کاربرد داده ساختارهای جنبشی در مسیریابی شبکه‌های حسگر متحرک

کامیار رفعتی[*]، نعیم اصفهانی[†]، محمد قدسی[‡§]

چکیده

یکی از موضوعات مطرح در طراحی الگوریتم‌ها بحث شبکه‌های حسگر می‌باشد. این شبکه‌ها متشکل از مجموعه‌ای از واحدهای متحرک و مستقل از هم با توان مصرفی و پردازشی محدود است که از طریق فرستنده‌های رادیویی با یکدیگر در ارتباطند و اقدام به جمع‌آوری اطلاعات می‌نمایند. مساله‌ی مسیریابی در این شبکه‌ها به گونه‌ای که حداقل انرژی مصرف شود، از دسته مسائل غیر چند جمله‌ای سخت می‌باشد که ارائه راه حل‌های تقریبی مناسب موضوع برخی از تحقیقات در این زمینه است. در بیشتر مدل‌های ارائه شده فرض بر ثابت بودن حسگرها است؛ در این مقاله سعی می‌شود الگوریتمی برای مسیریابی در شبکه‌ی حسگرهای متحرک ارائه شود. با توجه به ماهیت جنبشی این شبکه‌ها ، استفاده از داده ساختارهایی که بتواند ساختار زیر درخت فراگیر را به صورت بهینه نگاهداری نمایند بسیار سودمند است. در این تحقیق از داده ساختار جنبشی برای نگاهداری زیر درخت فراگیر استفاده شده است. در این مقاله این روش ارایه و بررسی می‌شود و نشان می‌دهیم که باعث کاهش پیچیدگی محاسباتی مسیریابی در این شبکه‌ها می‌شود.

کلمات کلیدی

الگوریتم، شبکه‌های حسگر، مسیریابی، داده ساختارهای جنبشی، کوچکترین زیر درخت فراگیر محلی

Kinetic Data Structures for Routing Problem in Mobile Sensor Networks

Kamyar Rafati, Naeem Esfahani, Mohammad Ghodsi

Abstract

"Sensor networks" is an important topic in computer science and algorithm design. These networks are constructed from a set of independent mobile units with limited power and process capability. These units communicate and gather information using radio transmitters. The problem of routing in these networks with minimum power consumption is a NP-hard problem. Therefore, many researches use approximation algorithms for this problem. Most of the proposed models work with fixed sensors. In this paper, we propose an algorithm for routing in mobile sensor networks. According to the inherent kinetic structure of such networks, the use of a kinetic data structure which efficiently maintains minimum spanning tree (MST) is useful. In this paper, we present such structure for our problem and show that this method reduces the time complexity of routing in sensor networks.

Keywords

Algorithm, Sensor Networks, Routing, Kinetic Data Structures, Minimum Spanning Trees

[*] دانشجوی کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه صنعتی شریف، دانشکده‌ی کامپیوتر، rafati@ce.sharif.edu

[†] دانشجوی کارشناسی ارشد مهندسی نرم‌افزار، دانشگاه صنعتی شریف، دانشکده‌ی کامپیوتر، esfahani@ce.sharif.edu

[‡] (استاد، دانشگاه صنعتی شریف، دانشکده‌ی کامپیوتر)،(پژوهشکده کامپیوتر، مرکز تحقیقات فیزیک نظری)، ghodsi@sharif.edu

[§] فعالیت این نویسنده توسط پژوهشکده کامپیوتر مرکز تحقیقات فیزیک نظری حمایت می‌شود (contract: CS1385-2-01)

۱- مقدمه

با ظهور ارتباطات بی‌سیم بین عناصر مختلف و به دنبال آن مسئله شبکه‌های بی سیم و متحرک، توجه بسیاری از اندیشمندان رشته علوم کامپیوتر به مسائل موجود در این شبکه از قبیل مسیریابی معطوف شد. اما این شبکه‌ها پاسخگوی تمام نیازها در زمینه ارتباطات بی سیم نبودند. به همین منظور مدل شبکه‌های ویژه[1] ارائه شد که در آنها ارتباطات از طریق فرستنده‌ها و گیرنده‌های رادیویی با فاصله ارتباطی محدود انجام می‌گرفت و در ضمن ساختار یکپارچه مرکزی برای مسیریابی و مدیریت ندارند. در قدم بعدی محدودیت توان مصرفی و عملیاتی نیز به مدل فوق افزوده شد و مدل شبکه حسگر معرفی شد.

