

## نمودار قطبی نقاط با قطب متحرک

\*بهرام صادقی بی غم      \*\*فاطمه ربانی

\*دانشیار، دانشکده علوم کامپیوتر و فناوری اطلاعات، دانشگاه تحصیلات تکمیلی علوم پایه زنجان  
\*\*کارشناسی ارشد، دانشکده علوم کامپیوتر و فناوری اطلاعات، دانشگاه تحصیلات تکمیلی علوم پایه زنجان  
تاریخ دریافت: ۱۳۹۸/۱۲/۱۰      تاریخ پذیرش: ۱۳۹۹/۰۶/۲۲

### چکیده

مسئله نمودار قطبی یکی از تعمیم‌های نمودار ورونوی است که در آن به جای متر اقلیدسی از مقدار زاویه برای محاسبه فاصله استفاده می‌شود. این مسئله کاربردهای زیادی در پردازش تصویر، مخابرات و مباحث مربوط به آنتن، رؤیت‌پذیری و مسیریابی ربات دارد. در سالهای اخیر دو نوع نمودار قطبی مطرح شده و برای انواع سایت‌ها الگوریتم‌های مناسبی ارائه شده است. همچنین روی همین مسائل با داده‌های جنبشی و حالات پویا الگوریتم‌هایی ارائه شده است. در این مقاله قطب به عنوان ناظر متحرک در نظر گرفته شده و الگوریتمی ارائه می‌شود که مسئله بازسازی نمودار قطبی با قطب نزدیک را به صورت کارا و در زمان خطی حل می‌کند. در این حالت زمان پیش‌پردازش الگوریتم  $O(n^4 \log_2 n)$  و زمان باز رسم نمودار در هر حرکت متوالی قطب برابر با  $O(\log n + k)$  است که در آن  $k$  تعداد سایت‌های درون ناحیه  $T$  است که احتمال تغییر در آنها وجود دارد.

**کلمات کلیدی:** نمودار قطبی، نمودار ورونوی، مخابرات، آنتن، زاویه قطبی، رویت‌پذیری

### ۱- مقدمه

پردازش برای بازسازی نمودار است. هنگامی که قطب به عنوان یک ناظر در نظر گرفته شود، این مسئله کاربردهای زیادی در پردازش تصاویر خواهد داشت. افزایش صفحه بعنوان پیش‌پردازش، همواره باعث افزایش سرعت در اجرای برنامه‌های آنلاین می‌شود که سرعت در آنها از اهمیت خاصی برخوردار است. نمونه‌ای از این کاربردها را میتوان در بازی‌های کامپیوتری دید. هنگامی که شی متحرکی (در اینجا منبع نور بعنوان روشنایی) لابلای موانع ثابت در حرکت است صحنه‌ای که در اثر سایه‌ها تولید می‌شود در هر لحظه متفاوت است که نیاز به بازسازی دارد. بدون این پیش‌پردازش و با استفاده از الگوریتم‌های قبلی مانند الگوریتم نقاش، محاسبه‌ی ناحیه‌ها زمان زیادی صرف می‌کند. با استفاده از الگوریتم ارایه شده و پس از یک بار پیش‌پردازش میتوان تاثیرات نور و سایه در محیط را بسیار سریع تر محاسبه کرد. در این نوع کاربرد فرض کنید قطب نوری به

یکی از مسائل مهم و پر کاربرد در هندسه محاسباتی مسئله نمودار ورونوی است که از نظر نوع سایت‌ها، متر، محدودیت‌های صفحه و موانع دارای انواع تعمیم‌هاست. یکی از این تعمیم‌ها حالتی است که معیار فاصله بین نقاط و سایت‌ها زاویه است. این نوع نمودار کاربردهای مهمی در بینایی ماشین و مخابرات دارد. این مسئله در دو نوع کلی نمودار قطبی با قطب دور (کلاسیک) و قطب نزدیک معرفی شده است که در هر مورد به تعمیم‌هایی از آن نیز پرداخته شده است. همچنین به حالاتی از مسئله که سایت‌ها در حالت حذف یا اضافه شدن باشند نیز توجه شده است. این مقاله حالتی از مسئله نمودار قطبی با قطب نزدیک را بررسی می‌کند که سایت‌ها به صورت نقاطی ثابت در صفحه باشند و قطب بتواند در خارج از پوسته محدب سایت‌ها جابجا شود. هدف اصلی، ارائه روشی سریع و کارا پس از یک پیش

در نمودار قطبی با قطب نزدیک، قطب در نزدیک سایتها قرار دارد. فرض می‌کنیم که قطب در سمت چپ همه سایتها قرار داشته باشد. در واقع ورودی این مسئله مجموعه‌ای از سایتها هستند که این مجموعه را با حرف S نشان می‌دهیم و یک نقطه P به عنوان قطب. خروجی مسئله یک افراز از صفحه می‌باشد که با استفاده از آن برای هر نقطه دلخواه X در صفحه می‌توان گفت که X متعلق به ناحیه قطبی کدام سایت  $S_i$  است. شکل ۱(b) نمونه‌ای از نمودار قطبی با قطب نزدیک می‌باشد. نمودار قطبی اولین بار در سال ۱۹۹۸ در چهاردهمین کارگاه اروپایی هندسه محاسباتی در اسپانیا توسط گریم<sup>۱</sup> و همکارانش معرفی شد [۱]. آن‌ها ضمن معرفی نمودار قطبی به ارائه الگوریتم رسم آن نیز پرداختند. همچنین نمودار قطبی را در مورد پاره‌خطها، چندضلعی‌ها و دوایر گسترش دادند و به کاربردهای این نمودار در مسائل پوسته محدب<sup>۲</sup> و رؤیت‌پذیری<sup>۳</sup> اشاره کردند [۲].

