصل حداقل شود. ایده‌ی اصلی الگوریتم جدید مدل کردن مسئله به صورت مسئله‌ی تطبیق‌دهی گراف‌ها و استفاده از الگوریتم‌های تطبیق‌دهی گراف است که منجر به بررسی مسئله‌ی فاصله‌ی پیوندی و مسیر با حداقل تعداد لینک می‌شود، سپس با به کار بردن مفهوم تصحیح خطا و یافتن 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 $O(N^2n+N^4)$ است.

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

۱- مقدمه

محاسبه‌ی تعییه‌ی یک ساختار بر روی مجموعه‌ای از نقاط و همچنین مسئله‌ی تصمیم‌گیری درباره‌ی این که آیا یک ساختار می‌تواند در مجموعه نقاط در روی یک صفحه تعییه شود به طوری که برخی ویژگی‌ها ی خاص را داشته باشد موضوع

مورد بحث در بسیاری زمینه‌ها از جمله ترسیم گراف^۲ است که نتایج به دست آمده در این موارد محدود است و در [۱۳، ۱۲، ۹، ۲] لیستی از مقالات دیده می‌شود. در [۶] مسئله‌ی محاسبه‌ی تعییه‌ی صعودی مستقیم الخط از یک گراف غیردوری جهت دار G^3 در مجموعه نقطه‌ی S بررسی شده است. مسئله‌ی تعییه‌ی مستقیم الخط گراف در مجموعه نقاط، هم از دیدگاه الگوریتمیک و هم از دیدگاه محاسباتی بررسی شده است. در [۱۱] اثبات شده است که یک تعییه‌ی مستقیم الخط از یک گراف در هر مجموعه از نقاط وجود دارد اگر و تنها اگر آن گراف، یک گراف Outer-planar باشد. از دیدگاه الگوریتمیک یک الگوریتم زمان $O(N \log 3N)$ در [۴] برای ساخت تعییه‌ی مستقیم الخط گراف‌های Outer-planar و درخت‌ها در مجموعه نقاط ارائه شده است. مسئله‌ی تصمیم‌گیری درباره‌ی این که یک تعییه‌ی 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 تعییه‌ی یک درخت آزاد در یک چند ضلعی، NP کامل است و در [۳] اثبات شده است. در [۵] مسئله‌ی تعییه‌ی درخت مورد بررسی قرار گرفته است یک درخت و یک مجموعه نقطه به عنوان ورودی مسئله هستند و تعییه‌ی مستقیم الخطی از درخت در این مجموعه نقاط در زمان $O(N \log N)$ ارائه شده است. این الگوریتم برای تعییه‌ی درخت از قسمت‌هایی از پوش محدب استفاده می‌کند. در [۱] مسئله‌ی تعییه‌ی 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

- یک مسیردر یک گراف دنباله‌ای از رئوس است که از هر یک از رئوسش 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 دقیقاً با یک مسیر به یکدیگر متصل شده باشد. به عبارت دیگر یک درخت، گراف همبندی است که دوری ندارد.
- چندضلعی ساده P در صفحه دنباله‌ای از n نقطه‌ی v_1, v_2, \dots, v_n است که رئوس نامیده می‌شوند و به وسیله‌ی n پاره خط $v_1v_2, v_2v_3, \dots, v_{n-1}v_n, v_nv_1$ یال نامیده می‌شوند مشخص می‌شود و هیچ 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 تعییه‌ی 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 $F: V_t \rightarrow V_m$ است، به طوری که $(u,v) \in E_t$ باشد اگر و تنها اگر $(f(u), f(v)) \in E_m$ باشد و اگر این f وجود داشته باشد آن‌گاه می‌گوئیم G_t با G_m هم‌ریخت است.
- فاصله‌ی پیوندی بین دو نقطه‌ی x و y حداقل تعداد پاره خط-های مستقیم در مسیر متصل کننده‌ی x و y درون چندضلعی است. این فاصله را با $d_t(x,y)$ نشان می‌دهند. مسیر با حداقل تعداد لینک بین دو نقطه‌ی $x \in P$ و y دقیقاً $d_t(x,y)$ 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 طوری که وزن یال (u,v) ، $d_t(u,v)$ است. الگوریتم ارائه شده در [۱۴]، چندضلعی را در زمان $O(n)$ پیش پردازش می‌کند و در زمان $O(\log n)$ به جستجوهای فاصله پیوندی پاسخ می‌دهد. سپس ما حداقل وزن تطبیق‌دهی درخت و گراف