شبکه های حسگر کاربرد بسیار وسیعی دارند. مثلا حسگرهای تشخیص آتش سوزی در یک جنگل و یا شهر همچنین حسگرهای تشخیص تشعشعات هسته‌ای در یک رآکتور هسته‌ای، نمونه‌هایی از این کاربردها هستند.

ویژگی‌های شبکه‌های حسگر را می‌توان به اجمال به این موارد تقسیم نمود: ۱. انرژی محدود عناصر .۲. پهنای باند محدود .۳. شبکه بدون ساختار و متغیر با زمان .۴. کیفیت پایین ارتباطات .۵. قدرت محاسبات محدود در عناصر.

از جمله مسائل مطرح در زمینه شبکه‌های حسگر، بحث مسیریابی در این شبکه‌ها است. الگوریتم‌های متفاوتی برای این مسئله ارائه شده است. الگوریتم‌های ارائه شده را می‌توان به دو دسته همگن و ناهمگن تقسیم نمود. الگوریتم‌های همگن فرض را بر یکسان بودن عناصر شبکه (از نظر برد فرستنده) می‌گذارند. الگوریتم‌های ناهمگن از انعطاف‌پذیری بیشتری برخوردار هستند. الگوریتم‌های ناهمگن با توجه به اطلاعاتی استفاده می‌کنند به سه دسته تقسیم می‌شوند. ۱- بر مبنای محل : در آنها محل دقیق عناصر مشخص می‌باشد. ۲- بر مبنای جهت : در آنها فرض می‌شود که هر کس جهت نسبی همسایگانش را نسبت به خود می‌داند. ۳- بر مبنای همسایه : در آنها فرض می‌شود که شناسه همسایه‌ها در اختیار است.

الگوریتم‌های ارائه شده را از یک منظر دیگر می‌توان به دو دسته متمرکز و نامتمرکز نیز تقسیم نمود. در الگوریتم‌های متمرکز، یک ناظر خارجی در سیستم وجود دارد که مسئولیت مسیریابی را به عهده دارد. البته فرض وجود چنین ناظری اولا با ماهیت شبکه‌های حسگر سازگار نیست در ضمن قابلیت مقیاس‌پذیری ندارد.

از جمله روش‌های رایج در زمینه مسیریابی استفاده از درخت فراگیر کمینه است. اما به دو دلیل که در ادامه خواهیم دید، استفاده از آنها در این شبکه‌ها محبوبیت پیدا نکرده است. اولا پیدا کردن کوچکترین درخت فراگیر یک الگوریتم ماهیتا متمرکز است و دوما به علت آنکه هزینه ساخت آن بالاست و در این شبکه‌ها - به علت متحرک بودن عناصر - نیاز است که مرتبا این درخت ساخته شود.

در این مقاله یک الگوریتم برای مسیریابی در شبکه‌های حسگر بر مبنای کوچکترین درخت فراگیر ارائه می‌شود ولی سعی شده که مشکلات ذکر شده در بالا در آن پاسخ داده شود. برای این منظور اولا از کوچکترین درخت فراگیر محلی استفاده شده است که نیاز ناظر را از بین می‌برد و همچنین از یک ساختار جنبشی برای نگهداری آن استفاده می‌شود که مشکل هزینه تغییرات را از بین می‌برد.

در زمینه مسیریابی در شبکه‌های حسگر کارهای گوناگونی انجام شده است ولی در تمام آنها فرض بر ثابت بودن ساختار شبکه در طول حیات شبکه است. همچنین داده ساختارهای گوناگونی برای نگاهداری اجزای شبکه مطرح شده است ولی اکثر آنها هزینه به روز رسانی بالایی دارند و همچنین برای مسئله مسیریابی مناسب نیستند. لذا در این مقاله تلاش شد تا فرض‌های مطرح شده بسیار به محیط واقعی شبیه باشند که تا زمان نوشتن این مقاله کاری با این درجه شباهت با محیط واقعی پیدا نکردیم. نتیجه حاصل نیز هزینه نگاهداری و به روز رسانی کمینه‌ای دارد که برای حسگر های با انرژی محدود مناسب است.

