

الگوریتم‌های جدید مسیریابی برخط و با حافظه‌ی محدود برای مثلث‌بندی‌های دلانی

اشکان رضازاده^۱، مصطفی نوری‌بایگی^۲

^۱ دانشجوی کارشناسی ارشد، دانشکده‌ی مهندسی کامپیوتر، دانشگاه فردوسی مشهد، ashkan.rezazadeh@mail.um.ac.ir

^۲ استادیار، دانشکده‌ی مهندسی کامپیوتر، دانشگاه فردوسی مشهد، nouribaygi@um.ac.ir

چکیده

«مسیریابی» همواره به عنوان یکی از مسائل اساسی و مهم در علوم کامپیوتری به شمار می‌رود و در بسیاری از زمینه‌ها از جمله رباتیک و شبکه‌های بی‌سیم کاربرد دارد. این مسئله اغلب در یک فضای هندسی که توسط یک «گراف هندسی» قابل مدل‌سازی است مطرح می‌شود، یکی از مفیدترین انواع گراف‌های هندسی که ویژگی‌های مفیدی جهت سهولت مسیریابی در بر دارد «مثلث‌بندی دلانی» است.

هنگامی که اطلاعات کاملی از گراف مورد نظر در دسترس باشد با استفاده از الگوریتم‌های پیدا کردن مسیر بهینه در گراف‌های جهت‌دار از جمله Dijkstra می‌توان به مسیر بهینه رسید، اما در حالت «برخط» که اطلاعات موجود محدود هستند و در هر مرحله انتخاب بسته/رباط مورد نظر باید فقط بر اساس اطلاعات محلی صورت بگیرد، مسئله با چالش‌های بیشتری روبرو می‌گردد.

در این مقاله، ما چهار الگوریتم برخط با پیچیدگی حافظه $O(1)$ برای مثلث‌بندی‌های دلانی را ارائه نموده و با مقایسه‌ی عملکرد آن‌ها با الگوریتم‌های مشابه روی نمونه‌های شبیه‌سازی شده، آن‌ها را ارزیابی می‌نماییم. نتایج به دست آمده در آزمایشات ما نشان می‌دهند که الگوریتم‌های جدید در عین حفظ سادگی و آسان بودن پیاده‌سازی، عملکرد بیشتر الگوریتم‌های مشابه موجود را بهبود می‌دهند.

کلمات کلیدی

مسیریابی، مسیریابی برخط، مثلث‌بندی دلانی، مسیریابی هندسی

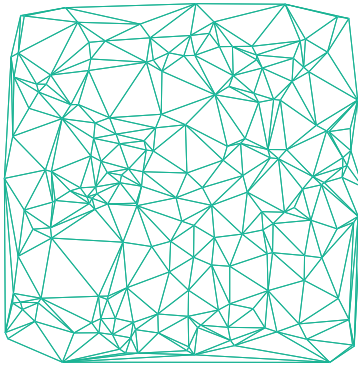
۱ مقدمه

یافتن یک مسیر در گراف یک مسئله‌ی اساسی در علوم کامپیوتر بوده و در چندین زمینه‌ی مختلف از جمله سیستم‌های اطلاعات جغرافیایی، شبکه‌ی حس‌گر بی‌سیم و رباتیک کاربرد دارد. در سال‌های اخیر، گسترش شبکه‌های بی‌سیم و وسیله‌های GPS موجب انجام تحقیقات زیادی روی الگوریتم‌های مسیریابی برای شبکه‌های هندسی شده‌اند. [۱، ۳، ۴]

مسیریابی اغلب در یک فضای هندسی که توسط یک گراف هندسی قابل مدل‌سازی است اتفاق می‌افتد و مسئله‌ی ما به یافتن یک مسیر از مبدا s به مقصد t در گراف هندسی G تبدیل می‌شود.

گراف هندسی $G = (V, E)$ گرافی وزن‌دار است که V مجموعه‌ی رئوس آن و زیرمجموعه‌ای از نقاط روی صفحه و E مجموعه‌ی یال‌های آن و شامل تعدادی پاره‌خط بین نقاط V است به طوری که وزن هر یک از این یال‌ها

برابر با فاصله‌ی اقلیدسی میان نقاط ابتدایی و انتهایی آن است. یکی از انواع مفید گراف‌های هندسی که به علت دارا بودن خصیصه‌های مطلوب، مثلث‌بندی دلانی است. اگر مجموعه‌ی P از نقاط روی صفحه را طوری مثلث‌بندی کنیم که هیچ یک از نقاط P درون دایره‌ی محیطی گذرنده از رئوس هر یک از مثلث‌ها قرار نگیرد، مثلث‌بندی حاصل، «دلانی» نام دارد. شکل ۱ نمونه‌ای از مثلث‌بندی دلانی برای مجموعه‌ی ۲۰۰ نقطه در صفحه را نشان می‌دهد.

