SID

سرویس های ویژه
سرویس ترجمه تخصصی
کارگاه های آموزشی
بلاگ مرکز اطلاعات علمی
عضویت در خبرنامه
فیلم های آموزشی

کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین
پروپوزال نویسی و پایان نامه نویسی

کارگاه آنلاین
روش تحقیق و مقاله نویسی علوم انسانی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و
ترفند های جستجو

انجمن کامپیوتر ایران
Computer Society of Iran
مرکز توسعه فناوری نیرو (متن)

درخت پوشای کمینه اقلیدسی روی نقاط متحرک

زاهد رحمتی
دانشکده علوم ریاضی، دانشگاه صنعتی شریف
rahmati@math.sharif.edu

علیرضا زارعی
دانشکده علوم ریاضی، دانشگاه صنعتی شریف
zarei@sharif.edu

چکیده: در این مقاله ما به ارائه‌ی یک داده ساختار جنبشی کارا و واکنشی برای نگه‌داری درخت پوشای کمینه اقلیدسی می‌پردازیم. این داده ساختار در واقع درخت پوشای کمینه اقلیدسی را روی نقاط متحرکی که مسیر حرکت هر کدام از نقاط یک تابع چند جمله‌ای با درجه ثابت بر حسب زمان است را نگه‌داری می‌کند. اندازه این داده ساختار $O(n^2)$ است و در زمان پیش‌پردازش‌ها $O(n^2 logn)$ ساخته می‌شود. کل تعداد رخدادهایی که در این داده ساختار پردازش می‌شوند $O(n^4)$ است، که هر رخداد در زمان $O(logn)$ پردازش می‌شود.

واژه های کلیدی: هندسه محاسباتی، درخت پوشای کمینه اقلیدسی، داده ساختار جنبشی.

۱- مقدمه

از جمله مسائل مهم هندسه محاسباتی، مسئله‌ی مجاورت هندسی(Geometric Proximity Problems) است. پیداکردن نزدیکترین جفت نقطه از مجموعه نقاط متحرک و همچنین داشتن درخت پوشای کمینه روی نقاط متحرک، از گونه‌های مسئله‌ی مجاورت هندسی روی نقاط متحرک هستند که کاربردهای فراوانی در خوشه-بندی و مطابقت الگو دارند. یکی از کاربردهای درخت پوشای کمینه روی نقاط متحرک، مسیریابی در شبکه های حسگر است. این شبکه‌ها متشکل از مجموعه‌ای از واحدهای متحرک و مستقل از هم هستند که از طریق فرستنده‌های رادیویی با یکدیگر در ارتباطند و اقدام به جمع آوری اطلاعات می‌کنند، بنابراین مسیریابی در این شبکه‌ها به گونه‌ای که کمترین انرژی را مصرف کنند بسیار حائز اهمیت است [1,2].

در گراف‌های همبند، بدون جهت و وزن‌دار، درختی که پوشا (یعنی شامل تمامی نودهای گراف) و کمترین وزن را در بین همه‌ی درخت-های پوشا داشته باشد، درخت پوشای کمینه (به اختصار MST) می-نامیم. اگر نقاط را روی سطح $\mathfrak{R}^2$ و در فضای اقلیدسی در نظر بگیریم و وزن هر یال فاصله اقلیدسی بین دو نقطه‌ی دو سر آن یال باشد، آنگاه درخت پوشای کمینه ساخته شده روی این چنین گرافی را درخت پوشای کمینه اقلیدسی (به اختصار EMST) می‌نامیم. درخت پوشای کمینه (اقلیدسی) توسط الگوریتم‌های مختلفی محاسبه می‌شود. الگوریتم کروسکال و پریم نمونه‌هایی از الگوریتم‌های پایه‌ای ساخت MST هستند که به ترتیب در زمان‌های $O(|E|log|E|)$ و $O(|V|^2)$ درخت پوشای کمینه را می‌سازند که V مجموعه نقاط و E مجموعه یال‌ها هستند [3,4,5]. سوالی که در اینجا مطرح می‌شود این است که: چگونه می‌توان درخت پوشای کمینه را روی نقاط متحرک در طول زمان نگه‌داری کرد؟