رنگ آبی و قرمز هستند بررسی شده است. در این مقاله در نظر داریم تا یک درخت با N گره را روی N نقطه درون یک چند ضلعی با n رأس تعییه کنیم این تعییه باید به گونه‌ای باشد که تعداد خم‌های درخت حاصل حداقل شود و نیازی نیست که این تعییه مسطح باشد. مسئله مطرح شده در این مقاله نسبتاً 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 تعییه‌ای حد بالا و حد پایین برای تعداد خم در یال درخت ارائه شده است. در این مقاله یک الگوریتم جدید مبتنی بر مسئله‌ی تطبیق‌دهی گراف‌ها برای مسئله‌ی تعییه‌ی درخت در مجموعه‌ی نقاط داخل چند ضلعی ارائه می‌دهیم. در این روش با اعمال یک الگوریتم که در [۱۴] ارائه شده است بر روی تمام نقاط داخل چند ضلعی با پیدا کردن فاصله‌ی پیوندی بین نقاط و در 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 بخش ۲ برخی تعاریف که برای الگوریتم پیشنهادی لازم است بیان می‌شود. در بخش ۳ به شرح الگوریتم و بیان روش تجزیه و ذکر یک مثال می‌پردازیم در بخش ۴ آنالیز پیچیدگی محاسباتی الگوریتم بررسی می‌شود 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 مسیری‌های با حداقل تعداد لینک و مسئله‌ی 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.

۳-۱ تجزیه ی گراف [۱۰]

ایده‌ی اصلی برای تجزیه‌ی گراف‌ها این است که ساختار کوچک‌تر گراف برای فرآیند تطبیق‌دهی آسان‌تر است و در نتیجه منجر به پیچیدگی کمتری می‌شود. گراف‌های حاصل از تجزیه، گراف‌های پایه هستند. در فرآیند تجزیه در هر گراف به تعداد گره‌ها، زیر ساختارهایی شکل می‌گیرد. به این صورت که هر زیرگراف، یک درخت یک سطحی است که شامل یک گره و همه‌ی گره‌هایی است که به این گره متصل شده است. یعنی هر گره و ارتباطات 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 می‌شود.

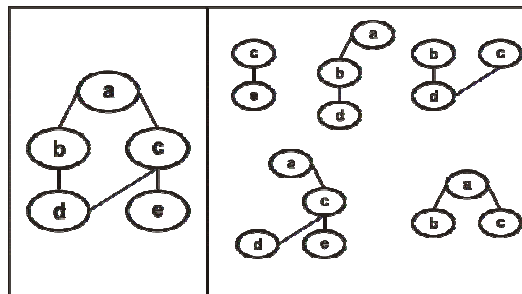
ما ایده‌ی اصلی در [۱۰] را به کار می‌بریم و فرآیند تطبیق‌دهی را با یک سری تغییرات، مبتنی بر تجزیه‌ی گراف‌ها انجام می‌دهیم. ما ماتریس فاصله را با در نظر گرفتن بالاترین اولویت برای زیردرخت‌ها با بیشترین درجه پر می‌کنیم. برچسب سطرها در ماتریس فاصله نشان دهنده‌ی گراف‌های پایه‌ی گراف ورودی و برچسب ستون‌ها نشان دهنده‌ی گراف‌های پایه‌ی گراف ورودی و برچسب سطرها در ماتریس فاصله نشان دهنده‌ی گراف‌های پایه‌ی گراف ورودی و برچسب ستون‌ها نشان دهنده‌ی گراف‌های پایه‌ی گراف ورودی است. با توجه به این نکته که وزن یال‌ها نشان‌دهنده‌ی تعداد خم است وزن یال‌ها در درخت ورودی صفر است. برای محاسبه‌ی فاصله‌ی بین دو زیردرخت 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 هر سطر 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 هزینه‌ی 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 دو قسمتی به کار رفته است تطبیق می‌یابد، یعنی ریشه‌ی زیردرخت پایه‌ی ورودی با ریشه‌ی زیردرخت پایه‌ی مرجع تطبیق می‌یابد

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 همسایه‌هایش یک گراف پایه را می‌سازند. سپس یک ماتریس فاصله‌ی 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.