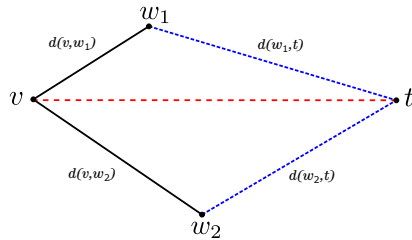


شکل ۱: نمونه‌ای از یک مثلث‌بندی دلانی برای یک مجموعه‌ی ۲۰۰ عضوی از نقاط صفحه مثلث‌بندی‌های دلانی به علت دارایی خصیصه‌های مطلوب جهت کمک به مسیریابی، چندین بار به عنوان توپولوژی شبکه پیشنهاد شده‌اند [۵، ۶، ۷]، برخی از این ویژگی‌ها عبارت‌اند از:

- مثلث‌بندی‌های دلانی در دسته‌ی گراف‌های مسطح قرار می‌گیرند.
- تعداد پیوندهای مثلث‌بندی دلانی کمتر از سه برابر تعداد رئوس آن و میانگین درجه‌ی هر راس کمتر از شش است [۸]، این دو نکته موجب تسهیل مسیریابی می‌شوند.
- مثلث‌بندی دلانی یک پوشش هندسی [۹] و ضریب پوشش آن ثابت و بین $1/5846$ و $1/998$ است [۱۰، ۱۱]. اگر طول کوتاه‌ترین مسیر بین هر دو راس u و v در گراف G حداکثر c برابر فاصله‌ی اقلیدسی میان u و v باشد، گراف G یک پوشش نامیده می‌شود که c ضریب پوشش (کشش) آن است. اگر G هندسی باشد، به آن پوشش هندسی نیز می‌گویند.

هنگامی که اطلاعات کامل شبکه پیش از مسیریابی در دسترس باشد، می‌توانیم با استفاده از الگوریتم‌های یافتن مسیر بهینه در گراف وزن‌دار مثل Dijkstra [۱۲] بهترین مسیر را بیابیم، اما در حالت برخط^۱ که شناخت کاملی

^۱Online



شکل ۲: ایده‌ی اصلی الگوریتم Two-step

وجود شرط $|wt| < |pt|$ این تضمین را به ما می‌دهد که فاصله‌ی بسته‌اریات ما تا t در هر مرحله کاهش یافته و در نهایت به t می‌رسیم. شکل ۲ درک بهتری از عملکرد این الگوریتم را در بر دارد. اگر یک الگوریتم مسیریابی برخط مانند A بتواند از هر راس مبدا مانند s به هر راس مقصد مانند t مسیری را در گراف G بیابد، می‌گوییم A برای G کار می‌کند [۱۷].

در حالی که الگوریتم‌های مسیریابی برخط و بدون حافظه‌ی ارائه شده برای تمامی مثلث‌بندی‌های دلانی کار می‌کنند، اما رقابتی بودن هیچ‌کدام تا به امروز اثبات نشده است، Bose و Morin [۱۳] با ارائه‌ی چند مثال نقض، رقابتی نبودن الگوریتم‌های Greedy، Randomized Compass و Compass را اثبات نمودند.

علاوه بر الگوریتم‌های مطرح شده دسته‌ی دیگری از الگوریتم‌ها که از حافظه‌ی ثابت و محدودی با پیچیدگی $O(1)$ استفاده می‌کنند و رقابتی هم هستند نیز برای مثلث‌بندی‌های دلانی ارائه شده اند که اولین مورد با ضریب رقابتی $45/75$ توسط Bose و Morin [۱۳] ارائه گردید و پس از آن چند الگوریتم دیگر با ضرایب رقابتی کوچک‌تر برای مثلث‌بندی‌های دلانی ارائه شدند که جدیدترین آن‌ها الگوریتم MixedChordArc [۱۵] با ضریب رقابتی کمتر یا مساوی $3/56$ است.

۳ الگوریتم‌های پیشنهادی

در این بخش به توصیف الگوریتم‌های پیشنهادی زیر می‌پردازیم:

۱. Two-step with average stretch factor

۲. Two-step with piecewise estimate of average stretch factors

۳. 2-local Two-step with average stretch factor

۴. Two-step Compass

با توجه به عملکرد بهتر الگوریتم مسیریابی Two-step از نظر طول اقلیدسی مسیر یافت شده نسبت به الگوریتم‌های مشابه، الگوریتم‌های ما بر پایه‌ی این الگوریتم و با هدف بهبود عملکرد در حالات مختلف طراحی شده‌اند. یکی از مفاهیمی که در الگوریتم‌های پیشنهادی مورد استفاده قرار گرفته است، مفهوم ضریب کشش^۲ گراف است. نسبت طول کوتاه‌ترین مسیر از هر راس u به هر راس v در G به فاصله‌ی اقلیدسی میان u و v را کشش (u, v) در G می‌نامیم؛ بزرگ‌ترین کشش در G به عنوان ضریب کشش G در نظر گرفته می‌شود. حال به ارائه‌ی الگوریتم‌های پیشنهادی می‌پردازیم.

۱-۳ Two-step with average stretch factor

با توجه به ایده‌ی اصلی الگوریتم مسیریابی Two-step که کمینه کردن $|pw| + |wt|$ در هر مرحله است، می‌توان گفت در این الگوریتم از $|wt|$ به عنوان تخمینی از طول مسیر بین w و t استفاده می‌شود.

از شبکه موجود نیست و تصمیم‌گیری در هر مرحله تنها تابعی از اطلاعات محلی و اطلاعات محدود موجود در حافظه است، چالش مسئله بیشتر می‌شود.