در بخش‌های بعدی ابتدا یک الگوریتم برای کوچکترین درخت فراگیر محلی ارائه می‌شود. سپس یک روش جنبشی برای نگهداری کوچکترین درخت فراگیر ارائه می‌شود. در ادامه الگوریتم اصلی که ترکیبی از این دو روش است معرفی می‌شود و بعضی خواص آن اثبات می‌شود . در انتها پیچیدگی الگوریتم و نتیجه‌گیری آورده شده است.

۲- کوچکترین درخت فراگیر محلی

در این قسمت روشی برای ساخت کوچکترین زیر درخت فراگیر به صورت محلی ارائه می‌شود. ایده اصلی از روش ارائه شده توسط لی و همکارانش [1] گرفته شده است.

الگوریتم ساخت این درخت در دو فاز انجام می‌شود. در مرحله اول اطلاعات بین عناصر شبکه تبادل می‌شود و در مرحله دوم هر عنصر به صورت مجزا کوچکترین زیر درخت فراگیر را برای خود می‌سازد. در ادامه هر یک از دو فاز را به تفضیل شرح می‌دهیم.

فاز تبادل اطلاعات : در این فاز همانند مدل بردار فاصله[2] در مسیریابی درون دامنه‌ای عمل می‌شود. به این صورت که هر عنصر در شبکه اطلاعات خود را از تمام عناصر شبکه به صورت یک بردار فاصله به همسایگانش می فرستد. به علت اینکه عناصر از وجود تمام عناصر دیگر آگاه نیستند استفاده از شناسه الزامی است. پس از اتمام این فاز ، تمام عناصر و یا گره‌های شبکه ، اطلاعات کل شبکه را در اختیار دارند.

فاز ساخت کوچکترین زیر درخت فراگیر : در این فاز ، همانند فاز دوم در روش ارائه شده توسط لی و همکارانش [1] ، هر گره با استفاده از الگوریتمی مانند پریم [4] کوچکترین زیر درخت فراگیر را می سازد. در الگوریتم پریم درخت حاصل یکتا نیست زیرا در مواردی که فاصله دو گره از یک گره یکسان باشد به صورت اتفاقی یکی از آنها انتخاب می‌شود. ولی به منظور اینکه تمام عناصر دید یکسانی از این درخت

داشته باشند ، ما تابع فاصله را به صورت زیر تغییر داده‌ایم تا همیشه درخت یکتایی تولید شود.

$$d'(u_1,v_1) > d'(u_2,v_2) \Leftrightarrow$$
$$d(u_1,v_1) > d(u_2,v_2)$$
$$\vee \begin{pmatrix} (d(u_1,v_1) = d(u_2,v_2)) \\ \wedge (\max\{id(u_1), id(v_1)\} > \max\{id(u_2), id(v_2)\}) \end{pmatrix}$$
$$\vee \begin{pmatrix} (d(u_1,v_1) = d(u_2,v_2)) \\ \wedge (\max\{id(u_1), id(v_1)\} = \max\{id(u_2), id(v_2)\}) \\ \wedge (\min\{id(u_1), id(v_1)\} > \min\{id(u_2), id(v_2)\}) \end{pmatrix}$$

که در آن $d(u,v)$ برابر فاصله راس u از راس v است. در انتهای این فاز هر عنصر یک درخت فراگیر دارد که در تمام گره‌های مختلف شبکه یکسان هستند و در حقیقت روی آن توافق شده است. در صورتی که عناصر شبکه در یک صفحه باشند اثبات می شود که بزرگترین درجه راس‌های درخت حداکثر ۶ می‌شود. این نکته باعث کاهش قابل توجهی از انرژی مصرفی هر گره می‌شود.

تا این مرحله هر گره ، کوچکترین زیر درخت فراگیر لازم برای مسیریابی را ساخته است. در بخش بعد روشی برای نگهداری بهینه این درخت در موارد وجود حرکت و یا حذف و ایجاد گره‌های جدید با کمک یک داده ساختار جنبشی ارائه می‌شود.