در سال ۲۰۰۳، ارتگا<sup>۴</sup> و همکارانش در مورد کاربرد نمودار قطبی در مسائل برنامه‌ریزی حرکت<sup>۵</sup> و مسائل رؤیت‌پذیری بحث نمودند [۳] و الگوریتم‌هایی را برای حل این مسائل ارائه کردند که از یک پیش‌پردازش زاویه‌ای بر پایه نمودار قطبی استفاده می‌کرد.

دوگان نمودار قطبی توسط صادقی و همکارانش معرفی و الگوریتمی برای محاسبه آن ارائه شد [۴]. این الگوریتم دوگان را در زمان بهینه  $\theta(n \log n)$  محاسبه می‌کند. در سال ۲۰۱۰ ارتگا و همکارانش این مسئله را به طور گسترده مورد بررسی قرار دادند و تحت عنوان یک راه حل برای مسئله مسیریابی با استفاده از نمودار قطبی به عنوان پیش‌پردازش، مقاله‌ای به چاپ رساندند [۵]. آنها در این مقاله الگوریتمی برای مسئله مسیریابی ارائه کردند که با استفاده از نمودار قطبی صفحه را به چندین ناحیه تقسیم بندی می‌کند. همچنین در سال ۲۰۱۳ ارتگا و همکارانش به کاربرد نمودار قطبی در مسائل رؤیت‌پذیری پرداختند [۶]. در الگوریتم بیان شده ابتدا چهار نمودار قطبی به عنوان پیش‌پردازش محاسبه و سپس در گام‌های الگوریتم از این چهار نمودار با

رنگ خاص سانس می‌کند و هر سایت نور و رنگ مخصوص به خود را دارند.

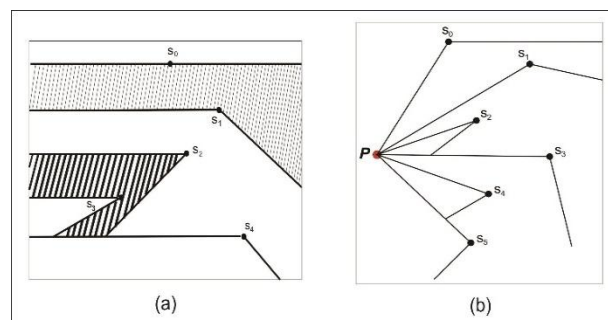
همچنین فرض کنید هر نقطه ی دلخواه از صفحه دارای رنگی است که از ترکیب رنگ سایت والد و قطب حاصل می‌شود. محاسبه ی رنگ هر پیکسل از صفحه با الگوریتم‌های رایج قبلی زمانی طولانی لازم دارد و با روش جدید و البته پس از یک بار پیش پردازش سرعت انجام اینکار زیر خطی است.

نمودار قطبی یکی از نمودارهای افراز صفحه مانند نمودار ورونوی است با این تفاوت که معیار اصلی در نمودار قطبی زاویه قطبی می‌باشد. زاویه قطبی سایت  $S_i$  نسبت به قطب P برابر است با زاویه ای که توسط نیم خط افقی رسم شده از P به سمت راست (جهت مثبت محور X ها) و پاره خط  $S_i P$  تولید می‌شود. فرض می‌شود قطب در سمت چپ تمام سایتها قرار دارد و بنابراین مقدار زاویه قطبی در بازه ی  $[-90, 90]$  قرار دارد.

ناحیه قطبی یک سایت دلخواه  $S_i$  که با  $P_S(S_i)$  نشان می‌دهیم، برابر با مکان هندسی تمام نقاطی از صفحه است که زاویه قطبی یا آنها تا سایت  $S_i$  کوچکتر از زاویه قطبی آن نقاط تا هر سایت دیگری باشد. زاویه قطبی نقطه ی  $(x, y)$  نسبت به سایت  $S_i$  یعنی  $\text{ang}(S_i)(x, y)$  زاویه ای است که بین نیمخط افقی با مبدا سایت  $S_i$  و به سمت منفی محور X و خط واصل بین سایت  $S_i$  و نقطه ی  $(x, y)$  است. ناحیه قطبی را می‌توان به صورت فرمول ۱ نشان داد:

$$P_S(S_i) = \{(x, y) \in E^2 \mid \text{ang}(S_i)(x, y) < \text{ang}(S_j)(x, y), \forall j \neq i\} \quad (1)$$

تاکنون دو نوع نمودار قطبی تعریف شده است. نمودار قطبی با قطب در بینهایت و نمودار قطبی باقطب نزدیک. در نمودار قطبی با قطب در بینهایت، قطب در دور از سایتها و در بینهایت قرار دارد. شکل ۱ نمونه‌ای از نمودار قطبی با قطب در بینهایت را نشان می‌دهد.