برای نگه‌داری یک مشخصه‌ی ویژه از نقاط متحرک (مانند پوش محدب یا درخت پوشای کمینه) الگوریتم و داده‌ساختارهایی نیاز است که این مشخصه‌ها را در طول زمان نگه‌داری کنند و تغییرات لازم را بصورت کارا اعمال نمایند، به مجموعه‌ی الگوریتم‌ها و داده‌ساختارهایی که این مشخصه‌ها را نگه‌داری کنند، داده ساختار جنبشی (Kinetic Data Structure به اختصار KDS) گفته می‌شود که نخستین بار در رساله‌ی دکترای باخ [6] مطرح شد. در داده ساختار جنبشی درستی پیکربندی کنونی با تعدادی *تاییدیه*(Certificate) مشخص می‌شود و زمانی که یک تاییدیه نامعتبر شود، به این معنی است که در آن زمان در محلی از داده ساختار یک *رخداد* (Event) اتفاق افتاده که درستی پیکربندی کنونی را از بین برده است. معمولا این تاییدیه‌ها در یک صف اولویت نگه‌داری می‌شوند که اولویت هر تاییدیه زمان نامعتبر شدن آن است که به آن *زمان خطا*(Failure Time) نیز گفته می‌شود. در این صف که به آن صف رخداد می‌گوییم، عملیات درج؛ حذف و یافتن کمترین مقدار در زمان لگاریتمی انجام می‌شود. با گذر زمان و رسیدن به نزدیکترین زمان خطا (کوچکترین عضو صف رخداد)، برخی از تاییدیه‌ها نامعتبر می‌شوند که باید از صف رخداد حذف شوند سپس در صورت لزوم با فراخوانی مکانیزم‌های خاصی، داده ساختار جنبشی اصلاح شده و تاییدیه‌های نامعتبر با نوع معتبر آن جایگزین می‌شوند.

Archive of SID

www.SID.ir

ارزیابی عملکرد یک داده ساختار جنبشی بر مبنای چهار معیار زیر است و هر چه یک داده ساختار جنبشی نسبت به این معیارها عملکرد بهتری داشته باشد، داده ساختار جنبشی بهتری است [6,7].

۱) *واکنشی بودن* (Responsive): با نامعتبر شدن یک تاییدیه، مدت زمان لازم جهت به‌روزکردن داده ساختار جنبشی را *زمان‌پاسخ* (Response Time) می‌نامیم. اگر زمان پاسخ حداکثر ضریب لگاریتمی با توان ثابت از اندازه‌ی ورودی (نقاط متحرک) باشد، داده ساختار جنبشی را واکنشی می‌گوییم.

۲) *فشرده بودن (تراکم،* Compact) : در صورتی که تعداد تاییدیه‌های فعال در هر لحظه از زمان یا به عبارت دیگر فضای مورد استفاده توسط داده ساختار جنبشی نسبت به اشیاء ورودی کمتر باشد، داده ساختار جنبشی فشرده‌تری داریم. اگر این تعداد حداکثر به صورت ضریب لگاریتمی با توان ثابت از اندازه ورودی بزرگتر باشد، داده ساختار جنبشی را فشرده (متراکم) نامیم.

۳) *محلی بودن* (Local) : اگر بیشترین تعداد رخدادهایی که در هر بازه زمانی ثابت به یک شئ خاص وابسته‌اند، حداکثر ضریب لگاریتمی با توان ثابت از اندازه‌ی ورودی باشد، داده ساختار جنبشی محلی است.

۴) *کارایی* (Efficient) : یکی از کلیدی‌ترین معیارهای ارزیابی عملکرد یک داده ساختار جنبشی، معیار کارایی است و به تعداد رخدادها وابستگی دارد. بررسی این معیار نیازمند متمایز کردن دو نوع رخداد است: بعضی از رخدادها باعث نامعتبر شدن تاییدیه‌ها می‌شوند ولی تنها تغییرات جزئی را در داده ساختار جنبشی ایجاد می‌کنند. و مشخصه‌ی ویژه‌ی نقاط که خروجی مسئله است را تغییر نمی‌دهند این رخدادها را *رخداد داخلی گوییم*. رخدادهایی که باعث تغییر در خروجی می‌شوند یا به بیانی دیگر آن دسته از رخدادها که باعث نامعتبر شدن مشخصه‌ی ویژه می‌شوند، *رخداد خارجی* هستند. هر چه نسبت رخدادهای داخلی به رخدادهای خارجی کمتر باشد، داده ساختار جنبشی کاراتری خواهیم داشت.

اگر داده ساختار جنبشی علاوه بر حرکت، درج و حذف اشیاء را به صورت کارایی پشتیبانی کند، آن را یک داده ساختار جنبشی و پویا می‌نامیم.