۳- کوچکترین درخت فراگیر پارامتری و جنبشی

برای مدل کردن ساختار جنبشی گره‌ها می‌توان روش‌های مختلفی را پیش گرفت. در ابتدایی‌ترین حالت می‌توان فرض کرد معادله‌ی حرکت گره‌ها دقیقا مشخص است و بر پایه‌ی آن داده ساختار مساله را حل کرد. مشکل این روش این است که اولا معادله‌ی حرکت یک گره ممکن است بسیار پیچیده باشد و بدست آوردن اطلاعات لازم از آن کار ساده‌ای نباشد؛ دوما ماهیت معادله‌ی حرکت یک گره یک مفهوم پیوسته است و برای ما مناسب‌تر است اگر بتوانیم آن را به صورت یک مفهوم گسسته مدل کنیم. بنابراین از مدل معرفی شده توسط آگاروال و همکارانش [2] استفاده می‌کنیم که در آن به جای در نظر گرفتن معادله‌ی حرکت یک گره، تغییرات وزن یک یال را داریم و آن را یک تابع خطی در نظر می‌گیریم و برای گسسته کردن این تابع از رابطه‌ی $w_e(\lambda) = x_e - \lambda y_e$ برای یال e استفاده می‌کنیم. در این تابع دو عدد x_e و y_e دو عدد حقیقی هستند و λ به عنوان یک پارامتر گسسته تغییر کرده و باعث تغییر وزن یال‌ها می‌شود. به طور کلی دو دسته الگوریتم جنبشی برای حل مساله‌ی کوچکترین درخت فراگیر داریم که هر کدام را می‌توان با دیگری شبیه‌سازی نمود:

- الگوریتم جنبشی ساختاری: که در آن یال‌ها اضافه و حذف می‌شوند و تغییر وزن را با حذف و اضافه کردن یال شبیه‌سازی می‌کنیم.
- الگوریتم جنبشی تابعی: که توانایی تغییر وزن یال‌ها را دارد و اضافه و حذف یال‌ها را با استفاده از یک عدد بسیار بزرگ به عنوان وزن یال حذف شده شبیه‌سازی می‌کند.

یکی از تکنیک‌هایی که در این روش استفاده می‌شود روش تنک کردن است که عملا روش تقسیم و حل می‌باشد. در این روش گراف را به صورت بازگشتی به تعدادی دسته تقسیم می‌کنیم. نکته‌ای این تقسیم بندی‌ها دارند این است که درخت نهایی حاصل از گراف به راحتی از کنار هم قرار دادن جواب‌های زیر درخت‌ها حاصل از زیر گراف‌ها بدست می‌آید. اپستین و همکارانش [7] نشان دادند که این عمل نتیجه‌ی درستی می‌دهد. فرناندز و همکارانش [8] نیز نشان دادند که این روش برای مساله‌ی پارامتری نیز درست کار می‌کند و هزینه‌ی آن را نیز محاسبه کردند.

نقطه‌ی عطف این روش مطرح کردن ایده‌های هندسه‌ی محاسباتی در کاربرد تئوری گراف‌هاست؛ نشان داده می‌شود که می‌توان اطلاعات مربوط به گره‌ها را توسط پوش محدب نگهداری کرد؛ به این ترتیب که با توجه به دسته‌بندی که انجام می‌شود، مجموعه‌هایی داریم که برای داشتن درخت فراگیر باید یکی از یال‌ها را انتخاب و حذف کرد. اگر در این انتخاب بزرگترین عنصر مجموعه را حذف کنیم درخت ما کمینه خواهد بود. در این‌جا جنبش باعث می‌شود که این بزرگ‌ترین عنصر با تغییر λ که حاصل از جنبش است عوض شود و برای داشتن کوچکترین درخت فراگیر مجبور به تعویض یال شویم. با استفاده از پوش محدب می‌توانیم در زمان $lg(n)$ بزرگ‌ترین یال جدید را پیدا کنیم و جای یال قبلی را با آن عوض کنیم. روند کار به این ترتیب است که با استفاده از تبدیل هو [Hough59] معادله‌ی وزن یال‌ها بر اساس λ را تبدیل به نقاط می‌کنیم. در مساله‌ی دوگان بدست آمده خطی که بر دو پوش محدب مماس می‌شود مشخص می‌کند کدام دو خط باید جابجا شوند. این دو نقطه‌ی پیدا شده در عمل نشان دهنده‌ی بیش‌ترین رشد وزن در یال‌هایی که در درخت هستند و بیش‌ترین کاهش وزن در یال‌هایی که در درخت نیستند می‌باشند و اگر قرار باشد جای دو یال عوض شود باید این دو یال باشند. دو یال می‌توانند در جابجایی روابط زیر را با هم داشته باشند:

- جابجایی درون افرازی: هر دو در یک افراز هستند.
- جابجایی افراز دوگان: یالی که روی درخت فراگیر کمینه بود -باید حذف شود- در یکی از افرازهایی است که یک سر یال دیگر در آن است.
- جابجایی بین افرازی: دو یال رابطه‌های بالا را با هم ندارند.

به طور کلی با اضافه کردن راس می‌توانیم کاری کنیم که یک گراف فقط شامل راس‌هایی با درجه‌ی ۳ یا ۱ باشد و عملا حالت دودویی داشته باشد. برای گراف داده شده هم این کار را انجام می‌دهیم . سپس برای جلوگیری از گسترش اطلاعات مربوط به تغییر مکان یک گره در کل گراف، اقدام به افراز گره‌ها می‌کنیم. و با توجه به

رابطه‌ی دو یال که با هم عوض می‌شوند، اقدام در جهت بروز رسانی درخت می‌نماییم. افراز انجام شده باعث می‌شود که تغییرات حتی‌الامکان محلی باقی بمانند و از حدی که لازم نیست فراتر نروند.

۴- کوچکترین درخت فراگیر محلی با نگهداری به کمک داده ساختار جنبشی

در دو بخش قبل ، روش‌هایی برای ساخت کوچکترین زیر درخت فراگیر محلی و همچنین ساختاری جنبشی برای نگاهداری کوچکترین زیر درخت فراگیر آشنا معرفی شدند. متاسفانه هیچ کدام از این روش-ها برای شبکه‌های حسگر مناسب نیستند. کوچکترین زیر درخت فراگیر محلی به علت تغییرات زیاد در محل عناصر شبکه حسگر هزینه بسیار بالایی را ، هم از نظر انرژی و هم از نظر پیغام‌های رد و بدل شده ، دارد. در مقابل داده ساختار جنبشی ارائه شده نیز با وجود اینکه هزینه به روز رسانی مناسبی دارد ولی به علت اینکه محلی نیست لذا بایستی که تغییرات آن در همه سیستم منعکس شود که نیازمند ارتباطات بسیار زیادی در شبکه می‌باشد.

در این بخش مدلی ارائه می‌شود که در آن سعی شده است که از مزایای هر دو روش فوق استفاده شود و مدلی مناسب برای ساخت و نگاهداری کوچکترین زیر درخت فراگیر در شبکه‌های حسگر ارائه شود.

این الگوریتم در دو مرحله انجام می شود. در مرحله اول به کمک الگوریتم محلی داده شده در بخش ۲ تمام گره‌های شبکه یک زیر درخت فراگیر ایجاد می‌کنند. نکته قابل توجه این است که در پایان این مرحله تمام عناصر شبکه دید یکسانی از شبکه دارند. در مرحله دوم هر گره به کمک داده ساختار ارائه شده در بخش ۳ ، این درخت ایجاد شده را در صورت بروز تغییرات به روز می‌کند. سپس تغییرات منتشر می‌شود. نکته قابل توجه این است که این تغییرات فقط به گره-هایی ارسال می‌شود که در درخت فراگیر ، حداقل یکی از گره‌های مجاور آن تغییر کرده باشد یعنی یالی از آن حذف شده باشد و یا اینکه یال جدیدی به آن وارد شده باشد. در ادامه این دو مرحله را بیشتر شرح می‌دهیم.