Morin و Bose [۱۳] الگوریتم‌های مسیریابی برخط را بر اساس میزان استفاده‌ی آن‌ها از حافظه و یا تصادفی بودن‌شان دسته‌بندی کرده‌اند. آن‌ها الگوریتمی که انتخابش در هر مرحله فقط با استفاده از مکان راس فعلی، همسایه‌های سطح اول آن و مکان مقصد صورت بگیرد را بدون حافظه می‌نامند. الگوریتم‌های بدون حافظه یا با حافظه‌ی محدود به علت سادگی، مصرف کم توان، ظرافت و مقیاس‌پذیری توجهات زیادی را به خود جلب کرده‌اند [۱۳، ۲، ۱۴، ۱۵].

الگوریتم‌های برخط بدون حافظه یا با حافظه‌ی محدود اغلب قادر به یافتن مسیر بهینه نیستند، بنابراین هدف پیدا کردن مسیری با طول هر چه نزدیک‌تر به مسیر بهینه است. اگر طول مسیر پیدا شده توسط الگوریتم A از هر راس مبدا s به هر راس مقصد t در G بیشتر از c برابر طول کوتاه‌ترین مسیر موجود از t در G نباشد، الگوریتم A c -رقابتی نامیده و c را به عنوان ضریب رقابتی آن می‌شناسیم.

در این مقاله، ما چهار الگوریتم مسیریابی برخط برای مثلث‌بندی‌های دلانی ارائه می‌نماییم که سه مورد با حافظه‌ی ثابت با پیچیدگی $O(1)$ و یک مورد بدون حافظه است و همچنین به ارزیابی و مقایسه‌ی عملکرد آن‌ها با الگوریتم‌های مشابه می‌پردازیم. ساختار ادامه‌ی مقاله بدین شرح است: در بخش ۲ پیشینه‌ی مسئله بررسی می‌گردد، بخش ۳ شامل معرفی و توصیف الگوریتم‌های جدید است، بخش ۴ شامل ارزیابی عملکرد الگوریتم‌هاست، بخش ۵ شامل مسائل باز و کارهای آینده بوده و بخش ۶ به نتیجه‌گیری اختصاص دارد.

۲ پیشینه

ابتدا با معرفی چند نماد شروع می‌کنیم. از این به بعد راس مبدا با s ، راس مقصد با t ، راس فعلی در هر مرحله با p ، همسایه‌هایی با فاصله‌ی یک یال از p با $N(p)$ ، فاصله‌ی اقلیدسی میان دو راس u و v با $|uv|$ و زاویه‌ی کم‌تر یا مساوی 180° بین دو پاره‌خط uv و vw با $\angle uvw$ نمایش داده می‌شوند.

تا به امروز، چندین الگوریتم مسیریابی برخط، بدون حافظه و غیر رقابتی برای مثلث‌بندی دلانی ارائه شده است که برخی از آن‌ها از زوایا، برخی از فواصل اقلیدسی و برخی از ترکیب این دو برای ارزیابی همسایگان خود و تصمیم‌گیری استفاده می‌کنند.

الگوریتم‌های مسیریابی قطعی Compass [۱۶] و Apex-angle [۱۴] و الگوریتم تصادفی Randomized Compass [۱۳] در دسته‌ی اول، الگوریتم‌های Two-step [۱۴] و مجموعه‌ی الگوریتم‌های Distance Referencing (مبتنی بر فاصله) از جمله Greedy [۱۳] و Midpoint [۱۴] در دسته‌ی دوم و مجموعه‌ی الگوریتم‌های Deterministic Compass از جمله Compass Midpoint [۱۴] و Greedy-Compass [۱۷] در دسته‌ی سوم قرار می‌گیرند.

طبق نتایج به دست آمده در آزمایشات ما و همچنین نتایج ارائه شده توسط Si و Zomaya [۱۴]، در بین الگوریتم‌های مذکور، Two-step از نظر طول اقلیدسی و Greedy از نظر تعداد گره‌های مسیر دارای بهترین عملکرد هستند. به همین علت و به جهت اهمیت الگوریتم Two-step در این مقاله، نیاز است با نحوه‌ی کار این الگوریتم آشنا شویم.

این الگوریتم در هر مرحله راس $w \in N(p)$ که به ازای آن $|wt| < |pt|$ بوده و $|pw| + |wt|$ کمینه شود را به عنوان مقصد بعدی انتخاب می‌کند، اگر این شرایط برای چند راس صدق کند، یکی از آن‌ها به دلخواه انتخاب می‌شود.

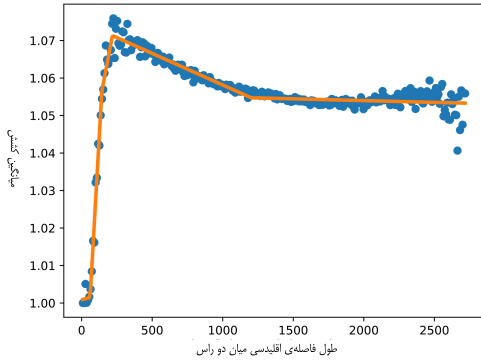
²Stretch factor

اگر انحراف از معیار مجموعه‌ی کشش‌های جفت رؤس گراف پایین باشد، میانگین کشش می‌تواند تخمین مناسبی از طول مسیر بین دو راس باشد، در غیر این صورت واضح است که مقادیر خیلی به میانگین نزدیک نیستند و در نتیجه ممکن است میانگین کشش نیز تخمین دقیقی نباشد.