فرض کنید که $P = \{p_1, p_2, \dots, p_n\}$ شامل مجموعه نقاطی باشد که هر نقطه دارای یک تابع مسیر حرکت مستقل از دیگر نقاط، در فضای $\mathfrak{R}^2$ است و تابع مسیر هر $p_i(t)$ با $p_i(t) = (x_i(t), y_i(t))$ نشان داده می‌شود که $x_i(t)$ و $y_i(t)$ توابع چندجمله‌ای با درجه ثابت هستند. هدف ما ارائه‌ی داده ساختار جنبشی مناسبی برای یافتن درخت پوشای کمینه اقلیدسی روی این نقاط است.

آقای اگروال و همکارانش [8] توانستد درخت پوشای کمینه جنبشی را برای گرافی که طول یال‌های آن تابع خطی بر حسب زمان است را بیابند، ولی این روش برای حالت کلی مسئله که در آن طول یال‌ها تابع غیر خطی است را شامل نمی‌شود. یکی دیگر از مسائلی که ارتباط نزدیکی به مسئله ما دارد، پیدا کردن نزدیکترین همسایگی هر نقطه از یک مجموعه نقاط روی نقاط متحرک است که این نقاط هر کدام دارای مسیر حرکت $p_i(t)$ هستند. آقای اگروال و همکارانش [9] یک داده ساختار جنبشی را برای این مسئله ارائه داده‌اند که معیار کارایی، واکنشی و فشردگی را دارد اما این داده ساختار معیار محلی بودن را مرتفع نمی‌کند.

با توجه به نداشتن داده ساختار جنبشی کارا برای درخت پوشای کمینه اقلیدسی، برخی تلاشها برای یافتن مقدار تقریبی EMST صورت گرفته است. آقای باخ و همکارانش [10] یک داده ساختار جنبشی را برای $(1+\epsilon) - \mathrm{EMST}$ روی نقاط متحرک در $\mathfrak{R}^2$ ارائه نمودند. تعداد رخدادها در این داده ساختار $O(\epsilon^{-(d-1)}n^3)$ است و برای هر رخداد می‌توان عمل به روز رسانی را در زمان $O(\log^d n)$ انجام داد و همچنین فضای مورد استفاده‌ی داده ساختار ارائه شده $O(\epsilon^{-(d-1)/2}\log^{d-1} n)$ است. بنابراین علاوه بر تقریبی بودن روش باخ، تنها معیارهای واکنشی، محلی و فشردگی را مرتفع می‌کند و این روش کارا نیست.

در این مقاله ما داده ساختار جنبشی ارائه می‌نماییم که علاوه بر معیار کارایی، معیار واکنشی را نیز داراست که در داده ساختارهای جنبشی بسیار حائز اهمیت است. در داده ساختار پیشنهای با زمان پیش‌پردازش $O(n^2 logn)$ داده ساختاری به اندازه‌ی $O(n^2)$ می‌سازیم که با استفاده از آن هر رخداد را در زمان $O(logn)$ پردازش می‌کنیم. نشان می‌دهیم کل تعداد رخدادهایی که در این روش پردازش می‌شوند با توجه به مسیر حرکت نقاط $O(n^4)$ است.

در بخش دوم این مقاله داده ساختار جنبشی پیشنهادی را ارائه می‌کنیم، سپس، رخدادهایی که باید پردازش شوند بررسی و در انتها تحلیل الگوریتم و نتیجه‌گیری را بیان می‌کنیم.

۲- داده ساختار جنبشی برای EMST

فاصله اقلیدسی بین دو نقطه‌ی P_i, P_j $(i \neq j)$ را با $e_{i,j}(t)$ نشان می‌دهیم. با توجه به تابع مسیر حرکت هر نقطه‌ی P_i که به صورت $p_i(t) = (x_i(t), y_i(t))$ است، $e_{i,j}(t)$ معادل است با تفاضل $p_i(t)$ و $p_j(t)$، که طول $e_{i,j}(t)$ به صورت زیر محاسبه می‌شود:

$$\left\|e_{i,j}(t)\right\|_2 = \sqrt{\left(X_i(t) - X_j(t)\right)^2 + \left(Y_i(t) - Y_j(t)\right)^2} \qquad (۱)$$

از این به بعد برای سادگی کار $e_{i,j}(t)$ها را با $e_k(t)$ نشان می‌دهیم که دامنه‌ی k از ۱ تا $O(n^2)$ تغییر می‌کند و دلیل آن هم

Archive of SID

اینست که به ازای i و jهای مختلف، تعدادکل یال‌ها از مرتبه‌ی $O(n^2)$ است. همچنین یال‌هایی که عضو EMST نیستند را *یال‌های مجازی* می‌نامیم.