شکل ۱ (a) نمودار قطبی با قطب دور، (b) نمودار قطبی با قطب نزدیک

Grima<sup>۱</sup>  
Convex hull<sup>۲</sup>  
Visibility<sup>۳</sup>  
Ortega<sup>۴</sup>  
Motion planning<sup>۵</sup>

پژوهش با استفاده از ساختمان داده  $^Y$ DCEL، مسئله با زمان پیش‌پردازش  $O(n^4 \log_2 n)$  و نیز با بروز کردن نمودار در زمان  $O(\log n + k)$  حل شده است. در ادامه به معرفی نمودار قطبی با قطب نزدیک پرداخته می‌شود و سپس حالتی را مورد بررسی قرار خواهیم داد که قطب میتواند به طور پیوسته و البته خارج از پوسته محدب سایت‌ها در صفحه حرکت کند. در بخش آخر نیز تحلیل نهایی از الگوریتم ارائه شده برای به روز رسانی نمودار قطبی با قطب نزدیک متحرک، صورت می‌گیرد.

## ۲- نمودار قطبی با قطب نزدیک متحرک برای نقاط

فرض کنید مجموعه  $S$  شامل  $n$  نقطه به عنوان سایت در صفحه و قطب  $P$  در نزدیک سایت‌ها داده شده است. همچنین فرض کنید که نمودار قطبی مجموعه  $S$  و قطب  $P$  با نام  $NPPD(S)$  با استفاده از الگوریتم افزایشی بیان شده در [۸] محاسبه شده و سپس قطب  $P$  شروع به حرکت پیوسته در صفحه می‌کند. با حرکت قطب در صفحه، نمودار قطبی دچار تغییر خواهد شد. باید الگوریتمی ارائه شود که  $NPPD(S)$  را در کمترین زمان ممکن بروزرسانی نماید.

ابتدا باید بررسی شود که با حرکت قطب چه رویدادهایی ممکن است برای نمودار قطبی رخ دهد و  $NPPD(S)$  تحت چه شرایطی دست خوش تغییرات می‌گردد.

اگر به ازای هر سایت  $S_i \in S$  که پیش از این در ناحیه قطبی سایت  $S_k \in S$  بوده است (یا به عبارت دیگر  $S_i \in P_S(S_k)$ ) همچنان سایت  $S_i$  در ناحیه قطبی سایت  $S_k$  بماند و عضو ناحیه قطبی سایت دیگری نشود، می‌گوییم توپولوژی نمودار  $NPPD$  دچار تغییر نشده است. بنابراین هر گاه والد یکی از سایت‌ها تغییر کند، توپولوژی نمودار تغییر خواهد کرد و لازم است نمودار در زمان سریع و قابل قبول ترمیم شود که در واقع هدف اصلی مقاله می‌باشد.

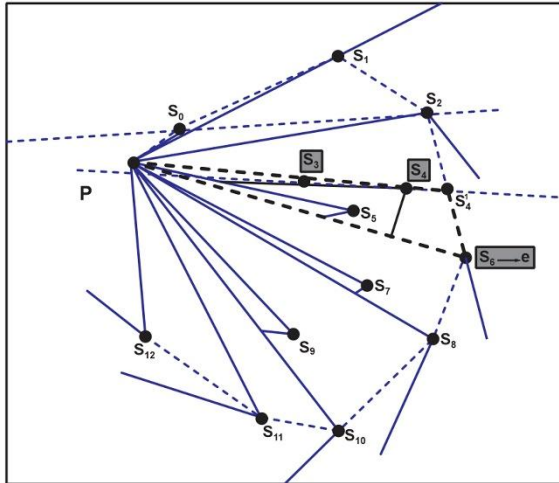
می‌توان گفت که در الگوریتم رسم نمودار  $NPPD$  هر تغییر در لیست مرتب شده ی سایت‌ها بر اساس زاویه قطبی، باعث تغییر در توپولوژی نمودار  $NPPD$  خواهد شد.

فرض کنید قطب به طور پیوسته و البته خارج از پوسته محدب سایت‌ها در حال حرکت در صفحه است، با توجه به نکته قبلی، کافی است که در هر لحظه لیست مرتب شده سایت‌ها بر اساس زاویه قطبی بررسی شود، اگر این لیست