در این الگوریتم، ما «تخمین تکه‌ای نمودار میانگین کشش براساس فواصل اقلیدسی» را جایگزین میانگین کشش می‌کنیم.

اگر N تعداد رؤس گراف باشد، محور افقی نمودار مورد نظر با تقسیم بازه‌ی 0 تا بیشترین فاصله‌ی اقلیدسی بین دو راس در گراف به $\frac{N_0}{1000}$ بازه‌ی مساوی به دست می‌آید، در این نمودار به ازای هر بازه یک نقطه وجود دارد که عرض آن از طریق محاسبه‌ی میانگین کشش‌های تمامی جفت رؤوسی که فاصله‌ی اقلیدسی آن‌ها متعلق به آن بازه است به دست می‌آید.

به منظور جلوگیری از مصرف بالای حافظه، نمودار را با تخمینی از طریق یک تابع خطی تکه‌ای به یک نمودار خطی ۵ قسمتی تبدیل و مختصات نقاط ابتدا و انتهای هر قسمت (۶ نقطه) را در هر گره با استفاده از یک حافظه‌ی محدود با پیچیدگی $O(1)$ ذخیره می‌کنیم. این مختصات ثابت بوده و تغییر نمی‌کنند. شکل ۴ نمونه‌ای از نمودار حاصل برای گرافی با 600 راس است.



شکل ۴: نمودار تخمین تکه‌ای میانگین کشش بر اساس فواصل اقلیدسی برای گرافی با 600 راس
رویکرد الگوریتم بدین شرح است: p در هر مرحله، از میان همسایگانش راس $w \in N(p)$ که به ازای آن $|wt| < |pt|$ بوده و

$$|pw| + (|wt| \cdot AvgStretchPerDist(|wt|))$$

کمینه گردد را به عنوان مقصد بعدی انتخاب می‌کند. در این الگوریتم $AvgStretchPerDist(|wt|)$ که بیان‌گر کشش میانگین مناسب برای w و t است و با توجه به $|wt|$ و مختصات ذخیره شده از نمودار به دست می‌آید جایگزین $AvgStretchFactor$ در الگوریتم قبلی شده است.

۳-۳ 2-local Two-step with average stretch factor

اگر در یک الگوریتم مسیریابی برخط، p مختصات همسایگانی که حداکثر k گره از آن فاصله دارند را در اختیار داشته باشد، به آن الگوریتم k -محلی می‌گویند. الگوریتمی که در این بخش به معرفی آن می‌پردازیم، نسخه‌ی ۲-محلی دو الگوریتم قبلی است.

دلایل انتخاب $k=2$ عبارت‌اند از امکان ذخیره‌ی مختصات همسایگان با فاصله در یک حافظه با پیچیدگی $O(1)$ و بیشت بودن بهبود عملکرد هنگام تغییر k از ۱ به ۲ نسبت به ۲ به ۳ با توجه به کندتر شدن اجرا در صورت افزایش k .

در اکثر موارد مسیریابی خصوصاً که کوتاه‌ترین مسیر از p به t از بیش از ۲ یال تشکیل شده است، کشش جفت راس (w, t) بیشتر از ۱ بوده و با بیشتر شدن تعداد یال‌های مسیر، بزرگ‌تر هم می‌شود، لذا در بسیاری از موارد ممکن است $|wt|$ تخمین دقیقی از طول مسیر بین w و t نباشد. در این الگوریتم با استفاده از میانگین کشش دقت این تخمین را افزایش مدهیم.

میانگین کشش G یک عدد بزرگ‌تر یا مساوی با ۱ است و از طریق محاسبه‌ی میانگین کشش تمامی جفت رؤس G در هنگام ساخت شبکه به دست می‌آید.

رویکرد این الگوریتم که مشابه Two-step بوده و تفاوتش در تخمین طول مسیر w به t است، بدین شرح است: در هر مرحله، p راس $w \in N(p)$ که به ازای آن $|wt| < |pt|$ بوده و

$$|pw| + (|wt| \cdot AvgStretchFactor)$$

کمینه گردد را به عنوان مقصد بعدی بسته/ریات انتخاب می‌کند. $AvgStretchFactor$ بیان‌گر میانگین کشش گراف است. جزئیات بیشتر الگوریتم در شکل ۳ قابل مشاهده است. لازم به ذکر است که میانگین کشش گراف ثابت بوده و می‌تواند با استفاده از یک حافظه‌ی ثابت با پیچیدگی $O(1)$ در اختیار هر گره و در مسیریابی مورد استفاده قرار بگیرد.

Algorithm 1 Two-step with average stretch factor

```

for all  $w \in N(p)$  do
  if  $w$  has a smaller  $|wt|$  than  $|pt|$  then
     $sum \leftarrow |pw| + (|wt| \cdot AvgStretchFactor)$ 
    if  $w$  has a smaller  $sum$  than previous neighbors
      then
         $next(p) \leftarrow w$ 
    end if
  end if
end for

```

شکل ۳: شبه کد الگوریتم Two-step with average stretch factor

۲-۳ Two-step with piecewise estimate of average stretch factors

در این الگوریتم قصد داریم دقت تخمین طول مسیر بین دو راس را بیش از الگوریتم قبلی بالا ببریم. به منظور ارائه‌ی این روش نیاز است با دو مفهوم زیر آشنا شویم:

- **انحراف معیار**^۳

در علم آمار، انحراف از معیار (نماد σ) یکی از شاخص‌های میزان پراکندگی مجموعه‌ای از مقادیر است که نشان می‌دهد به‌طور میانگین داده‌ها چه مقدار از مقدار متوسط فاصله دارند.