یک EMST تا زمانی معتبر باقی می‌ماند که مجموع وزن یال‌های آن، کمترین وزن در بین همه‌ی درخت‌های پوشا داشته باشد. حال اگر در طول زمان وزن یکی از یال‌های EMST کمتر از وزن یکی از یال‌های مجازی شود، این یال مجازی می‌تواند کاندیدی برای تعویض با یال مفروض EMST باشد و این اتفاق زمانی رخ می‌دهد که افزودن یال مجازی در EMST و حذف یال مورد نظر از EMST باعث ناهمبند شدن EMST نشود. لم زیر شرایطی که مجموعه‌ی یال‌های EMST تغییر می‌کنند را بیان می‌کند(از این به بعد $\|X\|$ را معادل $\|X\|_2$ در نظر می‌گیریم).

لم۱: شرایط *لازم* و *کافی* تغییر در EMST به صورت زیر است:

۱- $e_i(t) \in EMST$

۲- $e_j(t) \notin EMST$

۳- $\|e_j(t)\| \prec \|e_i(t)\|$

۴- $e_i(t) \in \varsigma(e_j(t))$، که $\varsigma(e_j(t))$ شامل مجموعه‌ای از یال‌هاست که در مسیر بین دو سر یال $e_j(t)$ در EMST واقع هستند.

اثبات: با توجه به شرایط ۱ تا ۳ واضح است که مجموع طول یال‌های EMST در صورت حذف $e_i(t)$ و درج $e_j(t)$ کمتر از حالت قبلی است. باید نشان دهیم این جایگزینی باعث حفظ EMST می‌شود: با افزودن یال $e_j(t)$ دقیقا یک دور در EMST ایجاد می‌شود که طبق شرط۴ با حذف $e_i(t)$ این دور از بین می‌رود. در نتیجه ۴ شرط بالا شروط کافی برای تغییر EMST هستند.

اگر مجموعه یال‌های EMST تغییر کنند حتما یکی از یال‌های مجازی جایگزین یکی از یال‌های EMST شده است (شروط ۱و۲) و همچنین این یال مجازی طول کمتری نسبت به یال EMST داشته است (شرط۳). شرط ۴ام را با برهان خلف اثبات می‌کنیم: اگر فرض کنیم $e_j(t) \notin \varsigma(e_i(t))$ آنگاه حذف $e_i(t)$ و درج $e_j(t)$ علاوه بر اینکه در درخت دور ایجاد می‌کند، درخت را به دو بخش جدا از هم تقسیم می‌کند (تناقض). درنتیجه تغییر EMST لازمه‌ی برقراری شرط۴ است. ◊

داده ساختار جنبشی که در اینجا ارائه می‌کنیم شامل داده ساختارهایی است که در طول زمان به‌روزرسانی‌های لازم در آنها صورت می‌گیرد و این به‌روزرسانی‌ها همزمان با نامعتبر شدن تاییدیه‌ها اتفاق می‌افتند.

تاییدیه‌ها در KDS پیشنهادی بر مبنای طول یال‌ها و نتایج حاصل از لم۱ استوار هستند و هر تاییدیه باید درستی ترتیب طول این یال‌ها را ضمانت کند. در این داده ساختار لیست یال‌های مجازی و یال‌های EMST به صورت مرتب شده بر حسب طول نگه‌داری می‌شوند. این لیست را با L_e نشان می‌دهیم و باید ترتیب یال‌ها را در طول زمان حفظ کند و ساختن اولیه‌ی آن در پیش‌پردازش انجام می‌شود.

لم۲: L_e را می‌توان در زمان پیش‌پردازش $O(n^2 \log n)$ و با حافظه‌ی $O(n^2)$ ساخت، بطوریکه زمان جابجا کردن دو عضو آن از مرتبه‌ی $O(1)$ باشد.

اثبات: با استفاده از الگوریتم‌های پایه‌ی مرتب کردن، می‌توان L_e را با زمان $O(n^2 \log n)$ و فضای $O(n^2)$ ساخت و نیز با توجه به متوالی بودن دو یال که قرار است جابجا شوند، جابجای دو عضو در آن از مرتبه زمان ثابت خواهد بود. ◊

داده ساختار بعدی برای متمایز کردن یال‌های EMST از یال‌های مجازی تعریف می‌شود که لیست یال‌های EMST را به صورت مرتب در طول زمان نگه‌داری می‌کند و این لیست را با L_{EMST} نشان می‌دهیم. ساختن L_{EMST} در پیش‌پردازش صورت می‌گیرد و هر یال از این لیست به یال متناظر در لیست L_e اشاره می‌کند و برعکس.

