

ارائه یک الگوریتم متمرکز مبتنی بر نمودار و رونوی برای تشخیص حفره در شبکه‌های حسگر بی‌سیم

منصور داودی منفرد^{۱*}، اسماعیل دلفراز پهلوانلو^۲، سجاد قبادی بابی^۳، مهتاب ماسوری^۴

۱- استادیار، ۲ و ۳ و ۴- دانشجوی کارشناسی ارشد، دانشکده علوم رایانه و فناوری اطلاعات، دانشگاه تحصیلات تکمیلی علوم پایه زنجان

(دریافت: ۹۵/۰۸/۲۲، پذیرش: ۹۵/۱۱/۲۵)

چکیده

یکی از چالش‌های مهم در شبکه‌های حسگر بی‌سیم، تشخیص و محاسبه مساحت حفره‌ها در محیط می‌باشد. حفره‌ها به دلایل مختلفی از جمله مرگ تصادفی حسگرها، انفجار در محیط و یا تمام شدن انرژی حسگرها در شبکه‌های حسگر بی‌سیم ایجاد می‌گردند و وظیفه نظارتی شبکه‌های حسگر بی‌سیم را مختل می‌کنند. زمانی که حسگرها به صورت تصادفی در مناطقی مانند جنگل‌های متراکم و زمین‌های ناهموار قرار می‌گیرند، تشخیص حفره به صورت دستی در محیط امکان‌پذیر نیست. به همین دلایل هدف ما در این مقاله ارائه یک الگوریتم متمرکز برای تشخیص و محاسبه مساحت حفره‌ها در محیط، با استفاده از رویکردهای هندسه محاسباتی است. ما در این مقاله مسئله تشخیص حفره را با در نظر گرفتن دو حالت بررسی می‌کنیم: ۱- هنگامی که محیط شامل مانع نباشد و فقط مجموعه‌ای از حسگرها با شعاع متفاوت در محیط قرار گرفته باشند. ۲- علاوه بر حسگرهای موجود در محیط، ناحیه موردنظر شامل مجموعه‌ای از موانع نیز باشد. در هر دو حالت الگوریتم‌های کارآیی ارائه داده و با استفاده از رویکردهای هندسه محاسباتی بعد از تشخیص حفره‌های موجود در محیط، مساحت هر حفره را همراه با یال‌های مرزی به صورت دقیق گزارش می‌کنیم. پیچیدگی الگوریتم در حالت بدون مانع $O(n \log^2 n)$ و در حالت با مانع $O(n \log^2 n + nm^2)$ است و نتایج حاصل از شبیه‌سازی نشان می‌دهد که الگوریتم‌های ارائه شده حفره‌های موجود در محیط را به درستی تشخیص می‌دهند. در نتیجه، ما الگوریتم ارائه شده در حالت (۱) را با یکی از الگوریتم‌های جدید ارائه شده مقایسه می‌کنیم. نتایج حاصل از شبیه‌سازی کارا و دقیق بودن الگوریتم ما را نشان می‌دهد.

واژه‌های کلیدی: شبکه‌های حسگر بی‌سیم، تشخیص حفره، نمودار و رونوی و زنده‌دار

۱- مقدمه

حسگر بی‌سیم در بسیاری از زمینه‌های نظامی و غیرنظامی از قبیل: نظارت منطقه جنگی^۵، تخمین و بررسی خسارات نبرد، حفاظت از محیط‌زیست، بررسی جابجایی حیوانات، تشخیص و نگاشت پیچیدگی‌های زیست‌شناسی^۶ و کنترل ترافیک استفاده می‌شود [۱-۴].