- **تابع خطی تکه‌ای**^۴

یک تابع خطی تکه‌ای، تابعی است که از چند بخش خطی تعریف شده در تعداد مساوی از فواصل و اغلب با اندازه‌ی برابر تشکیل شده است.

³Standard deviation

⁴Piecewise linear function

Algorithm 3 Two-step-Compass

```
for all  $w \in N(p)$  do
  if  $w$  has a smaller  $|wt|$  than  $|pt|$  then
     $sum \leftarrow \alpha \cdot (|pw| + |wt|) + \beta \cdot \angle tpw$ 
    if  $w$  has a smaller  $sum$  than previous neighbors then
       $next(p) \leftarrow w$ 
    end if
  end if
end for
```

شکل ۶: شبه کد الگوریتم Two-step-Compass

۴ ارزیابی

در این بخش به ارزیابی و مقایسه‌ی الگوریتم‌های پیشنهادی با الگوریتم‌های مشابه می‌پردازیم. در میان معیارهای ارزیابی الگوریتم‌های مسیریابی، کیفیت مسیر در مسیریابی هندسی مهم‌ترین معیار به شمار می‌رود. معیار ارزیابی کیفیت مسیری که ما استفاده می‌کنیم، کشش میانگین الگوریتم (*average stretch*) است [۱۸]. کشش الگوریتم A برای جفت راس (u, v) در G برابر است با نسبت طول مسیر پیشنهادی از u به v توسط A به کوتاه‌ترین مسیر موجود از u به v . کشش میانگین A با محاسبه‌ی میانگین کشش‌های A برای تمام جفت رئوس G به دست می‌آید. دو معیار متداول طول مسیر که ما از آن‌ها استفاده می‌کنیم عبارت‌اند از:

- اقلیدسی: جمع طول اقلیدسی یال‌های پیمایش‌شده در مسیر (نرم L_2)
- پیوند: تعداد گره‌های پیمایش‌شده در مسیر

و الگوریتم‌های مشابهی که در این بخش از آن‌ها استفاده می‌کنیم عبارت‌اند از:

- Greedy: بهترین در معیار پیوند (بر اساس نتایج آزمایش‌ها)
- Midpoint: دومین بهترین در معیار پیوند (بر اساس نتایج آزمایش‌ها)
- Two-step: بهترین در معیار اقلیدسی (بر اساس نتایج آزمایش‌ها)
- Two-local Two-step: نسخه‌ی ۲-محلی الگوریتم Two-step
- MixedChordArc: بهترین الگوریتم رقابتی با حافظه‌ی ثابت $O(1)$

۴-۱ شرایط آزمایش

آزمایشات از طریق توسعه‌ی یک برنامه و پیاده‌سازی الگوریتم‌های مسیریابی در آن انجام شده‌اند. این برنامه ۱۰۰ گراف تصادفی با ۱۰۰، ۲۰۰، ۴۰۰، ۶۰۰، ۸۰۰ و ۱۰۰۰ راس (در مجموع ۶۰۰ نمونه) تولید نموده و الگوریتم‌ها را روی آن‌ها اجرا می‌نماید. مجموعه‌های نقاط نیز به شکل تصادفی و با توزیع یکنواخت در یک محیط مربعی تولید شده‌اند، این توزیع رایج‌ترین نوع در تحقیقات انجام شده است [۴، ۱۹].

ما به منظور محاسبه‌ی کوتاه‌ترین مسیر از الگوریتم مشهور Dijkstra استفاده کرده‌ایم. همچنین مقایسه‌ی عملکرد الگوریتم‌ها را با بررسی نمودار میانگین کشش میانگین بر اساس تعداد رئوس انجام می‌دهیم. برای هر تعداد راس، میانگین ۱۰۰ کشش میانگین به دست آمده (به ازای هر گراف) به عنوان عرض نقاط در نمودار استفاده می‌شود.

لازم به ذکر است که در آزمایشات هر دو پارامتر الگوریتم Two-step-Compass را برابر با ۱ فرض کرده‌ایم.

رویکرد الگوریتم بدین شرح است: p در هر مرحله، پیام/بسته را به راس $w \in N(p)$ می‌فرستد که به ازای آن $|pt| < |wt|$ و راسی مانند $w' \in N(w)$ که به ازای آن $|w't| < |wt|$ بوده و

$$|pw| + |ww'| + (|ww't| \cdot AvgStretchFactor)$$

کمینه گردد، وجود داشته باشد. لازم به ذکر است که در این الگوریتم $AvgStretchPerDist(|wt|)$ و $AvgStretchFactor$ هر دو قابل استفاده‌اند. جزئیات بیشتر الگوریتم در شکل ۵ قابل مشاهده است.