لم۳: L_{EMST} در زمان پیش‌پردازش $O(n \log n)$ و حافظه‌ی $O(n)$ ساخته می‌شود و درج و حذف یک یال در آن زمان ثابت اجرا می‌شود.

اثبات: زمان پیش‌پردازش و حافظه‌ی مصرفی براحتی از الگوریتم مرتب‌سازی قابل محاسبه است. از طرفی از لم۱ نیز می‌توان نتیجه گرفت که با توجه به جابجایی بین دو یال متوالی $e_i(t)$ و $e_j(t)$ از لیست L_e و اینکه $e_i(t)$ به یال متناظر در L_{EMST} اشاره می‌کند، یال $e_j(t)$ در واقع از نظر مکانی در لیست L_{EMST} حتما در همان مکان $e_i(t)$ قرار خواهد گرفت. پس در نهایت عمل درج و حذف یال در L_{EMST} در زمان ثابت انجام می‌شود. ◊

همانطور که گفته شد در KDS نیاز داریم که تاییدیه مربوط به نزدیکترین زمان خطا (زمان نامعتبر شدن یک تاییدیه) را مشخص کنیم. لیست L_e حاوی یال‌های مرتب متوالی است و باید به ازای همه‌ی تاییدیه‌های مرتبط با این یال‌های متوالی، نزدیکترین زمان خطایی که ترتیب فاصله‌ی بین دو یال متوالی را عوض می‌کند، پیدا کنیم. به عنوان مثال اگر در لیست L_e داشته باشیم:

$$\|e_i(t)\| \le \|e_j(t)\| \quad , i = j-1 (j = 1, \ldots, O(n^2))$$

آنگاه زمان خطای تاییدیه‌ی مرتبط با دو یال متوالی $e_i(t)$ و $e_j(t)$ کمترین زمانی است که از حل نامعادله‌ی $e_i(t) - e_j(t) \ge 0$ بدست می‌آید. بنابراین سومین زیر داده ساختار جنبشی باید علاوه بر اینکه کمترین زمان خطا در بین همه‌ی تاییدیه‌ها را به ما بدهد، درج و حذف تاییدیه و جستجوی یال در آن در زمان مناسب انجام شود. ساختار مناسب برای داشتن موارد فوق، استفاده از یک صف اولویت است که هر عضو آن شامل یک تاییدیه و دو یال مربوط به این تاییدیه با کلید زمان خطاست.

برای داشتن زمان مناسب جستجوی یال در این صف اولویت که با Q نشان می‌دهیم، هر یال $e_i(t)$ در لیست L_e (که در دو تاییدیه

Archive of SID

مشارکت دارد) به دو عضو از Q (که شامل یال متناظر با $e_i(t)$ هستند) اشاره می‌کند و برعکس برای هر عضو Q (که دارای دو یال است) دو اشاره‌گر به دو عضو از لیست L_e (به دو یال متناظر با یال‌های آن عضو Q) داریم.

لم۴: صف اولویت Q در زمان پیش‌پردازش $O(n^2 \log n)$ و حافظه‌ی $O(n^2)$ قابل ساخت است. هر درج و حذف در آن در زمان $O(\log n)$ و جستجوی یک یال از لیست L_e در Q و همچنین یافتن نزدیکترین زمان خطا از مرتبه‌ی $O(1)$ است.

اثبات: با استفاده از روش‌های موجود برای صف اولویت، عمق Q از مرتبه‌ی $O(\log n)$ است و درج و حذف یک عضو در Q به زمان $O(\log n)$ جهت به‌روزکردن Q نیاز دارد. بنابراین زمان ساختن Q برای تعداد $O(n^2)$ عضو (تاییدیه) از مرتبه‌ی $O(n^2 \log n)$ است و حافظه مصرفی به تعداد تاییدیه‌هاست. اولین عضو (ریشه) Q کوچکترین عضو آن یعنی نزدیکترین زمان خطاست که در زمان ثابت قابل یافتن است. از طرفی، با توجه به اینکه هر یال از لیست L_e به اعضای Q اشاره می‌کنند در نتیجه زمان جستجوی یک یال از لیست L_e در Q، $O(1)$ است. ◊

برای بررسی شرط۴ از لم۱ به داده ساختاری نیاز است که برای دو یال دلخواه $e_i(t)$ و $e_j(t)$ و در زمان مناسب رخدادن یا رخندادن این شرط را نشان دهد. در این شرط باید بررسی شود که آیا مسیر بین دو نقطه‌ی دو سر یال $e_j(t)$ در EMST شامل یال $e_i(t)$ هست یا نه. با حذف $e_i(t)$ از EMST مجموعه نقاط به دو زیرمجموعه‌ی جدا از هم $S_{i,1}$ و $S_{i,2}$ افراز می‌شوند، می‌توان با جستجو کردن نقاط دو سر $e_j(t)$ در این دو مجموعه به سوال مورد نظر پاسخ داد. یعنی اگر یکی از نقاط دو سر $e_j(t)$ در مجموعه‌ی $S_{i,1}$ و نقطه‌ی دیگر در مجموعه‌ی $S_{i,2}$ باشد آنگاه شرط۴ از لم۱ برقرار است.