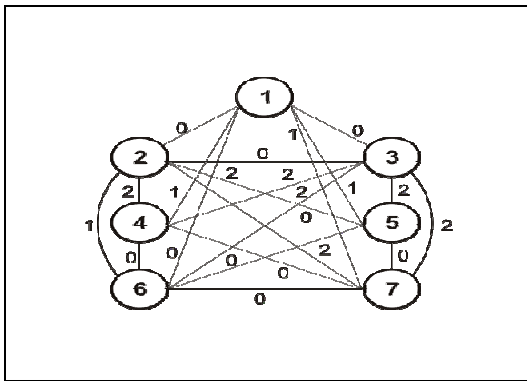
$$\text{Distance } (U, V) = W_{ns} * \text{dist}(ni, vj) + \min(wbi, wbd) * \text{abs}(k - p) + \min(wni, wnd) * \text{abs}(k - p) + \text{dist}(b's, e's)$$

که $\text{dist}(ni, vj)$ فاصله‌ی بین برچسب‌های گره ni و vj

است. k و p تعداد یال‌های متصل شده به گره ریشه‌ی هر گراف پایه است که تطبیق می‌یابد و $\text{dist}(b's, e's)$ حداقل وزن گراف دو قسمتی 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 دو قسمتی وزن‌دار که معادل با ماتریس فاصله است ساخته می‌شود. گراف‌های پایه‌ی گراف ورودی یک بخش و گراف‌های پایه‌ی گراف مرجع بخش دیگر گراف دو قسمتی را تشکیل می‌دهند و حداقل وزن گراف دو قسمتی 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 دو قسمتی وزن‌دار وجود دارد که می‌توان به [۱۵] اشاره کرد.

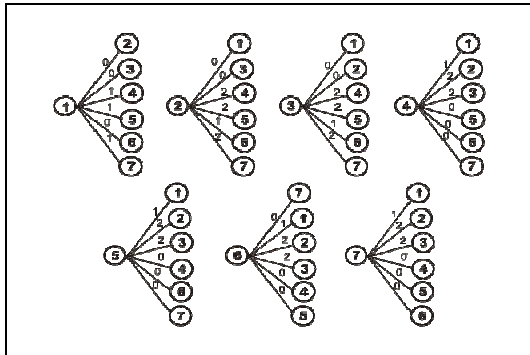


شکل ۱: مثالی از تجزیه [۱۰]

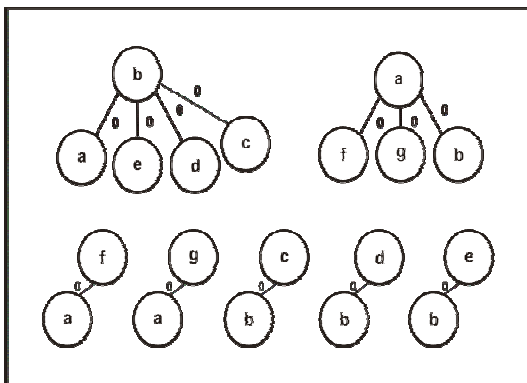


شکل ۴: گراف وزن دار حاصل از اعمال الگوریتم بر روی نقاط درون چندضلعی

مرحله دو: با اعمال روش تجزیه ی گراف شرح داده شده در بخش ۱-۳ بر روی گراف مرجع و درخت ورودی، نتایج به صورت شکل ۵ و ۶ به دست آمده است. مرحله سه و چهار: در این مرحله ماتریس فاصله بین گراف ورودی و مرجع ساخته می شود که معادل با فاصله بین گراف های پایه ورودی و مرجع است و در هر مرحله تطبیق دهی را انجام می دهیم.