توجه به جهت مشخص، استفاده شده است. در سال ۲۰۰۷ نوع دیگری از نمودار قطبی توسط صادقی و همکارانش معرفی شد که در آن قطب در نزدیک سایت‌ها قرار داشت [۷]. به این نوع از نمودار قطبی، نمودار قطبی با قطب نزدیک گفته می‌شود. این گروه علاوه بر معرفی نمودار قطبی با قطب نزدیک، در مورد دوگان این نوع نمودار قطبی نیز بحث نمودند. همچنین نمودار قطبی با قطب نزدیک را برای پاره‌خط‌ها و چندضلعی‌ها و دوائر گسترش دادند و به کاربردهای این نمودار در مسائل رؤیت پذیری و مسیریابی اشاره کردند [۸]. بعد از آن و در سال ۲۰۰۸، صادقی و همکارانش نمودار قطبی پویا را معرفی نمودند [۹]. در نمودار قطبی پویا دو موضوع مورد بررسی قرار گرفته است. یکی حالتی است که در نمودار قطبی یک سایت اضافه و یا حذف شود. نمودار قطبی موجود با توجه به این که کدام اتفاق رخ داده است، تغییر می‌کند و به روز می‌گردد. این تغییر از نمودار به گونه‌ای است که بخش زیادی از نمودار دست نخورده باقی می‌ماند و این موجب افزایش سرعت به روز رسانی نمودار می‌شود. این الگوریتم تغییرات را در زمان قابل قبولی در نمودار قطبی اعمال می‌کرد. سپس در سال ۲۰۱۰ احسانفر و همکارانش الگوریتمی ارائه کردند که اضافه یا حذف شدن سایت‌ها در نمودار قطبی پویا به صورت بهینه انجام گردد [۱۰]. نوع دیگری از نمودار قطبی پویا با عنوان نمودار قطبی اشیاء متحرک نیز توسط نوری بیگی و همکارش در سال ۲۰۰۸ مورد بررسی قرار گرفت [۱۱]. در این نوع از نمودار قطبی پویا سایت‌ها در صفحه به صورت پیوسته حرکت می‌کنند و هدف بروز رسانی بهینه نمودار قطبی مجموعه سایت‌ها می‌باشد. آنها از ساختمان داده جنبشی<sup>۶</sup> برای حل این مسئله استفاده کرده‌اند. همچنین الگوریتم بیان شده نمودار قطبی را در زمان بهینه بروز رسانی می‌نماید. در سال ۲۰۰۹ بزاز در پایان‌نامه کارشناسی ارشد خود در مورد نوع دیگری از نمودار قطبی بحث کرد که به کاربردهایی از آن اشاره دارد [۱۲]. نتایج بدست آمده در پایان‌نامه ی بزاز و سپس در پایان نامه ی فاطمه ربانی [۱۷] در خصوص نقاط در این پژوهش بررسی و به کار گرفته شده است در این نوع نمودار قطبی، قطب میتواند به طور پیوسته در صفحه و خارج از پوسته محدب سایت‌ها حرکت کند و نمودار قطبی با توجه به این حرکت به روز شود. در این

<sup>7</sup> Doubly-Connected Edges List

<sup>6</sup> Kinetic Data Structure



شکل ۳: عبور قطب از امتداد خط  $S_2S_3$

از حافظه  $O(n)$  استفاده می‌کند و هر پرس و جو از مکان‌یابی یک نقطه را در زمان  $O(\log n)$  پاسخ می‌دهد [۱۴]. در روشی که برای افزایش صفحه در مسئله نمودار قطبی با قطب نزدیک متحرک بیان شد، تعداد خطوط از مرتبه  $O(n^2)$  و تعداد یال‌های به وجود آمده  $O(n^4)$  می‌باشد. بنابراین بررسی این که قطب در کدام ناحیه قرار گرفته است، در زمان  $O(\log n)$  امکان پذیر می‌باشد [۱۵]. حال باید با استفاده از این ساختار داده و با توجه به تغییراتی که با حرکت قطب در نمودار NPPD رخ می‌دهد، زمان لازم برای بروز رسانی نمودار را تحلیل کنیم. در ادامه فرض کنید خط گذرنده از دو سایت  $S_i$  و  $S_j$  را با  $L(s_i, s_j)$  نشان می‌دهیم و هر خط صفحه را به دو نیم‌صفحه تقسیم می‌کند. واضح است که هر گاه قطب از خط  $L(s_i, s_j)$  عبور کند، ترتیب دو سایت  $S_i$  و  $S_j$  در لیست مرتب شده بر اساس زاویه قطبی، جابجا می‌شود و هر گاه قطب در درون یکی از ناحیه‌ها حرکت کند، توپولوژی نمودار NPPD تغییر نخواهد کرد و تنها زمانی نمودار NPPD دچار تغییر می‌شود که قطب از خط گذرنده بین دو سایت  $S_i$  و  $S_j$  عبور کند. در واقع با حرکت قطب در صفحه و عبور آن از یکی از خطوط مجاور رسم شده بین هر دو سایت دلخواه، تنها دو سایت متوالی در لیست مرتب شده‌ی سایت‌ها بر اساس زاویه قطبی، می‌توانند با یکدیگر جابجا شوند. ابتدا رویه محدب مجموعه سایت‌ها و قطب محاسبه می‌شود. این رویه محدب با  $CH_{S+p}$  نشان داده می‌شود و با حرکت قطب به روز رسانی می‌گردد. رسم  $CH_{S+p}$  برای اولین بار از مرتبه زمانی  $\theta(n \log n)$  می‌باشد ولی بروز کردن آن مطابق با حرکت قطب، در زمان ثابت امکان پذیر خواهد بود. در واقع حرکت پیوسته قطب و

مرتب شده دچار تغییر شده باشد، به این معنا است که توپولوژی نمودار قطبی تغییر کرده است.

بررسی لیست مرتب شده در هر لحظه کاری پرهزینه است. لذا روشی دیگر پیشنهاد می‌شود که در آن نیازی نیست که در هر لحظه لیست مرتب شده بررسی گردد. در این روش ابتدا با یک پیش پردازش، صفحه به چندین ناحیه تقسیم‌بندی می‌شود. ویژگی هر ناحیه در این تقسیم‌بندی این است که با حرکت قطب در هر یک از این ناحیه‌ها، ترتیب سایتها در لیست مرتب شده تغییر نمی‌کند و لذا توپولوژی NPPD بدون تغییر باقی می‌ماند. ویژگی دیگر این نواحی این است که با عبور قطب از مرز هر یک از آنها، دقیقاً جای دو سایت در لیست مرتب شده جابجا خواهند شد. در ادامه به تشریح نحوه ی افزایش صفحه به این نواحی پرداخته می‌شود.