برای این کار، به ازای هر یال $e_i(t)$ از EMST درخت جستجوی دودویی متوازنی نگه‌داری می‌کنیم که شامل اعضای مجموعه‌ی $S_{i,1}$ و $S_{i,2}$ هستند و با $BBST_{i,1}$ و $BBST_{i,2}$ نشان می‌دهیم. در این صورت بررسی شرط۴ از لم۱ در زمان $O(\log n)$ امکانپذیر است.

لم۵: ساختن داده ساختارهای $BBST_{i,1}$ و $BBST_{i,2}$ برای همه‌ی یال‌های e_i از EMST با زمان پیش‌پردازش $O(n^2 \log n)$ و حافظه‌ی مصرفی $O(n^2)$ انجام پذیر است. با داشتن این داده ساختارها بررسی شرط۴ از لم۱ در مرتبه‌ی زمانی $O(\log n)$ امکانپذیر است.

اثبات: با توجه به اینکه تعداد یال‌های EMST، (n-1)تاست و مرتبه‌ی زمانی و حافظه مصرفی در ساختن هر BBST به ترتیب $O(n \log n)$ و $O(n)$ است در نتیجه کل زمان پیش‌پردازش $O(n^2 \log n)$ و حافظه مصرفی $O(n^2)$ است. از طرفی زمان جستجو در هر BBST، $O(\log n)$است و برای بررسی شرط۴ از لم۱ به دو جستجو در این درخت‌ها نیاز داریم. ◊

با ترکیب نتایج لم‌های ۱ تا ۵، نتیجه‌ی زیر در مورد داده ساختار جنبشی پیشنهادی برای نگه‌داری EMST برقرار است:

قضیه۱: KDS لازم برای نگه‌داری EMST روی نقاط متحرک در زمان پیش‌پردازش $O(n^2 \log n)$ قابل ساخت و حافظه مصرفی آن از مرتبه‌ی $O(n^2)$ است.

۳- پردازش رخدادها

با توجه به داده ساختارهای مطرح شده برای KDS در بخش قبل، اگر x عضوی از Q باشد که زمان نامعتبر شدن تاییدیه در آن نزدیکترین زمان خطا است (یعنی x اولین عضو Q است) و x شامل $e_k(t)$ و $e_j(t)$ باشد که ترتیب یال‌ها بر حسب طول در لیست L_e به صورت زیر است:

$$^{\min}\left[\ldots, e_i(t), e_j(t), e_k(t), e_l(t), \ldots\right]^{\max}$$

آنگاه اگر کوچکترین عضو Q یعنی زمان خطای تاییدیه‌ی x با زمان فعلی برابر شد (که معادل جابجایی ترتیب دو یال متوالی در لیست مرتب شده‌ی بالا است) باید مراحل زیر را جهت جایگزین کردن تاییدیه نامعتبر با نوع معتبر، اجرا و در صورت لزوم تغییرات لازم در EMST اعمال شود:

۱) دو یال $e_k(t)$ و $e_j(t)$ را در زمان ثابت در لیست L_e جابجا می‌کنیم.

۲) با توجه به نامعتبر شدن تاییدیه مربوط به عضو x، x باید از Q حذف شود و عضو جدید y را به این صورت می‌سازیم که یال‌های آن همان یال‌های x و کلید آن اولین t بعد از زمان فعلی است که در نامعادله‌ی $e_j(t) - e_k(t) \leq 0$ صدق می‌کند. با توجه به ساختار Q، حذف x و درج y در زمان $O(\log n)$ قابل اجراست.

۳) جابجا کردن دو یال $e_k(t)$ و $e_j(t)$، تاییدیه‌های مربوط به جفت یال‌های $e_i(t)$ ، $e_j(t)$ و همچنین $e_k(t)$ ، $e_l(t)$ را نیز نامعتبر می‌کند، بدین معنی که این زوج یال‌ها دیگر در L_e مجاور هم نیستند. بنابراین باید رخدادهای مربوط به این دو زوج یال از Q حذف شوند و بجای آنها رخدادهای مربوط به زوج یال‌های $e_i(t)$ ، $e_k(t)$ و $e_j(t)$ ، $e_l(t)$ در Q درج شوند.