در مرحله اول ، ابتدا اطلاعات مانند روش بردار فاصله در مسیریابی درون دامنه‌ای تبادل می‌شود تا تمام گره‌ها ، اطلاعات تمام عناصر شبکه را جمع آوری کنند. سپس با تابع فاصله تعریف شده در بخش ۲ ، کوچکترین زیر درخت فراگیر را ایجاد می‌کنند که همان طور که گفته شد یکتا است.

مرحله دوم که در حقیقت مرحله نگاهداری از درخت فراگیر ایجاد شده است ، هر گره به کمک ساختار جنبشی بخش ۳ درخت خود را به روز نگاه می‌دارد و در صورت نیاز همسایگان خود را از تغییرات درخت مطلع می‌سازد. در این مرحله هر گره با داشتن محل و جهت حرکت و سرعت خود و همچنین محل و جهت حرکت و سرعت همسایگان خود ، می‌تواند زمان دقیق تغییر در درخت فراگیر را محاسبه نماید و در صف وقایع خود قرار دهد. در هنگام ایجاد تغییر ، گره‌هایی که آن تغییر را در صف وقایع خود دارند ابتدا درخت خود را به روز می‌کنند. سپس نوبت به آن می‌رسد که تغییرات را به سایر گره-ها اطلاع دهند. در صورتی که قرار باشد که تمام گره‌ها از این موضوع مطلع شوند ، الگوریتم محلی بودن خود را از دست می‌دهد و به یک الگوریتم غیر محلی تبدیل می‌شود. برای حفظ محلی بودن الگوریتم هر گره که درخت فراگیر آن به روز شده است تغییرات را فقط برای گره-هایی می فرستد که یال‌های تغییر کرده در بیش از یکی از زیر درخت-های آن در زیر درخت فراگیر باشد. در این صورت انتشار تغییرات محدود به قسمت‌هایی است که تغییراتی در آنها رخ داده است و سایر گره‌ها در بخش‌های بدون تغییر ساختاری ، مطلع نخواهند بود.

نکته مهم در مورد مرحله دوم این است که چون تغییرات به صورت محلی منتشر می‌شود لذا بعد از مدتی گره‌های شبکه ، کوچکترین زیر درخت فراگیر متفاوتی از شبکه خواهند داشت و دیگر یکتایی که در مرحله اول وجود داشت رعایت نمی‌شود. بنابراین باید نشان دهیم که با وجود این دیدهای متفاوت ، همچنان خواص اصلی حفظ می‌شود. در ادامه لمی بیان می‌شود که در آن اثبات می‌کنیم که با وجود این دیدهای غیر یکسان همچنان بسته‌ها روی مسیرهای کوچکترین درخت فراگیر حرکت می‌کنند..

لم: *اگر برای فرستادن هر بسته ، هر گره آن بسته را به گره بعد از خود در مسیر بین خود و مقصد در درخت فراگیر تحویل دهد آنگاه بسته حتما روی مسیر موجود در کوچکترین زیر درخت فراگیر کل شبکه حرکت خواهد کرد.*

اثبات: فرض می‌کنیم که a بسته‌ای است که روی مسیر بهینه خود روی درخت فراگیر کلی حرکت نکرده است و گره u اولین گره-ای است که آن را در مسیر حرکت خود اشتباه هدایت کرده است. گره‌ای که u بسته را به آن هدایت کرده است را v می‌نامیم و گره‌ای که در درخت کلی بعد از u قرار دارد را v' می‌نامیم. زمان i را زمانی در نظر می‌گیریم که گره v از مسیر درخت اصلی خارج شده است و v' جایگزین آن شده است. چون که یال‌های گره u تغییر کرده است پس با توجه به الگوریتم حتما باید تغییرات به آن نیز اطلاع داده می‌شد. پس فرض وجود راس u اشتباه است و همیشه بسته روی مسیر اصلی منتقل می‌شود. □

بنابر لم فوق الگوریتم ما همیشه بسته‌ها را از مسیر بهینه کوچکترین زیر درخت فراگیر عبور می دهد.

۵- پیچیدگی الگوریتم

این الگوریتم همان طور که در بخش قبل گفته شد از دو مرحله تشکیل شده است. در این بخش پیچیدگی هر یک را به صورت جداگانه بررسی می کنیم.