صرف انرژی اهمیت حیاتی در شبکه‌های حسگر بی‌سیم دارد و به طور معمول انرژی حسگرها توسط باتری که بر روی آنها نصب شده است تأمین می‌شود و بعد از استقرار این حسگرها در محیط، هیچ مراقبتی از آنها صورت نمی‌گیرد بنابراین از این حسگرها انتظار می‌رود تا ماه‌ها و حتی سال‌ها بدون تعویض باتری کار کنند. در کاربردهای خاصی از شبکه‌های حسگر بی‌سیم، مکان حسگرها از قبل می‌تواند مشخص باشد و می‌توان حسگرها

شبکه‌های حسگر بی‌سیم^۲ شامل مجموعه‌ای از حسگرها هستند که در یک ناحیه جغرافیایی قرار داده شده‌اند. هر کدام از این حسگرها توانایی ارتباط بی‌سیم با یکدیگر و همچنین انجام یک وظیفه خاص مانند ردیابی هدف^۳ که با کمترین تعداد حسگرها در یک محیط هدف خاصی را به صورت مشارکتی ردیابی می‌کنند، دارند [۱-۲]. یک حسگر به تنهایی یک منبع محدود با قدرت کم پردازش سیگنال و توانایی‌های محاسباتی و ارتباطی است که فقط می‌تواند ناحیه محدودی از محیط را حس^۴ کند. بنابراین زمانی که گروهی از حسگرها با هم مشارکت می‌کنند، می‌توانند وظیفه کارآمدی را انجام دهند. امروزه از شبکه‌های

* رایانامه نویسنده مسئول: mdmonfared@iasbs.ac.ir

موضوع‌های کلیدی در شبکه‌های حسگر بی‌سیم است و تشخیص حفره با گزارش مرز آن صورت می‌گیرد. تشخیص مرز دقیق حفره برای بهبود عملکرد شبکه‌های حسگر بی‌سیم از اهمیت بسیاری برخوردار است زیرا: ۱- موجب سهولت بخشیدن به عملیات پوشش محیط برای رسیدن به پوشش بیشینه در محیط می‌شود. ۲- کمک کردن به تخمین مقدار داده‌های حس شده، درون حفره‌هایی که داده‌های حس شده ویژگی‌های خاصی دارند. با توجه به اینکه حسگرهایی که قبلاً حفره را پوشش داده بودند وظیفه حس کردن داده‌های ناحیه حفره را بر عهده داشتند، ولی با ایجاد حفره در محیط وظیفه حس کردن این داده‌ها را حسگرهای مرزی حفره انجام می‌دهند و بر اساس مقدار داده‌هایی که این حسگرهای مرزی قبل از ایجاد حفره حس می‌کردند و مقدار داده‌های حس شده بعد از ایجاد حفره، می‌توان اندازه مقدار داده‌های حس شده درون حفره را توسط حسگرهای مرزی حفره تخمین زد [۱۰]. ۳- کمک کردن برای تشخیص عواملی که باعث انتشار حفره در محیط می‌شوند.

در قسمت دوم مقاله کارهای مرتبط را بیان می‌کنیم. در قسمت سوم مقاله صورت مسئله و تعاریف اولیه را ذکر می‌کنیم. در قسمت چهارم به ارائه الگوریتم در حالتی که محیط بدون مانع است می‌پردازیم و در قسمت پنجم به ارائه الگوریتم در حالتی که محیط شامل مانع است و تحلیل الگوریتم ارائه شده در این حالت می‌پردازیم و در فصل ششم نتیجه‌گیری و کارهای آتی را بیان می‌کنیم.

۲- کارهای مرتبط

از مهم‌ترین ابزارهای هندسه محاسباتی که می‌توان برای تشخیص حفره و تخمین اندازه آن استفاده کرد نمودار ورونوی^۶ و مثلث‌بندی^۷ است [۱۱-۱۲]. در رویکرد نمودار ورونوی، ابتدا نمودار ورونوی به ازای حسگرهای ثابت^۸ موجود در محیط ساخته می‌شود. فرض کنید هر حسگر مختصات مکان خود و همسایه‌های خود را می‌داند. Wang و همکارانش [۱۳] یک الگوریتم ساخت محلی برای ساخت نمودار ورونوی ارائه دادند که هر حسگر سلول ورونوی خود را بر اساس همسایه‌های $1-hop$ (حسگرهایی که فاصله آنها از حسگر موردنظر حداکثر دو برابر شعاع آن حسگر

را به صورت دستی^۱ در محیط قرار داد. اما در برخی از موارد مخصوصاً مناطق حادثه‌دیده (آلوده به مواد شیمیایی، زلزله و ...) نمی‌توان حسگرها را به صورت دقیق در یک مکان مشخص شده از قبل قرار داد به همین دلیل از راهبرد استقرار تصادفی^۲ استفاده می‌شود. با توجه به اینکه در راهبرد استقرار تصادفی، حسگرها به صورت تصادفی در محیط قرار داده می‌شوند بنابراین توزیع حسگرها در محیط به صورت یکنواخت نمی‌باشد و نمی‌توان نظارت کامل بر محیط را اعمال کرد [۵].