Algorithm 2 2-local Two-step with average stretch factor

```
for all  $w \in N(p)$  do
  if  $|wt|$  is less than  $|pt|$  then
    for all  $w' \in N(w)$  do
      if  $|w't|$  is less than  $|wt|$  then
         $sum \leftarrow |pw| + |ww'| +$ 
           $(|w't| \cdot AvgStretchFactor)$ 
        if  $w'$  has a smaller  $sum$  than previous ones then
           $next(p) \leftarrow w$ 
        end if
      end if
    end for
  end if
end for
```

شکل ۵: شبه کد الگوریتم 2-local Two-step with average stretch factor

۳-۴ Two-step-Compass

همان‌طور که از نام آن پیداست، ایده‌ی اصلی این الگوریتم ترکیب دو الگوریتم Two-step و Compass است که اولی از نظر طول اقلیدسی و دومی از نظر تعداد گره‌های مسیر عملکرد بهتری نسبت به دیگری دارد. این الگوریتم بدون حافظه جهت بهبود عملکرد دو الگوریتم پایه و عملکرد قابل قبول در هر دو معیار طراحی شده است. در این الگوریتم دو ضریب $\alpha, \beta \in [0, 1]$ نیز جهت انعطاف و کارایی در سناریوهای متفاوت مورد استفاده قرار می‌گیرند.

رویکرد این الگوریتم بدین شرح است: در هر مرحله p راس $w \in N(p)$ که به ازای آن $|pt| < |wt|$ باشد و

$$\alpha \cdot (|pw| + |wt|) + \beta \cdot \angle tpw$$

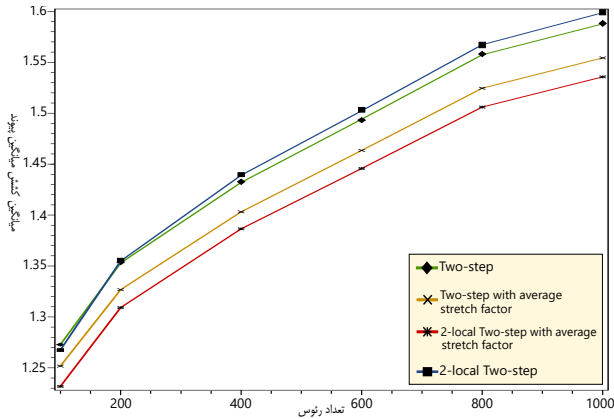
کمینه گردد را به عنوان مقصد بعدی انتخاب می‌کند. جزئیات بیشتر الگوریتم در شکل ۶ قابل مشاهده است. اگر کوتاه‌تر کردن طول اقلیدسی مسیر مهم‌تر باشد (مثل رباتیک) باید $\alpha > \beta$ و مقدارش به ۱ نزدیک‌تر باشد، اما اگر کاهش تعداد گره‌های مسیر مهم‌تر باشد (مثل شبکه‌های حسگر بی‌سیم) باید $\alpha > \beta$ و مقدارش به ۱ نزدیک‌تر باشد.

در [۱۴، ۱۳] ثابت شده است که الگوریتم‌های Two-step و Compass برای مثلث‌بندی‌های دلانی کار می‌کنند، الگوریتم‌های ما نیز بر پایه‌ی یک یا هر دوی این الگوریتم‌ها طراحی شده و عمل می‌کنند، لذا می‌توان گفت الگوریتم‌های پیشنهادی نیز برای مثلث‌بندی‌های دلانی کار می‌کنند.

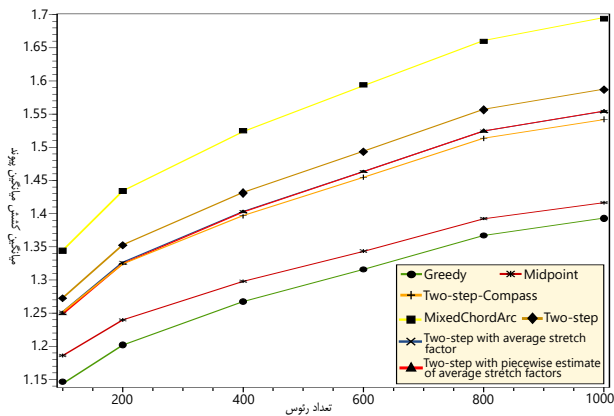
۲-۴ نتایج آزمایشات

شکل ۷ نشان می‌دهد که الگوریتم‌های ۱-محلی پیشنهادی با حفظ سادگی و ظرافت، عملکرد الگوریتم‌های دیگر از جمله Two-step و MixedChordArc را در معیار اقلیدسی بهبود می‌دهند، علاوه بر این با مشاهده‌ی شکل می‌توان گفت الگوریتم Two-step with piecewise es- در معیار اقلیدسی بهترین عملکرد را در بین الگوریتم‌های ۱-محلی دارد.

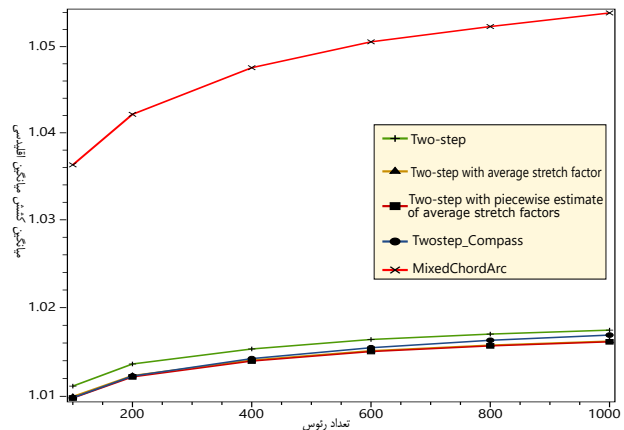
علاوه بر این بررسی این نمودار نشان می‌دهد که استفاده از «تخمین تکه‌ای» به جای «میانگین کشش» بهبود چندانی حاصل نمی‌کند. در این نمودار مشخص است که الگوریتم Two-step-Compass در معیار پیوند، بر خلاف اقلیدسی دارای عملکرد بهتری نسبت به سایر الگوریتم‌های پیشنهادی ۱-محلی است.