برای تقسیم‌بندی صفحه، خط گذرنده از هر دو سایت، در صفحه رسم می‌شود. واضح است که با این روش  $O(n^2)$  خط در صفحه رسم خواهد شد. اثبات شده است که در یک چیدمان از خطوط که شامل  $n$  خط می‌باشد،  $O(n^2)$  رأس، یال و وجه وجود دارد [۱۳]. بنابراین با وجود  $O(n^2)$  خط در صفحه،  $O(n^4)$  رأس، یال و وجه در صفحه وجود خواهد داشت که این همان پیچیدگی فضایی الگوریتم نیز است. در کلیه ی مراحل پیش پردازش و پرس و جو این مقدار فضا برای تمام ذخیره سازی‌ها کافی است. هر یک از ناحیه‌های بدست آمده، یک ناحیه محدب می‌باشد و این ناحیه‌ها یا کراندار هستند و یا یک ناحیه بدون مرز. برای ذخیره این افزایش از صفحه می‌توان از ساختار داده‌ای DCEL استفاده نمود. این ساختمان داده برای نگهداری تقسیم‌بندی‌هایی از صفحه به کار می‌رود که تمامی نواحی کراندار و محدود شده باشند. پس از معرفی ساختمان داده مناسب برای ذخیره سازی افزایش صفحه باید دید با حرکت قطب چه رویدادهایی برای نمودار قطبی رخ خواهد داد. قبل از این که قطب شروع به حرکت کند، لازم است که موقعیت قطب مشخص گردد. بدین معنی که ابتدا باید دید که قطب در کدام یک از وجه‌ها قرار دارد. بررسی این که قطب در کدام یک از این وجه‌ها قرار دارد، یکی از مسائل مکان‌یابی<sup>۸</sup> می‌باشد. میدانیم که اگر  $S$  یک افزایش از صفحه شامل  $n$  یال باشد، در زمان مورد انتظار  $O(n \log n)$  می‌توان ساختمان داده ای ساخت که

<sup>8</sup> Point location

همان‌طور که در تصویر نیز مشخص شده است، وقتی قطب از خط  $L_{(S_2, S_3)}$  عبور کرد، سایت  $S_3$  از رویه محدب حذف می‌گردد. حال اگر فرض کنیم که ابتدا قطب در مکان  $P$  قرار داشته باشد، با حرکت قطب از سمت چپ خط  $L_{(S_2, S_3)}$  به سمت راست آن، سایت  $S_3$  به رویه محدب اضافه می‌گردد.

در نمودار قطبی رسم شده، هر سایت که روی  $CH_{S+P}$  قرار دارد، در ناحیه قطبی سایتی قرار دارد که در  $CH_{S+P}$  در ترتیب ساعتگرد، قبل از آن قرار دارد. به عنوان مثال می‌توان در شکل ۱- (b) گفت که سایت‌های  $S_0, S_1, S_3, S_5$  به ترتیب در ناحیه قطبی سایت قبلی خود قرار دارند. همچنین در همان شکل ۱- (b) مشخص است که اگر هر سایت  $v$  (به طور مثال سایت  $S_3$ ) روی  $CH_{S+P}$  قرار داشته باشد، این سایت مانع دید سایت‌های قبل از خود ش  $(S_0, S_1, S_2)$  نسبت به سایت‌های بعدی  $(S_4, S_5)$  است.

اکنون با توجه به مطالب بیان شده، ناحیه‌ای را تعیین می‌کنیم که با حرکت قطب، تنها نمودار قطبی سایت‌های درون این ناحیه می‌توانند دچار تغییر شوند: فرض کنید قطب ابتدا در مکان  $P$  قرار داشته باشد. حال در صدد آن هستیم که با عبور قطب از خط  $L_{(S_i, S_j)}$  به طوری که  $S_j$  در لیست مرتب شده قبل از  $S_i$  قرار بگیرد (برای مثال در شکل ۳ عبور قطب از خط  $L_{(S_3, S_4)}$ ، ناحیه‌ای از صفحه مانند  $T$  را مشخص کنیم که فقط نمودار قطبی داخل آن امکان تغییر دارد.  $T$  ناحیه مثلثی است که یکی از رئوس آن در مکان قطب ( $P$ ) قرار دارد و دو رأس دیگر آن به صورت زیر تعیین می‌شوند:

$T$  ناحیه‌ای از  $CH_{S+P}$  است که بین دو خط  $L_{(P, S_j)}$  و  $L_{(P, e)}$  قرار گرفته است که در آن سایتی است که در دو شرط زیر به طور هم‌زمان صدق کند:

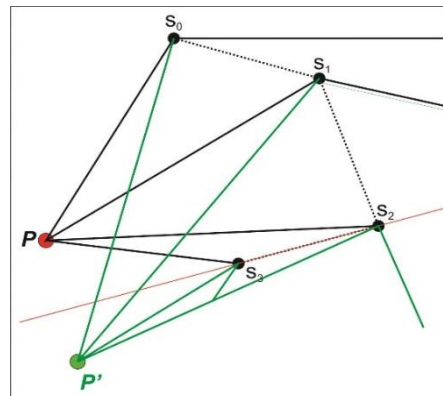
۱. در لیست مرتب شده ی سایتها، بعد از سایت‌های  $S_j$  و  $S_i$  قرار گرفته باشد. (لزومی ندارد دقیقاً سایت بعدی آنها در لیست مرتب شده باشد، بلکه کافی است که شیب  $Pe$  از شیب  $Ps_i$  و  $Ps_j$  کمتر باشد).