برای این که بتوانیم یک محیط را پوشش دهیم و به اطلاعات آن دسترسی داشته باشیم باید هر نقطه در محیط حداقل توسط یک حسگر پوشش داده شده باشد. اگر در یک محیط نقطه‌ای توسط هیچ حسگری پوشش داده نشده باشد حفره^۳ ایجاد می‌شود. به دلیل مرگ تصادفی حسگرها، انفجار در محیط و یا تمام شدن انرژی حسگرها پوشش کامل یک محیط دشوار است [۱]. همچنین وقتی حسگرها به صورت تصادفی در محیط قرار داده می‌شوند احتمال دارد تراکم حسگرها در یک ناحیه از محیط زیاد و در یک ناحیه دیگر کم باشد که این باعث می‌شود یک ناحیه از محیط توسط چند حسگر پوشش داده شود ولی ناحیه دیگر از محیط توسط هیچ حسگری پوشش داده نشود که این عامل موجب بوجود آمدن حفره در محیط می‌شود. بنابراین استقرار تصادفی حسگرها در محیط و یا فعالیت‌هایی که در هر لحظه از زمان داخل شبکه انجام می‌گیرد (مانند تمام شدن عمر باتری حسگرها، جابجایی حسگرها و ...) جزء دلایل اصلی به وجود آمدن حفره در یک محیط می‌باشند [۶].

وجود حفره در محیط می‌تواند بر کل فعالیت شبکه تأثیر گذاشته و عملکرد شبکه حسگر بی‌سیم را در محیط کاهش دهد، از جمله این که قابلیت اطمینان^۴ داده را کاهش می‌دهد، هم‌بندی سراسری شبکه را تغییر می‌دهد [۷]، موجب تشدید بار اطلاعات انتقال داده شده توسط حسگرهای مرزی^۵ حفره می‌شود [۸]. از طرف دیگر وجود حفره می‌تواند بیانگر ویژگی‌های محیط از جمله وجود دریاچه در محیط و یا وقوع پدیده‌هایی مانند آتش‌سوزی و سیل که مانع از قرار گرفتن حسگرهای جدید در آن ناحیه می‌شوند، باشد [۹]. بنابراین تشخیص حفره یکی از

1- Manually

2- Random Deployment Strategy

3- Hole

4- Reliability

5- Boundary

6- Voronoi Diagram

7- Triangulation

8- Static Sensors

بیشینه تعداد حسگرهای اضافی^۸ در یک حفره بزرگ^۹ است. پیچیدگی این الگوریتم به اندازه شبکه بستگی ندارد اما وقتی شبکه متراکم باشد الگوریتم برای تشخیص حفره با مشکلات زیادی روبرو می‌شود.

Xiaoyun Li و همکارانش [۱۹-۲۰] یک الگوریتم توزیع شده با نام $3Mesh$ ^{۱۰} ارائه دادند که این الگوریتم بر اساس مدل ساده ارتباطی حسگرها با هم که به صورت حلقه می‌باشد، حلقه $3Mesh$ نامیده شده است. این الگوریتم از هندسه محاسباتی و نظریه گراف^{۱۱} استفاده کرده و بر اساس همسایگی‌های هر حسگر و بدون در اختیار داشتن مختصات مکان حسگرها، حفره موجود در محیط را تشخیص می‌دهد. این الگوریتم مشابه [۱۸] دارای پیچیدگی زمانی $O(ht^2)$ می‌باشد.