شکل ۹: مقایسه‌ی عملکرد نسخه‌ی ۲-محلی و ۱-محلی دو الگوریتم Two-step و Two-step with average stretch factor با معیار پیوند



شکل ۱۰: مقایسه‌ی عملکرد الگوریتم‌های ۱-محلی در معیار پیوند



شکل ۷: مقایسه‌ی عملکرد الگوریتم‌های ۱-محلی در معیار اقلیدسی

۵ مسائل باز

در این بخش به طرح دو مسئله‌ی باز برای تحقیقات آینده می‌پردازیم:

- نتایج آزمایشات نشان می‌دهند عملکرد برخی الگوریتم‌ها (خصوصاً بدون حافظه) در معیار اقلیدسی و پیوند عکس هم است، آیا می‌توان یک الگوریتم بدون حافظه طراحی نمود که در هر دو معیار بهترین باشد؟
- علی‌رغم این که الگوریتم‌های پیشنهادی در مثال‌های نقض ارائه شده در [۱۳] جهت اثبات غیررقابتی بودن چند الگوریتم موفق عمل می‌کنند، رقابتی بودن یا نبودن آن‌ها به اثبات نرسیده است. در صورت اثبات رقابتی بودن، ارزش و موارد کاربرد این الگوریتم‌ها بیشتر خواهد شد.

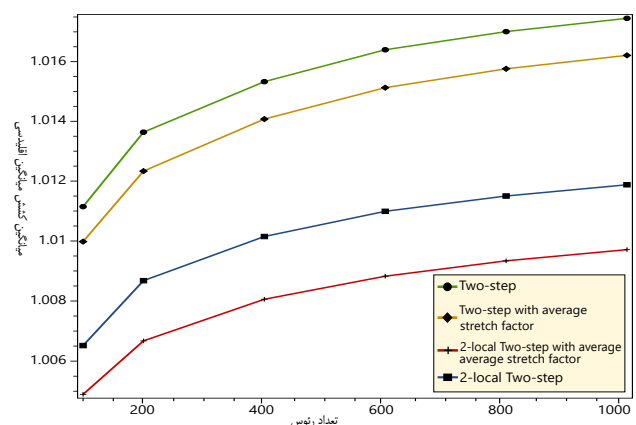
۶ نتیجه‌گیری

در این مقاله به معرفی چهار الگوریتم مسیریابی برخط برای مثلث‌بندی دلانی پرداختیم:

- Two-step with average stretch factor
- Two-step with piecewise estimate of average stretch factors
- 2-local Two-step with average stretch factor
- Two-step Compass

سه مورد اول به یک حافظه‌ی محدود و با پیچیدگی $O(1)$ نیاز داشته و مورد چهارم بدون حافظه است. تمامی آن‌ها بر پایه‌ی الگوریتم مسیریابی Two-step و با هدف بهبود عملکرد طراحی و معرفی شدند، هر کدام از این الگوریتم‌ها خصیصه‌هایی مشخصی دارند و می‌توانند در سناریوهای مختلف کارآمد باشند.

با توجه به مقایسه‌ی نسخه‌های محلی و ۲-محلی الگوریتم‌های Two-step و Two-step with average stretch factor در معیار اقلیدسی و پیوند در شکل ۸ و ۹، در معیار اقلیدسی نسخه‌ی ۲-محلی هر الگوریتم بهتر عمل کرده و در واقع الگوریتم ۲-محلی پیشنهادی دارای بهترین عملکرد است اما در معیار پیوند هیچ‌کدام از این دو گزاره صحیح نیست.



شکل ۸: مقایسه‌ی عملکرد نسخه‌ی ۲-محلی و ۱-محلی دو الگوریتم Two-step و Two-step with average stretch factor با معیار اقلیدسی

علی‌رغم این که الگوریتم‌های پیشنهادی در معیار اقلیدسی از الگوریتم‌های پیشین عملکرد بهتری دارند، اما بررسی شکل ۱۰ نشان می‌دهد که در معیار پیوند هیچ‌یک از آن‌ها دارای بهترین عملکرد نیست، اما می‌توان گفت عملکرد برخی از الگوریتم‌ها از جمله Two-step، MixedChordArc، Apex-angle در هر دو معیار توسط یک یا چند مورد از الگوریتم‌های پیشنهادی بهبود یافته است.