در فاز پخش اطلاعات در مرحله اول همان طور که گفته شد به کمک الگوریتم بردار فاصله تمام گره‌ها ساختار درخت فراگیر را در خود ایجاد می‌کنند. این عمل نیازمند جابجایی تعداد زیادی پیغام است و در نتیجه کارآیی بالایی ندارد. اما نکته مهم این است که این مرحله تنها یک بار و در ابتدای کار شبکه انجام می‌شود و در سایر مراحل دیگر تکرار نمی‌شود. سپس هر گره به کمک الگوریتم پریم کوچکترین زیر درخت فراگیر را با هزینه $O(E + V \lg V)$ محاسبه می‌کند.

در مرحله دوم بایستی که ساختار درخت در مقابل تغییرات روی داده حفظ شود. اثبات شده است که الگوریتم بخش ۳ می‌تواند یک گراف را با E یال را به کمک ساختار جنبشی با هزینه $O(pE^{2/3} \log^{4/3} E)$ ، که در آن p برابر تعداد افزوده شدن یال و یا حذف یال و یا تغییر حاصل در ساختار درخت است ، نگاهداری کند. همچنین اثبات می‌شود که اگر از تکنیک تصادفی کردن استفاده شود هزینه برابر $O(pE^{2/3} \log E)$ می‌شود. اما نکته مهم در مورد این الگوریتم ، هزینه پایین انتشار تغییرات است که چون محلی عمل می‌شود بسیار اندک می‌باشد.

۶- نتیجه

در این مقاله یک الگوریتم جنبشی و در عین حال محلی برای ساخت و نگاهداری کوچکترین زیر درخت فراگیر ارائه شده است. این دو ویژگی ، این الگوریتم را برای مسئله مسیریابی در شبکه‌های حسگر مناسب می‌نماید. همچنین اثبات شده است که با وجود اینکه گره‌های مختلف با گذشت زمان دید یکسانی از کل شبکه ندارند ولی باز هم بسته‌ها در مسیر بهینه کوچکترین درخت فراگیر کلی حرکت می‌کنند.

به نظر می رسد که مرحله اول انتشار ساختار شبکه که فعلا به کمک الگوریتم بردار فاصله انجام می‌شود می‌تواند به کمک الگوریتم‌های با تعداد پیغام کمتر جایگزین شود که می‌تواند در آینده توسعه‌ای بر مدل کنونی باشد. همچنین یافتن الگوریتم‌های جنبشی مناسب‌تر نیز می تواند به عنوان کارهای بعدی انجام پذیرد.

مراجع

[1] P. Santi, *Topology control in wireless ad hoc and sensor networks*, ACM Computer Survey 37, 2 (Jun. 2005), 164-194.

[2] P. K. Agarwal, D. Eppstein, , L. J. Guibas, M. R. Henzinger, *Parametric and Kinetic Minimum Spanning Trees*, In Proceedings of the 39th Annual Symposium on Foundations of Computer Science (November 08 - 11, 1998), FOCS IEEE Computer Society, Washington, DC, 596.

[3] N. Li, J. C. Hou, L. Sha, *Design and analysis of an MST-based topology control algorithm*, in Proceedings of the IEEE Infocom 2003.

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

[5] W. Baek, David S. L. Wei, C. C. Jay Kuo, *Power-Aware Topology Control for Wireless Ad-Hoc Networks*, IEEE SECON 2005.

[6] C. Gentile and R.E. VanDyck, *Kinetic Spanning Trees for Minimum Power Routing in MANETs*, IEEE VTC Spring, Birmingham, AL., May 2002.

[7] D. Eppstein, Z. Galil, G. F. Italiano, A. Nissenzweig, *Sparsification - A technique for speeding up dynamic 11 graph algorithms*, Journal of ACM, 44(1):669--696, 1997.

[8] D. Fernandez-Baca, G. Slutzki, and D. Eppstein, *Using sparsification for parametric minimum spanning tree problems*, Proc. 5th Scand. Workshop Algorithm Theory. Springer-Verlag, Lecture Notes in Computer Science 1097, July 1996.

زیرنویس‌ها

[1] Ad−Hoc Networks
[2] Distance Vector