۲. مانند شکل ۳ روی  $CH_{S+P}$  قرار گرفته باشد.

با توجه به مطالب گفته شده، می‌توان قضیه زیر را جمع‌بندی نمود.

به روز رسانی نمودار قطبی به این صورت انجام می‌پذیرد که با عبور قطب از هر یک از خطوط مربوط به افراز صفحه، تغییرات لازم در نمودار قطبی اعمال می‌گردد. همچنین به روز رسانی  $CH_{S+P}$  نیز هنگامی انجام می‌شود که قطب از یکی از خطوط افراز صفحه عبور نموده است. به همین دلیل اگر قطب از خط  $L_{(S_i, S_j)}$  عبور کرده باشد، تنها دو سایت  $S_i$  و  $S_j$  برای بروز رسانی  $CH_{S+P}$  مورد بررسی قرار خواهند گرفت. اگر خط  $L_{(S_i, S_j)}$  یکی از اضلاع  $CH_{S+P}$  نباشد، آنگاه رویه محدب تغییری نخواهد کرد و اگر  $L_{(S_i, S_j)}$  یکی از اضلاع رویه محدب باشد، با توجه به جهت حرکت قطب یکی از دو سایت  $S_i$  و  $S_j$  از رویه محدب حذف و یا به آن اضافه می‌شوند. این یک سایت که به  $CH_{S+P}$  اضافه می‌شود یا از آن حذف می‌شود، لزوماً یکی از اعضا رویه محدب مجموعه  $S$  می‌باشد که در زمان ثابت می‌توان آن را مشخص کرد.

فرض کنید قطب از خط  $L_{(S_a, S_c)}$  عبور کرده باشد  $S_c$  و  $S_a$  از رئوس رویه محدب مجموعه  $S$  می‌باشند. اگر قطب از سمت چپ این خط به سمت راست آن حرکت کرده باشد، هرکدام از دو سایت  $S_c$  و  $S_a$  که تا کنون عضو  $CH_{S+P}$  نبوده‌اند، از این پس یکی از رئوس  $CH_{S+P}$  خواهند بود و اگر قطب از سمت راست این خط به سمت چپ آن حرکت کند، خط مماس از قطب به رویه محدب مجموعه  $S$ ، از یکی از دو سایت  $S_c$  و  $S_a$  خواهد گذشت که این سایت عضو  $CH_{S+P}$  باقی می‌ماند و سایت دیگر از  $CH_{S+P}$  حذف خواهد شد.



شکل ۲: عبور قطب از امتداد خط  $S_3S_4$

برای مثال تصویر ۲ نشان می‌دهد که قطب از خط  $L_{(S_2, S_3)}$  عبور کرده است. خط  $L_{(S_2, S_3)}$  امتداد یکی از اضلاع رویه محدب مجموعه  $\{S_0, S_1, S_2, S_3, P\}$  می‌باشد و قطب از سمت راست این خط به سمت چپ آن حرکت کرده است. با عبور قطب از سایت  $S_2$  و  $S_3$  مورد بررسی قرار می‌گیرند.



که قطب از کدام یک از این اضلاع عبور کرده است، لازم است که تقاطع بین پاره خط  $PP'$  و چندضلعی ای که قطب در آن قرار دارد، محاسبه گردد. در ادامه قضیه‌ای بیان می‌شود که نشان می‌دهد پیدا کردن این تقاطع در زمان  $O(\log n)$  امکان پذیر می‌باشد [۱۶]. حال فرض کنید قطب از خط  $L(s_n, s_m)$  عبور کرده باشد.

در گام چهارم باید ناحیه  $T$  که در قضیه ۲،۱ بیان شد، مشخص گردد و ناحیه قطبی سایت‌های درون آن به روز رسانی گردد.

فرض کنید  $k$  سایت درون ناحیه  $T$  قرار گرفته باشند. بروز رسانی نمودار قطبی این  $k$  از مرتب  $O(k)$  می‌باشد. زیرا ترتیب این سایت‌ها بر اساس زاویه قطبی در نمودار قطبی اصلی موجود است.

## ۲-۲- پیچیدگی زمانی به روز رسانی نمودار قطبی با

Algorithm 1
Input: $S$ includes $n$ Point as a set of sites, Point $P=(P_x, P_y)$ as a pole, $NPPD(S)$ , $V$ as set of vertexes in planar subdivision, $F$ as set of faces in planar subdivision, $E$ as set of edges in planner region of $P$
Output: Updated $NPPD(S)$ when pole moves to new point $P'=(P'_x, P'_y)$ that $P'$ is in the adjacent region of $P$
Step1: Determine the polar region where the pole is located
Step2: Determine the face ( $f_i \in F$ ) where the pole is located
Step3: Determine the edge of $f_i$ that is crossed by moving the Pole ( $P$ ), suppose $e_i=L(s_a, s_b)$
Step4: Specify region $T$ and update polar region of the sites inside the $T$ , suppose that number of sites inside the $T$ be $k$

### قطب نزدیک متحرک

قضیه ۲،۲ برای  $n$  سایت داده شده در صفحه و نمودار قطبی با قطب نزدیک و متحرک  $P$ ، زمان مورد نیاز برای پیش‌پردازش  $O(n^4 \log_2 n)$  و زمان بازسازی نمودار در هر حرکت متوالی قطب برابر  $O(\log n + k)$  است که در آن  $k$  تعداد سایت‌های داخل ناحیه  $T$  است.