۴) شروط لم۱ را روی دو یال جابجا شده‌ی $e_j(t)$ و $e_k(t)$ بررسی می‌کنیم و در صورت برقراری همه‌ی شرایط که در زمان $O(\log n)$ قابل تشخیص است، باید *ابتدا* $e_j(t)$ از EMST حذف و یال $e_k(t)$ درج گردد و درخت‌های $BBST_{j,1}$ و $BBST_{j,2}$ حذف شوند و درخت‌های

Archive of SID

$BBST_{k,1}$ و $BBST_{k,2}$ ایجاد شوند. نکته‌ی مهمی که در اینجا وجود دارد این است که $S_{j,1}$ و $S_{j,2}$ با $S_{k,1}$ و $S_{k,2}$ برابر هستند و نتیجه‌ی آن این است که نیازی به ساختن $BBST_{k,1}$ و $BBST_{k,2}$ نیست و از $BBST_{j,1}$ و $BBST_{j,2}$ می‌توان استفاده کرد. بنابراین کافی است بجای $e_j(t)$ در L_{EMST} مقدار $e_k(t)$ قرار گیرد. *در نهایت*، باید یال $e_k(t)$ از لیست L_e به مکان قبلی $e_j(t)$ در لیست L_{EMST} (که اکنون $e_k(t)$ در آنجا قرار دارد) اشاره نماید. و از اینکه دیگر $e_k(t) \notin L_{EMST}$ باید اشاره‌گر آن به لیست L_{EMST} حذف شود، با توجه به ساختار KDS ارائه شده این به‌روزرسانی‌ها در زمان ثابت قابل اجرا هستند.

لم۶: زمان پردازش هر رخداد در KDS از مرتبه‌ی $O(\log n)$ است.

اثبات: با توجه به موارد چهارگانه بیان شده در این بخش مشخص است که زمان پردازش هر مورد و در نتیجه هر رخداد $O(\log n)$ است. ◊

۴- تحلیل عملکرد

در این بخش به بررسی کیفیت داده ساختار جنبشی پیشنهادی بر اساس معیارهای ارزیابی عملکرد KDS می‌پردازیم.

هر یال در لیست L_e می‌تواند جابجایی‌های با سایر یال‌ها در این لیست داشته باشد، حال سوالی که در اینجا مطرح می‌شود این است که تعداد کل این جابجایی‌ها چقدر است؟ واضح است که پاسخ دادن به این سوال معادل پاسخ دادن به تعداد رخدادهاست. مطابق با چهار موردی که در بخش۳ به آنها پرداختیم، هر جابجایی بین یال‌ها منجر به اعمال حداقل سه مورد اول از این چهار مورد خواهد شد. در صورتی که تعداد کل رخدادهایی که فقط تغییرات سه مورد اول لازمه‌ی آنهاست را بشماریم، تعداد رخدادهای داخلی محاسبه می‌شود.

با توجه به اینکه تابع طول هر یال $e_i(t)$ (یعنی $\|e_i(t)\|$) یک چند جمله‌ای با حداکثر درجه ثابت است و جابجایی دو یال $e_i(t)$ و $e_j(t)$ در لیست L_e متناظر با رابطه‌ی $e_i(t) - e_j(t) = 0$ است، پس تعداد جابجایی‌های دو یال مشخص $e_i(t)$ و $e_j(t)$ در این لیست از مرتبه‌ی $O(1)$ است. بنابراین کل تعداد جابجایی‌ها و در نتیجه تعداد کل رخدادهای داخلی $O(n^4)$ خواهد بود.

رخدادهایی که علاوه بر سه مورد اول مورد چهارم بخش قبل در فرایند پردازش آنها اجرا شود (یعنی رخدادهایی که باعث تغییر EMST می‌شوند) را رخدادهای خارجی می‌نامیم. بدیهی است که تعداد رخدادهای خارجی از مرتبه‌ی $O(n^4)$ است زیرا حداکثر به اندازه‌ی تعداد رخدادهای داخلی خواهد بود. با مثال زیر نشان خواهیم داد که تعداد این رخدادها نیز $\Omega(n^2)$ است: n نقطه را در نظر بگیرید که روی یک خط قرار دارند و ترتیب قرار گرفتن آنها از نقطه‌ی $P_1(t)$ تا $P_n(t)$ است و پس از حرکت ترتیب قرارگیری آنها روی همین خط از نقطه‌ی $P_n(t)$ تا $P_1(t)$ خواهد بود. در واقع در این تعریف نقطه‌ی $P_1(t)$ از تمام نقاط $P_2(t)$ تا $P_n(t)$ عبور کرده است. واضح است که EMST همواره به صورت یک پاره‌خط از کوچکترین نقطه تا بزرگترین نقطه است و بر اساس شکل حرکت نقاط هر $e_{i,j}(t)$ در مقطعی از زمان در EMST ظاهر شده است (زمانی که $P_i(t)$ و $P_j(t)$ روی خط مربوطه مجاور هم بوده‌اند). بنابراین تعداد رخدادهای خارجی $\Omega(n^2)$ است. قضیه زیر معیارهای ارزیابی عملکرد KDS را بیان می‌کند:

قضیه۲: KDSی که برای نگه‌داری EMST مطرح شد، واکنشی است و معیار کارایی را از مرتبه‌ی $O(n^2)$ دارد ولی معیار محلی بودن و فشرده بودن را ندارد.

اثبات: از لم۶ نتیجه می‌شود که زمان پردازش هر رخدادها از مرتبه $O(\log n)$ است، پس KDS واکنشی است. با توجه به تعداد رخدادهای داخلی و خارجی، نسبت رخدادهای داخلی به خارجی (یعنی همان مرتبه‌ی معیار کارایی) از $O(n^2)$ است.

محلی نبون KDS نیز واضح است چرا که هر نقطه می‌تواند با $O(n)$ یال در تماس باشد که تعداد رخدادهای این یالها از مرتبه‌ی لگاریتمی بیشتر خواهد شد، پس KDS معیار محلی بودن را نیز ندارد. متراکم نبودن KDS نیز براحتی از قضیه۱ بدست می‌آید. ◊

۵- نتیجه گیری

در این مقاله یک داده ساختار جنبشی را برای درخت پوشای کمینه (اقلیدسی) ارائه نمودیم. با توجه اهمیت ارزیابی عملکرد داده ساختارهای جنبشی و داشتن معیارهای اساسی در این داده ساختارها، KDSی که برای EMST پیشنهاد نمودیم دارای معیارهای کارایی و واکنشی است. در این ساختار معیارهای محلی بودن و فشردگی نسبت به تعداد نقاط مرتفع نشده است.

در گام‌های بعدی تلاش برای اضافه نمودن معیارهای محلی بودن و فشردگی نسبت به تعداد نقاط باید در نظر گرفته شود. بنابراین یافتن داده ساختار جنبشی که چهار معیار ارزیابی عملکرد KDS (واکنشی، کارایی، محلی بودن و فشردگی) را داشته باشد، می‌تواند به عنوان کارهای بعدی صورت گیرد.

مراجع

[1] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, pp. 164 - 194, 2005.

[2] Xiang-Yang Li, Wen-Zhan Song, Weizhao Wang, "A unified energy-efficient topology for unicast and broadcast," *International Conference on Mobile Computing and Networking*, pp. 1 - 15, 2005.

[3] J. B. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proceedings of the American Mathematical Society*,

Archive of SID

www.SID.ir

Algorithms, no. 1,2008.

[10] J. Basch, L. J. Guibas, and L. Zhang., "Proximity Problems on moving points," *Annual Symposium on Computational Geometry*, vol. 13, pp. 344-351, 1997.

vol. 7, p. 48–50, 1956.

[4] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, p. 1389–1401, 1957.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd ed. North America: MIT Press, 2009.

[6] J. Basch, *Kinetic data structures*.: PhD Thesis, Stanford University, 1999.

[7] J. Basch, L. J. Guibas and J. Hershberge, "Data structures for mobile data," *Journal of Algorithms*, vol. 31, no. 1, pp. 1-28, 1999.

[8] P.K.Agarwal, D.Eppstein, L.J.Guibas and M.R.Henzinger, "Parametric and kinetic minimum spanning," *39th IEEE Sympos Found Comput Sci*, pp. 596-605, 1998.

[9] P. K. Agarwal, Haim Kapalan and Micha Sharir, "Kinetic and dynamic data structures for closest pair and all nearest neighbors," *ACM Transactions on*

Archive of SID

www.SID.ir

SID

فیلم های آموزشی

عضویت در خبرنامه

بلاگ مرکز اطلاعات علمی

کارگاه های آموزشی

سرویس ترجمه تخصصی

سرویس های ویژه

کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترفند های جستجو

کارگاه آنلاین روش تحقیق و مقاله نویسی علوم انسانی

کارگاه آنلاین پروپوزال نویسی و پایان نامه نویسی