شکل ۵: زیر درخت های حاصل از تجزیه ی گراف مرجع



شکل ۶: زیردرخت های حاصل از تجزیه ی درخت ورودی

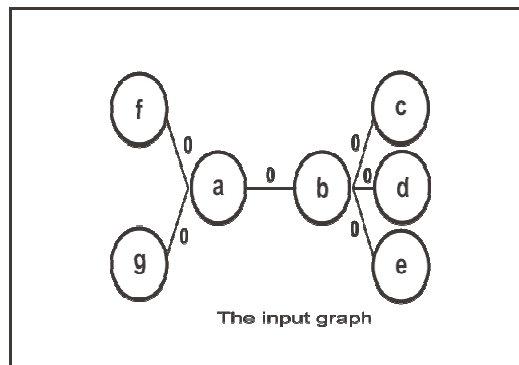
وسپس سطر وستون مربوط به آن را از ماتریس فاصله حذف می کنیم. ما از این نتایج برای تکمیل کردن ماتریس فاصله در سطر های بعدی استفاده می کنیم، یعنی باید در تخمین هزینه برای هر یک از فرزندان زیر درخت ورودی، وزن ارتباط گره فرزند با گره مرجعی که پدر با آن تطبیق یافته در نظر گرفته شود و باید به این نکته توجه کنیم که یال هایی را که در تخمین هزینه ی هر سطر استفاده می شوند نمی توان برای تخمین هزینه در سطر های بعدی به کار برد و این پروسیژر را تا تطبیق یافتن تمام گره های درخت ادامه می دهیم.

۳-۲- بررسی الگوریتم با یک مثال

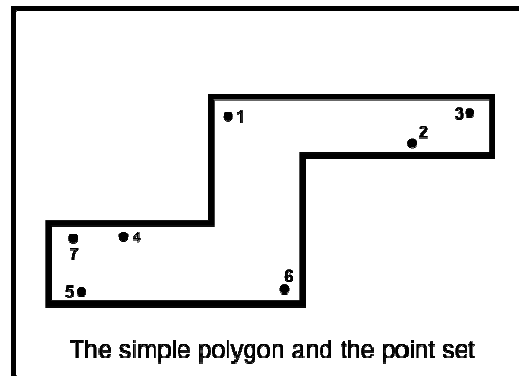
ما در این بخش الگوریتم پیشنهادی را با یک مثال بررسی می کنیم.

در نظر داریم تا درخت T با ۷ گره را روی ۷ نقطه درون یک چندضلعی ساده تعییه کنیم. درخت، چندضلعی و مجموعه نقاط در شکل ۲ و ۳ نشان داده شده است.

مرحله یک: با اجرای الگوریتم [۱۴] روی نقاط درون چند ضلعی و پیدا کردن فاصله ی پیوندی بین هر زوج از نقاط، گراف کامل وزن دار مانند شکل ۴ حاصل می شود.



شکل ۲: درخت ورودی



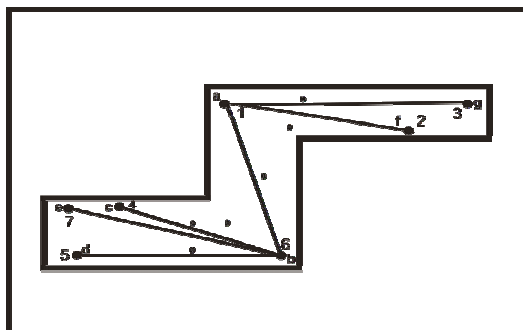
شکل ۳: چندضلعی و مجموعه نقاط ورودی

جدول ۱: ماتریس فاصله

DISTANCE MATRIX								
REFERENCE INFD								
	1	3	3	1	1	0 1457	1	
	0 623	×	×	4	4	×	4	
	×	0	0	1	1	×	1	
	×	0	0	1	1	×	1	
	×	1	1	0	0	×	0	
	×	1	1	0	0	×	0	
	×	1	1	0	0	×	0	

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.