**قضیه ۲،۱:** فرض کنیم قطب از خط  $L(s_i, s_j)$  عبور کرده باشد،  $T$  مکان هندسی ناحیه‌ای از نمودار است که در خارج از آن تغییر توپولوژی رخ نخواهد داد.

اثبات: سایت‌هایی که در لیست مرتب شده براساس زاویه قطبی قبل از  $s_i$  و  $s_j$  هستند، تحت تأثیر ناحیه قطبی این دو سایت قرار ندارند. بنابراین مرز بالایی ناحیه‌ای که تغییر توپولوژی در خارج از آن اتفاق نمی‌افتد، خط  $L(p, s_j)$  خواهد بود. همانطور که قبلاً گفته شد، سایت  $e$  مانع از دید سایت‌های قبل از خود نسبت به سایت‌هایی می‌شود که در لیست مرتب شده بعد از آن قرار دارند. بنابراین مرز پایینی مکان هندسی مورد نظر خط  $L(p, e)$  است. □

شکل ۳ نمودار قطبی را پس از عبور قطب از خط  $L(s_3, s_4)$  نشان می‌دهد. تغییرات توپولوژی فقط در مثلث  $S'_4PS_6$  رخ داده است.

## ۲-۱- الگوریتم به روز رسانی نمودار قطبی با قطب نزدیک متحرک

تصویر ۴ مراحل اجرای الگوریتم نمودار قطبی با قطب نزدیک متحرک را به صورت شبه کد نشان می‌دهد. برای محاسبه پیچیدگی زمانی بروز رسانی نمودار قطبی، ابتدا در هر گام از الگوریتم، پیچیدگی زمانی را تحلیل می‌کنیم. در گام اول مشخص می‌شود که قطب در ناحیه قطبی کدام سایت قرار گرفته است. با داشتن  $NPPD(S)$ ، در زمان  $O(\log n)$  می‌توان تعیین کرد که قطب در ناحیه قطبی کدام سایت است. فرض کنید قطب در ناحیه قطبی سایت  $s_i$  قرار داشته باشد.

در گام دوم تعیین می‌گردد که قطب در کدام وجه از تقسیم‌بندی صفحه قرار گرفته است.

طبق قضیه ای که پیش‌تر بیان شده است، با  $O(n^2)$  یال در تقسیم‌بندی صفحه، در زمان  $O(\log n)$  می‌توان تعیین نمود که قطب در کدام وجه قرار دارد [۱۵].

در گام سوم تعیین می‌شود که قطب با حرکت پیوسته خود، از کدام یک از یال‌های وجه‌ای که در آن قرار گرفته است، عبور کرده است و این یال مربوط به خط واصل کدام دو سایت می‌باشد.

افراز صفحه به روش بیان شده، موجب می‌شود که وجه‌هایی به صورت چندضلعی‌های محدب (کراندار یا نامحدود) در صفحه ایجاد شوند. این چندضلعی‌ها در بدترین حالت ممکن است  $O(n)$  ضلع داشته باشند. برای این که مشخص شود

قضیه ۲,۲ نتایج نظری به دست آمده را خلاصه سازی می کند. این نتایج نظری به راحتی می توانند در انواع کاربردها در زمینه های ردیابی اتومبیل های خودراننده [۱۸] هدایت اشیا زیر سطحی (نظیر زیر دریایی) [۱۹]. ارتباطات مغناطیسی [۲۰] و سایر کاربردها دیده شوند و بدیهی است استفاده از نتایج جدید در خصوص پیچیدگی زمانی، باعث ارتقا کیفیت و سرعت پردازش نرم افزارهای مرتبط خواهد شد.

### ۳- نتیجه گیری

مسئله اصلی که در این مقاله به عنوان پژوهش اصلی مورد توجه است، حل مسئله نمودارقطبی نقاط با قطب متحرک است که در آن قطب در نزدیکی سایتها و البته خارج پوسته محدب آنها، در حال حرکت است. الگوریتمی که در این مقاله ارائه شده است، ابتدا با یک پیش پردازش صفحه را افزایش می کند و ناحیه ای را که قطب در آن قرار گرفته است، مشخص می کند. سپس ناحیه ای را تعیین می کند که با حرکت قطب، دچار تغییرات شده است و نیاز است که این بخش از نمودار قطبی با قطب نزدیک به روز رسانی گردد. در این الگوریتم زمان پیش پردازش الگوریتم  $O(n^4 \log_2 n)$  و زمان باز رسم نمودار در هر حرکت متوالی قطب برابر با  $O(\log n + k)$  است که در آن  $k$  تعداد سایت های درون ناحیه  $T$  است که احتمال تغییر در آنها وجود دارد.

با توجه به کاربردهای بالقوه زیاد این مفهوم در بینایی ماشین و مخابرات پیشنهاد میشود علاقمندان در آینده همین مساله را برای اشیا مختلف حل کنند و کاربردهای دقیقی از آن بدهند.

برای محاسبه پیچیدگی زمان این الگوریتم دو نوع زمان مورد بررسی قرار می گیرد. یکی زمان پیش پردازش و دیگری زمان اجرا.