Zhiping و همکارانش [۳] یک الگوریتم غیرمتمرکز^{۱۲} مبتنی بر حسگر، برای تشخیص حفره ارائه دادند که این الگوریتم بر روی حسگرها اجرا شده و نقاط بحرانی مرزی^{۱۳} حفره‌ها را به ازای تمامی حسگرها، (بدون در نظر گرفتن مختصات مکان حسگرها) به کمک همسایگی‌های هر حسگر به دست می‌آورد و بعد از محاسبه نقاط بحرانی مرزی حفره‌ها به ازای هر حسگر با حرکت روی مرز هر حسگر، یال‌های مرزی حفره‌ها را به دست می‌آورد. این الگوریتم دارای پیچیدگی زمانی $O(adn)$ می‌باشد که n نشان‌دهنده تعداد حسگرهای موجود در محیط است و d نشان‌دهنده بیشینه تعداد حسگرهای همسایه با یک حسگر است $(d \leq n)$ و a نشان‌دهنده بیشینه تعداد نقاط بحرانی مرزی در شبکه می‌باشد. همچنین Sahoo و همکارانش [۲۱] یک الگوریتم توزیع شده برای تشخیص حفره ارائه دادند که بعد از محاسبه نقاط شکاف مرزی^{۱۴} حسگرها، یال‌های مرزی حفره‌ها را به دست می‌آورد. این الگوریتم دارای پیچیدگی زمانی $O(n^3)$ می‌باشد.

Wei Li و Wei Zhang [۱] الگوریتمی با استفاده از ویژگی‌های دایره خالی ارائه دادند که بر این اساس تصمیم می‌گیرد که آیا حفره‌ای در محیط وجود دارد یا خیر و در صورت وجود حفره، حسگرهایی که مرز حفره را تشکیل می‌دهند گزارش

می‌باشد) خود می‌سازد و به کمک اطلاعات همسایه‌های ورونوی خود، درمورد وجود یا عدم وجود حفره در آن سلول ورونوی تصمیم‌گیری می‌کند. Ghosh [۵]، Wang و همکارانش [۱۴] بعد از ساخت نمودار ورونوی در یک محیط شامل مجموعه‌ای از حسگرهای ثابت که به صورت تصادفی در محیط قرار داده شده‌اند، با یک روش توزیع شده^۱ اندازه دقیق هر حفره را در هر سلول ورونوی بر اساس حسگر ثابت موجود در آن سلول ورونوی و به کمک همسایه‌های آن به دست آوردند. در روش مثلث‌بندی، یک محیط شامل مجموعه‌ای از حسگرهای ثابت در نظر گرفته شده است [۱۵] و بر اساس مرکز حسگرها یک مثلث به ازای هر سه حسگر همسایه ساخته می‌شود. در این رویکرد بر اساس مساحت هر مثلث و مساحت پوشش داده شده در هر مثلث توسط هر حسگر، اندازه حفره در هر مثلث به دست می‌آید.

در رویکرد آماری^۲ برای تشخیص حفره، Fekete و همکارانش [۱۶] یک الگوریتم تشخیص مرزی برای حسگرهایی که به صورت تصادفی در یک ناحیه هندسی قرار داده شده‌اند، ارائه دادند. که ایده اصلی آنها بر این اساس است که حسگرهایی که مرز حفره را تشکیل می‌دهند دارای میانگین درجه پوشش^۳ کمتری نسبت به حسگرهایی که درون شبکه هستند می‌باشند. Ghrist و همکارانش [۱۷] و V. De Silva و همکارانش [۱۸] مسئله تشخیص حفره در محیط را با روش جبری و با استفاده از نظریه همسانی^۴ مورد مطالعه قرار دادند که در آن هیچ اطلاعاتی در مورد مکان حسگرها در دسترس نیست. در [۱۷] یک الگوریتم متمرکز^۵ بر اساس مدل محاسباتی ارائه شده است که این الگوریتم بر اساس ارتباطی که حسگرها با هم دارند حفره موجود در محیط را تشخیص می‌دهد و دارای پیچیدگی زمانی $O(n^5)$ است که n نشان‌دهنده تعداد حسگرهای موجود در محیط است. روش ارائه شده در این مقاله هیچ تضمینی در مورد تشخیص دقیق مرز حفره ارائه نمی‌دهد. همچنین الگوریتم ارائه شده در [۱۸] برای تشخیص حفره دارای پیچیدگی زمانی $O(ht^2)$ می‌باشد که t نشان‌دهنده بیشینه تعداد حسگرهای فعالی^۶ است که با یک حسگر در محیط همپوشانی^۷ دارند و h نشان‌دهنده