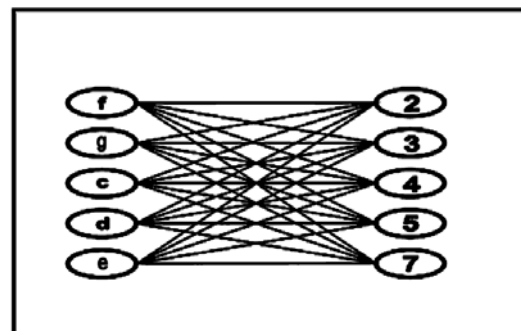


شکل ۸: نتیجه‌ی حاصل از الگوریتم با صفر خم

همان‌طور که 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.

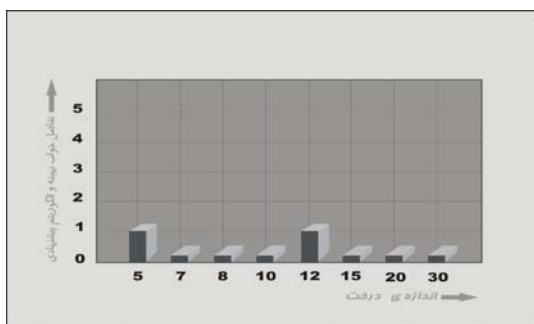
$b \rightarrow 6 \quad a \rightarrow 1 \rightarrow 2 \quad g \rightarrow 3 \quad c \rightarrow 4 \quad d \rightarrow 5 \quad e \rightarrow 7$



شکل ۷: گراف دو قسمتی برای زیردرخت‌های پایه با درجه‌ی یک

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. ایده ی اصلی الگوریتم جدید، مدل کردن مسئله به مسئله ی تطبیق دهی گراف است که منجر به بررسی مسئله ی فاصله پیوندی و مسیر با حداقل تعداد لینک می شود. الگوریتم ما برای حل مسئله از تکنیک های اخیر در مسئله ی فاصله ی پیوندی و مسئله تطبیق دهی گراف استفاده می کند. الگوریتم با در نظر گرفتن روابط همسایگی بین رئوس درخت و استفاده از مفهوم تصحیح خطا و پیدا کردن یک تابع هزینه ی مناسب یک را حل مؤثر برای مسئله ی مطرح شده ارائه می کند. الگوریتم در بیشتر موارد جواب های بهینه یا نزدیک به بهینه ارائه می کند و دارای پیچیدگی محاسباتی $O(N^2n+N^4)$ است.

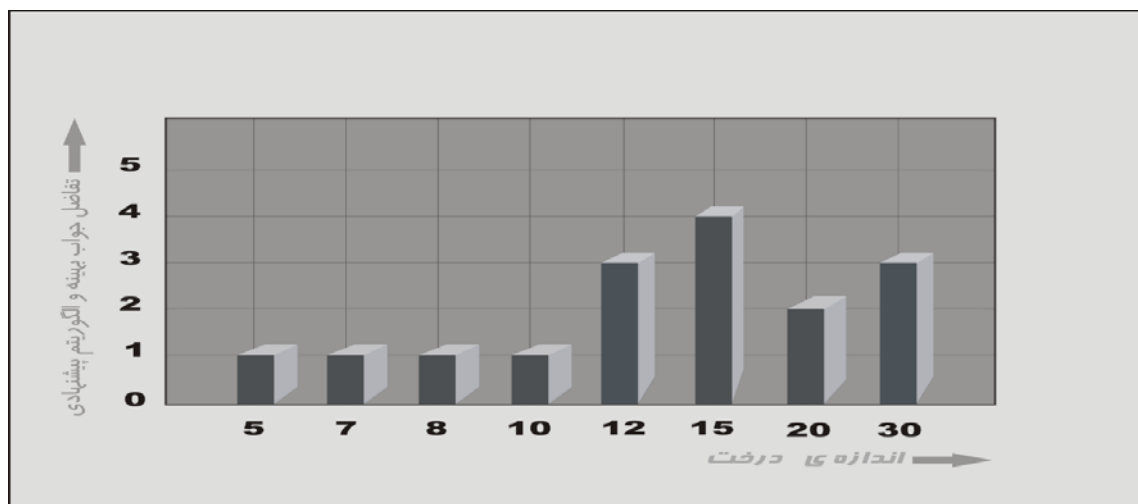


شکل ۹: نتیجه ی اجرای الگوریتم بر روی درخت هایی که زیر درخت با بیشترین درجه ی آنها یکتا است.