در زمان پیش پردازش، نمودار قطبی  $NPPD(S)$  محاسبه می گردد و افزایش صفحه انجام می شود و همچنین یک پیش پردازش نیز برای مسئله مکان یابی نیز صورت می گیرد که مرتبه زمانی تمام این موارد برابر است با:

$$O(n \log n + n^2 + n^4 \log n) = O(n^4 \log n)$$

در زمان اجرا پس از عبور قطب از هر یک از خطوط افزایش صفحه، ابتدا باید مشخص نمود که قطب از کدام خطوط افزایش صفحه عبور کرده است که با توجه به مطالب گفته شده این کار در زمان  $O(\log n)$  انجام می گیرد. سپس باید نمودار قطبی به روز رسانی گردد. که تنها کافیست نمودار قطبی در درون ناحیه  $T$  دوباره رسم شود. با فرض این که  $k$  سایت در این ناحیه وجود دارند که ترتیب آنها بر اساس زاویه قطبی مشخص است، میتوان در زمان  $O(\log n + k)$  نمودار قطبی را بروز رسانی نمود. بنابراین پیچیدگی زمان اجرا برابر است با:  $O(\log n + k)$

در ادامه و در قالب یک قضیه نتایج فوق در مورد الگوریتم و پیچیدگی های زمانی و فضایی را جمع بندی می کنیم.

قضیه ۲,۲: الگوریتم ۱ مساله ی محاسبه ی دیاگرام قطبی با قطب نزدیک و متحرک را برای سایت با صرف  $O(n^4)$  فضا و در زمان پیش پردازش  $O(n^4 \log_2 n)$  حل میکند. این زمان برای هر پرس و چو در مراحل بعدی برابر  $O(\log n + k)$  است.

Sadeghi, and Madadi, Najmeh. An optimal solution for dynamic Polar diagram. in CCCG, PP. 51–54, 2010.e

11. Beygi, Mojtaba Nouri and Ghodsi, Mohammad. Polar diagram of moving objects. In 20th Annual Canadian Conference on Computational Geometry, P. 51. Citeseer, 2008.

۱۲. بزاز، زینب، کاربرد دیاگرام ورونوی حساس به زاویه در مسایل بینایی، پایان نامه کارشناسی ارشد، دانشگاه صنعتی امیرکبیر، ۱۳۸۹

13. Aronov, Boris, Edelsbrunner, Herbert, Guibas, Leonidas J., and Sharir, Micha. The number of edges of many faces in a line segment arrangement. *Combinatorica*, 12(3):261–274, 1992.

14. De Berg, Mark, Cheong, Otfried, van Kreveld, Marc, and Overmars, Mark. *Computational geometry*. Springer Berlin Heidelberg, Berlin, third ed., 2008.

15. KirkPatrick, David. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12(1):28–35, 1983.

16. Chazelle, Bernard and Dobkin, David P. Intersection of convex objects in two and three dimensions. *Journal of the ACM (JACM)*, 34(1):1–27, 1987.

۱۷. ربانی، فاطمه، دیاگرام قطبی با قطب متحرک. پایان نامه کارشناسی ارشد، دانشگاه تحصیلات تکمیلی علوم پایه زنجان، ۱۳۹۱

18. Sun, Qinbo, et al. "Tacking Control of an Autonomous Sailboat Based on Force Polar Diagram." 2018 13th World Congress on Intelligent Control and Automation (WCICA). IEEE, 2018.

19. The magnetotelluric (MT) method is commonly used to estimate the subsurface conductivity structure.

20. Pranata, Erick, Selvi Misnia Irawati, and Sintia Windhi Niasari. "Magnetotelluric Data Analysis using Swift Skew, Bahr Skew, Polar Diagram, and Phase Tensor: a Case Study in Yellowstone, US."

#### ۴-منابع

1. Grima, CI, Márquez, A, and Ortega, L. A locus approach to angle Problems in computational geometry. In Proc. of 14th European Workshop in Computational Geometry, Barcelona, 1998.

2. Grima, CI, Márquez, A, and Ortega, L. Polar diagrams of geometric objects. In 15th European Workshop in Computational Geometry, 1999.

3. Grima, C, Márquez, A, and Ortega, L. Motion Planning and visibility Problems using Polar diagrams. In Annual conference of the European association for computer graphics, EG. Citeseer, 2003.

4. Bigham, B Sadeghi and Mohades, Ali. The dual of Polar diagrams and its extraction. In International Conference of Computational Methods in Sciences and Engineering ICCMSE, vol. 7, PP. 451–454, 2006.

5. Ortega, Lidia M, Rueda, Antonio J, and Feito, Francisco R. A solution to the Path Planning Problem using angle Pre-Processing. *Robotics and Autonomous Systems*, 58(1):27–36, 2010.

6. Ortega, Lidia and Robles-Ortega, M Dolores. Visibility resolution with Polar diagrams. *APPL. Math*, 7(5):1651–1669, 2013.

7. Bigham, B Sadeghi and Mohades, Ali. Polar diagram with respect to a near Pole. In 23rd European Workshop on Computational Geometry EWCG07, Austria, PP. 206–209. Citeseer, 2007.

8. Bigham, Bahram Sadeghi, Eskandari, Marzieh, and Tahmasbi, Maryam. Near-Pole Polar diagram of objects and duality. *Journal of Computational Science*, 3(3):127–131, 2012.

9. Sadeghi Bigham, Bahram, Mohades, Ali, and Ortega, Lidia. Dynamic Polar diagram. *Information Processing Letters*, 109(2):142–146, 2008.

10. Ehsanfar, Ebrahim, Bigham, Bahram