7- Overlap

8- Redundant

9- Large Hole

10- Triangle Mesh Self-Healing

11- Graph Theory

12- Decentralized

13- Boundary Critical Points

14- Boundary Gap Points

1- Distributed

2- Statistical Approach

3- Coverage Degree

4- Homology Theory

5- Centralized

6- Active

بر پایه دانش نویسندگان این مقاله، تاکنون هیچ مطالعه‌ای با هدف تشخیص حفره در یک محیط با مانع وقتی که شعاع حسگرها متفاوت باشند صورت نگرفته است. بیشتر الگوریتم‌های پیشین مسئله تشخیص حفره، برای حالتی در نظر گرفته شده‌اند که محیط بدون مانع بوده و حسگرها دارای شعاع یکسانی هستند که این الگوریتم‌ها دارای پیچیدگی زمانی بالایی هستند و نقص عمده‌ای که بیشتر این الگوریتم‌ها دارند این است که در بسیاری از موارد پس از تشخیص وجود حفره در محیط، مرز اطراف حفره را به صورت دقیق گزارش نمی‌کنند و با خطا همراه هستند. همچنین این الگوریتم‌ها برای حالتی که محیط شامل مانع باشد و حسگرها دارای شعاع متفاوت باشند کارا نمی‌باشند و اگر بخواهیم آنها را برای چنین حالت‌هایی نیز بسط دهیم، باید تغییرات زیادی در الگوریتم ایجاد کنیم که این کار باعث می‌شود الگوریتم کارایی قبلی خود را نداشته باشد و همچنین زمان اجرای الگوریتم افزایش یابد. در نتیجه در این مقاله به دنبال الگوریتم کارایی با پیچیدگی زمانی قابل قبول برای تشخیص حفره در یک محیط با مانع، وقتی که شعاع حسگرها متفاوت است، هستیم. الگوریتم ارائه شده در این مقاله علاوه بر این که حفره‌های موجود در یک محیط بدون مانع را همراه با یال‌های مرزی آن و مساحت هر حفره به درستی گزارش می‌دهد، جواب کارایی هم به حالتی که محیط شامل مانع باشد می‌دهد و دارای پیچیدگی زمانی کمتری نسبت به الگوریتم‌های پیشین تشخیص حفره می‌باشد. ما در این مقاله مسئله تشخیص حفره در یک محیط را در دو حالت بررسی می‌کنیم: ۱- هنگامی که محیط شامل مانع نباشد و فقط مجموعه‌ای از حسگرها با شعاع متفاوت در محیط قرار گرفته باشند. ۲- علاوه بر حسگرهای موجود در محیط، ناحیه موردنظر شامل مجموعه‌ای از موانع نیز باشد. در حالت اول یک الگوریتم متمرکز با استفاده از نمودار ورونوی و زندهار جمعی^۵ ارائه می‌دهیم که قادر است پس از تشخیص حفره‌های موجود در محیط، یال‌های مرزی و همچنین مساحت دقیق هر حفره را به صورت دقیق گزارش کند. این الگوریتم در حالت بدون مانع دارای پیچیدگی زمانی $O(n \log^2 n)$ است که n نشان‌دهنده تعداد حسگرهای موجود در محیط است. در حالت دوم هم با استفاده از بسط الگوریتم در حالت بدون مانع به حالت با مانع، الگوریتمی با پیچیدگی زمانی $O(n \log^2 n + nm^2)$ ارائه می‌دهیم که m نشان‌دهنده تعداد رؤس موانع موجود در محیط