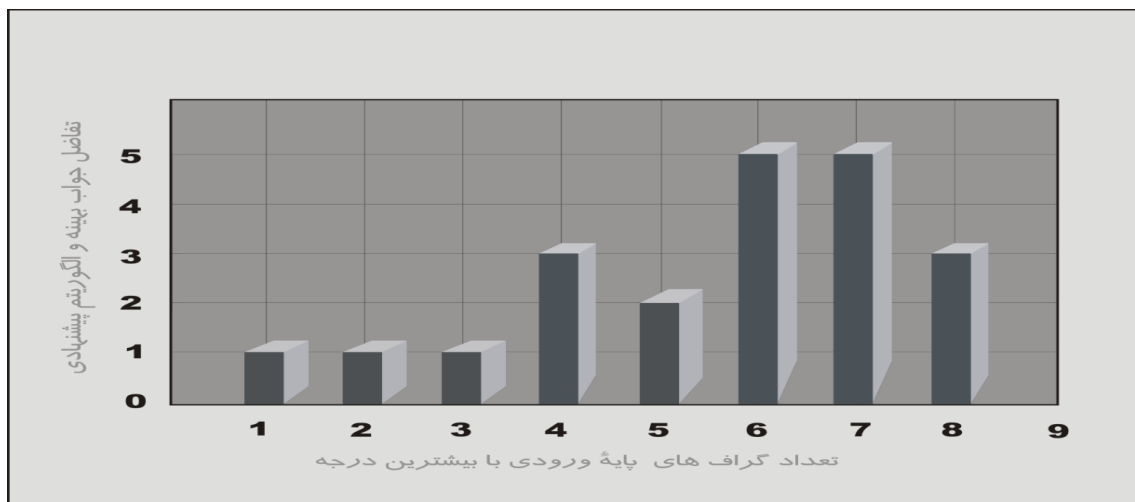
دومین مرحله، تجزیه ی درخت ورودی و گراف مرجع است که پیچیدگی محاسباتی این مرحله $O(N^2)$ است. مرحله ی سوم تکمیل ماتریس فاصله و ساخت گراف دو قسمتی وزن دار و انجام تطبیق دهی است که این مرحله دارای پیچیدگی $O(N^4)$ است. پس کل الگوریتم دارای پیچیدگی محاسباتی $O(N^2n+N^4)$ است.

۵- نتیجه ی آزمایش

در این بخش الگوریتم پیشنهاد شده با درخت های ورودی با تعداد گره های متفاوت و انواع چندضلعی آزمایش شده است. با توجه به این که در الگوریتم، زیر درخت ها با بیشترین درجه اولویت بالاتری دارد نتایج آزمایشات نشان می دهد که افزایش اندازه ی درخت احتمال به دست آوردن جواب بهینه را کاهش نمی دهد و در مواردی که تعداد زیر درخت ها با بیشترین درجه در درخت ورودی یکتا باشد الگوریتم بسیار مؤثر بوده و جواب های بهینه و یا نزدیک به بهینه ارائه می کند و با توجه به انتخاب تصادفی یک زیر درخت با بیشترین درجه اگر تعداد زیر درخت ها با بیشترین درجه یکتا نباشد احتمال یافتن جواب بهینه در اولین انتخاب کم می شود 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 می گیرد و سعی در انتخاب همسایه ها با کمترین 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، ۱۰ و ۱۱ آمده است.



شکل ۱۰: نتیجه ی اجرای الگوریتم بر روی درخت هایی که زیر درخت با بیشترین درجه ی آنها یکتا نیست.



شکل ۱۱: نمایش تعداد زیردرخت‌ها با بیشترین درجه در هر درخت.

- [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.

مراجع

- [1] Abellanas M, Garcia-Lopez J, Hernandez 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.