- [7] X.-y. Li, I. Stojmenovic, and Y. Wang, "Partial delaunay triangulation and degree limited localized blue-tooth," 05 2004.
- [8] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf *Computational Geometry: Algorithms and Applications, second ed.* Springer, 2000.
- [9] D. P. Dobkin, S. J. Friedman, and K. J. Supowit, "Delaunay graphs are almost as good as complete graphs," *Discrete & Computational Geometry*, vol.5, no.4, pp.399–407, 1990.
- [10] P. Bose, L. Devroye, M. Löffler, J. Snoeyink, and V. Verma, "The spanning ratio of the delaunay triangulation is greater than $\pi/2$," pp.165–167, 01 2009.
- [11] G. Xia, "The stretch factor of the delaunay triangulation is less than 1.998," *SIAM Journal on Computing (SICOMP)*, vol.42, no.4, pp.1620–1659, 2013.
- [12] N. Karumanchi. *Graph Algorithms, Dijkstra*. Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles, 2015.
- [13] P. Bose and P. Morin, "Online routing in triangulations," *SIAM Journal on Computing (SICOMP)*, vol.33, no.4, pp.937–951, 2004.
- [14] W. Si and A. Y. Zomaya, "New memoryless online routing algorithms for delaunay triangulations," *IEEE Transactions on Parallel and Distributed Systems*, vol.23, no.8, 2012.
- [15] N. Bonichon, P. Bose, J. D. Carufel, V. Despré, D. Hill, and M. H. M. Smid, "Improved routing on the delaunay triangulation," *In 26th Annual European Symposium on Algorithms (ESA)*, p.22:1–22:13, 2018.
- [16] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proceedings of the Canadian Conference on Computational Geometry (CCCg)*, pp.51–54.
- [17] P. Bose, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and J. I. Munro, "Online routing in convex subdivisions," *International Journal of Computational Geometry & Applications*, vol.12, no.4, pp.283–296, 2002.
- [18] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing: geographic routing with local face information," in *13TH IEEE International Conference on Network Protocols (ICNP'05)*, pp.12 pp.–158, 2005.
- [19] Jie Gao, L. J. Guibas, J. Hershberger, Li Zhang, and An Zhu, "Geometric spanners for routing in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol.23, pp.174–185, Jan 2005.

ما همچنین عملکرد الگوریتم‌های پیشنهادی را از نظر میانگین کشش میانگین در دو معیار اقلیدسی و پیوند با بهترین الگوریتم‌های مشابه موجود مقایسه کردیم. نتایج آزمایشات ما نشان می‌دهند که تمامی الگوریتم‌های ارائه شده در عین حفظ سادگی و ظرافت، در معیار اقلیدسی بهتر از سایر الگوریتم‌های مشابه عمل می‌کنند.

تمامی الگوریتم‌های پیشنهادی در معیار اقلیدسی عملکردی بهتر از الگوریتم‌های پیشین دارند اما در معیار پیوند، عملکرد برخی را بهبود داده ولی همچنان عملکرد ضعیف‌تری نسبت به الگوریتم‌های Greedy و Midpoint دارند. آن‌ها همچنین در هر دو معیار از الگوریتم Two-step که در معیار اقلیدسی دارای بهترین عملکرد بوده و MixedChordArc که دارای کم‌ترین ضریب رقابتی است، بهتر عمل می‌کنند.

نتایج آزمایشات ما نشان می‌دهد که در بین الگوریتم‌های پیشنهادی، الگوریتم ۲-محلی ارائه شده در هر دو معیار بهترین عملکرد را داشته و در بین الگوریتم‌های ۱-محلی، در معیار اقلیدسی الگوریتم Two-step with piece-wise estimate of average stretch factors و در معیار پیوند الگوریتم Two-step Compass بهترین عملکرد را دارد.

کشش‌های میانگین پایین هر یک از الگوریتم‌های پیشنهادی نشان می‌دهد که این الگوریتم‌ها در سناریوهای متفاوت در عمل کاربردی هستند. الگوریتم‌هایی که عملکرد بهتری در معیار اقلیدسی دارند برای سناریوهایی مانند رباتیک که هدف کاهش طول اقلیدسی مسیر است و الگوریتم‌هایی که عملکرد بهتری در معیار پیوند دارند برای سناریوهایی مانند شبکه‌های حس‌گر بی‌سیم، یک انتخاب مناسب و کاربردی به حساب می‌آیند.

مراجع

- [1] Xiang-Yang Li, G. Calinescu, Peng-Jun Wan, and Yu Wang, "Localized delaunay triangulation with application in ad hoc wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol.14, pp.1035–1047, Oct 2003.
- [2] D. Chen, L. Devroye, V. Dujmovic, and P. Morin, "Memoryless routing in convex subdivisions: Random walks are optimal," *CoRR*, vol.abs/0911.2484, 2009.
- [3] S. Durocher, D. G. Kirkpatrick, and L. Narayanan, "On routing with guaranteed delivery in three-dimensional ad hoc wireless networks," *Wireless Networks*, vol.16, no.1, pp.227–235, 2010.
- [4] Y. Wang and X.-Y. Li, "Efficient delaunay-based localized routing for wireless sensor networks," *International Journal of Communication Systems*, vol.20, no.7, pp.767–789, 2007.
- [5] M. B. Haider and K. Sugihara, "Almost delaunay triangulation routing in wireless sensor networks," in *2007 10th international conference on computer and information technology*, pp.1–7, Dec 2007.
- [6] D. Satyanarayana, R. SV, et al., "Constrained delaunay triangulation for ad hoc networks," *Journal of Computer Systems, Networks, and Communications*, vol.2008, 2008.