می‌دهد. در این الگوریتم، ابتدا مثلث‌بندی دلونی^۱ حسگرها برای نشان دادن توپولوژی شبکه حسگر بی‌سیم به دست می‌آید و با توجه به این که هر دایره محیطی^۲ یک مثلث دلونی، شامل هیچ رأس از بقیه مثلث‌ها نمی‌باشد، درباره وجود حفره در محیط و همچنین حسگرهای مرزی حفره‌ها بحث می‌کند. Ma و همکارانش [۲۲] یک الگوریتم توزیع شده با استفاده از رویکرد هندسه محاسباتی برای تشخیص حفره در یک محیط ارائه داده‌اند که در این الگوریتم هر حسگر بر اساس همسایگی‌های $hop - 1$ و $hop - 2$ (حسگرهایی که فاصله آنها از حسگر موردنظر حداکثر سه برابر شعاع آن حسگر می‌باشد) خود، حفره موجود در محیط و همچنین حسگرهایی که مرز حفره را تشکیل می‌دهند به دست می‌آورد. این الگوریتم دارای پیچیدگی زمانی $O(nd)$ می‌باشد که d نشان‌دهنده بیشینه تعداد حسگرهای همسایه با یک حسگر است. $Chi Zhang$ و همکارانش [۲۳] یک پروتکل توزیع شده ارائه دادند که به هر کدام از حسگرهای موجود در محیط این امکان را می‌دهد تا بر اساس همسایه‌های $hop - 1$ و اطلاعات سلول ورونوی خود، مشخص کنند که جزء مرز حفره هستند یا خیر.

$B. kun$ و همکارانش [۲۴] الگوریتمی برای تشخیص توپولوژی حفره بدون اطلاعات محلی ارائه دادند که این الگوریتم توزیع شده بر اساس توپولوژی گرافی که از ارتباط حسگرها ساخته شده است کار می‌کند. در این الگوریتم هر حسگر با مقایسه درجه خود با میانگین درجه همسایه‌های $hop - 2$ ، درمورد اینکه جزء مرز حفره است یا خیر تصمیم‌گیری می‌کند. این الگوریتم نمی‌تواند تمامی حسگرهای مرزی حفره را به صورت دقیق به دست آورد و برای شبکه‌های حسگر بی‌سیم گسترده با تعداد حفره‌های کم موجود در محیط، کارا نمی‌باشد. $Funke$ و همکارانش مسئله تشخیص حفره را در [۲۶-۲۵] مورد مطالعه قرار دادند و یک الگوریتم اکتشافی^۳ بر اساس توپولوژی ارتباطی بین حسگرها [۲۵] و یک الگوریتم برای تشخیص حفره بر اساس گراف ارتباطی بین حسگرها و مدل گراف دیسک واحد^۴ ارائه دادند [۲۶]. این الگوریتم برای شبکه‌های مترام کارا می‌باشد ولی وقتی تراکم شبکه کاهش یابد عملکرد الگوریتم هم کاهش می‌یابد.

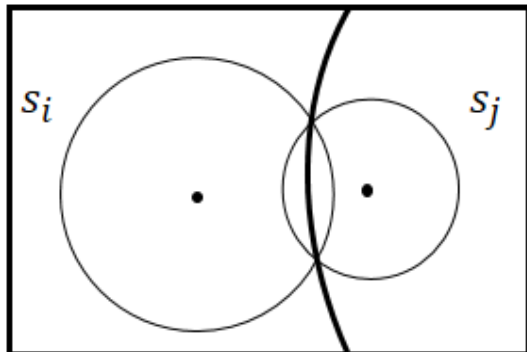
- 1- Delaunay Triangulation
- 2- Circumcircle
- 3- Heuristic
- 4- Unit Disk Graph Model

5- Additively Weighted Voronoi Diagram

می‌باشد.

۴- تشخیص حفره در یک محیط بدون مانع

ابتدا مسئله تشخیص حفره را در حالتی که محیط بدون مانع است، بررسی می‌کنیم. در این مرحله، حفره‌های تشکیل شده در محیط را با توجه به موقعیت حسگرهای موجود به دست می‌آوریم. برای تشخیص حفره از نمودار ورونوی وزندار جمعی [۲۷] استفاده می‌کنیم. به صورت غیررسمی، نمودار ورونوی وزندار جمعی مجموعه‌ای از نقاط وزندار یا دیسک (حسگر)، محیط را به مجموعه‌ای از سلول‌هایی تقسیم می‌کند که هر نقطه داخل هر سلول ورونوی کمترین فاصله را از مرز نقطه وزندار یا دیسک (حسگر) مربوط به آن سلول ورونوی نسبت به مرز نقطه وزندار یا دیسک (حسگر) سلول‌های ورونوی دیگری دارد و همچنین هر حسگر وظیفه پوشش ناحیه مربوط به سلول ورونوی خود را دارد. به بیان دیگر، اگر $S = \{s_1, s_2, \dots, s_n\}$ نشان‌دهنده مجموعه حسگرهای موجود در محیط باشد که $s_i = (c_i, r_i)$ و $c_i = (x_i, y_i)$ مرکز و r_i شعاع حسگر s_i می‌باشد، نمودار ورونوی وزندار جمعی به ازای دو حسگر s_i, s_j به صورت زیر تعریف می‌شود:



شکل (۱): نمودار ورونوی وزندار جمعی

بنابراین نمودار ورونوی وزندار جمعی به ازای مجموعه حسگرهای $S = \{s_1, s_2, \dots, s_n\}$ با توجه به شکل (۲) برای هر $i \neq j$ با فرض این که هیچ حسگری به طور کامل درون حسگر دیگر قرار ندارد، به صورت زیر تعریف می‌شود:

$$Vor(s_i) = \{p \in E^2: d(c_i, p) - r_i \leq d(c_j, p) - r_j\} \quad (2)$$

$$i = 1, \dots, n \text{ و } j = 1, \dots, n$$

$d(c_i, p)$ بیانگر فاصله نقطه p یک نقطه در فضای اقلیدسی است از مرکز حسگر s_i است.

۳- تعریف مسئله

با توجه به قسمت قبل، ما مسئله تشخیص حفره در یک محیط با مجموعه‌ای از حسگرها را در حالت‌های مختلف مشاهده کردیم و کارهایی که در این زمینه صورت گرفته بود را مورد مطالعه قرار دادیم. حال در این قسمت، مسئله‌ای را که در این مقاله به بررسی آن می‌پردازیم بیان کرده و در قسمت‌های بعدی بر آن هستیم تا الگوریتم کارآیی را برای آن ارائه دهیم.

$$d(c_i, p) - r_i = d(c_j, p) - r_j \quad (1)$$

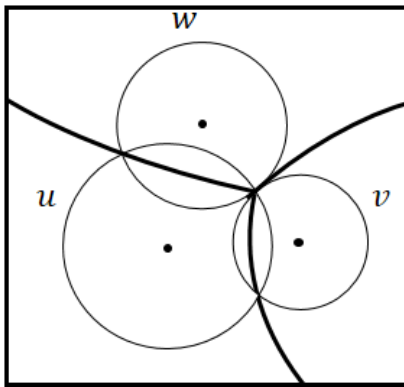
فرض کنید یک محیط با مانع شامل مجموعه‌ای از حسگرها با شعاع‌های متفاوت را به عنوان ورودی داریم، هدف این است که با استفاده از این حسگرها، حفره‌های موجود در محیط را تشخیص داده و یال‌های مرزی هر حفره به صورت دقیق گزارش شود. همچنین محیط شامل موانع شفاف^۱ است که مرکز حسگرها نمی‌توانند داخل موانع قرار بگیرند ولی اشعه حسگرها می‌توانند به موانع نفوذ کنند. شکل محیط و موانع هم به صورت چندضلعی نامحدب می‌باشد و اطلاعات مرز موانع و محیط و همچنین مختصات مراکز حسگرها را نیز به عنوان ورودی داریم. به بیان دقیق‌تر، مجموعه $O = \{o_1, o_2, \dots, o_l\}$ که نشان‌دهنده موانع^۲ موجود در محیط و مجموعه $S = \{s_1, s_2, \dots, s_n\}$ که نشان‌دهنده حسگرها با شعاع متفاوت است به عنوان ورودی در یک محیط داده می‌شود و هدف این است که نخست حفره‌های موجود در محیط را که با $H = \{H_1, H_2, \dots, H_{|H|}\}$ نشان می‌دهیم که $|H|$ نشان‌دهنده تعداد حفره‌های موجود در محیط است، تشخیص داده و سپس با توجه به مجموعه نقاط مرزی حفره‌ها که آن را با B نشان می‌دهیم و به صورت $B = \{h_1^1, h_1^2, \dots, h_1^{|H_1|}, \dots, h_{|H|}^1, h_{|H|}^2, \dots, h_{|H|}^{|H_{|H|}|}\}$ تعریف می‌شود که $|h_i|$ نشان‌دهنده تعداد نقاطی می‌باشد که حفره h_i را تشکیل می‌دهند، به ازای هر حفره مجموعه یال‌های مرزی آن را که به صورت $H_i = \{e_{i1}, e_{i2}, \dots, e_{i|h_i|}\}$ نشان می‌دهیم و هر یال حفره به صورت $e_{ij} = h_i^j h_i^{j+1}$ که $j = 1, \dots, |h_i|$ می‌باشد، گزارش کنیم.

1- Transparent Obstacle
2- Obstacle

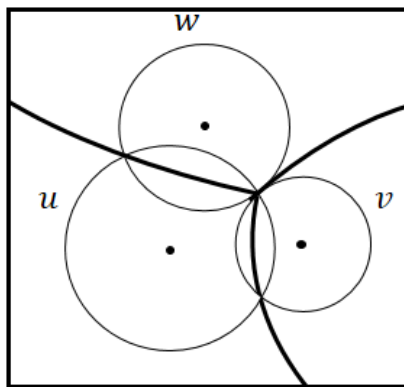
بنابراین، طبق ویژگی نمودار ورونوی وزندار جمعی این نقاط جزء مجموعه رأس‌های ورونوی این سه حسگر محسوب می‌شوند، بر این اساس، داریم:

$$d(c_w, b_{u,v}) = d(c_u, b_{u,v}) = d(c_v, b_{u,v}) = 0$$

از اصل ۱ و لم ۱ می‌توان نتیجه گرفت که تمام نقاط تقاطع حسگرها روی یال‌ها یا رأس‌های ورونوی قرار می‌گیرند مگر این که این نقاط تقاطع توسط حسگر دیگری پوشش داده شده باشند، پس نقاط تقاطعی که روی یال‌های ورونوی قرار می‌گیرند نشان‌دهنده نقاط مرزی حفره‌ها هستند و مرز حفره‌ها توسط این نقاط تقاطع به دست می‌آیند. نقاط مرزی حفره‌ها نقاطی هستند که از تقاطع حسگرها به دست آمده‌اند و توسط هیچ حسگر دیگری پوشش داده نشده‌اند.

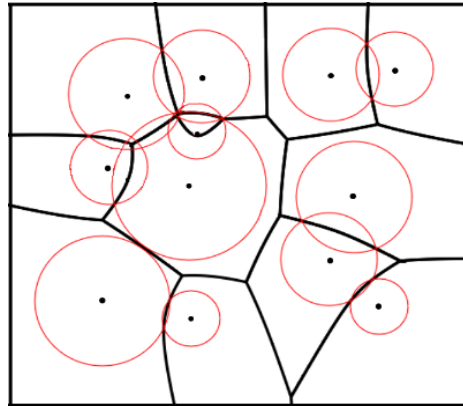


شکل (۳): حسگر w حداقل بر یکی از نقاط تقاطع حسگر u و v مماس است.



شکل (۴): حسگر w حداقل بر یکی از نقاط تقاطع حسگر u و v مماس است.

قضیه ۱: در یک محیط بدون مانع با n حسگر ثابت، تعداد کل نقاط مرزی حفره‌ها که از تقاطع حسگرها به دست می‌آیند از مرتبه $O(n)$ می‌باشد.

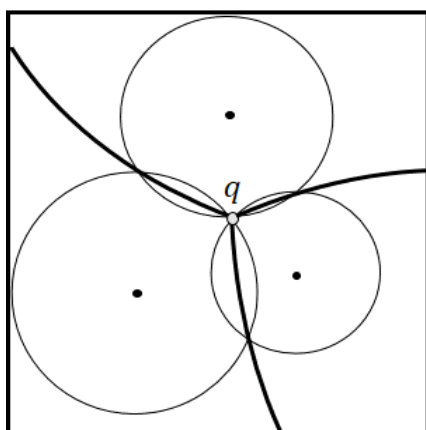


شکل (۲): نمودار ورونوی وزندار جمعی به ازای مجموعه‌ای از حسگرها در این نمودار یال‌های ورونوی می‌توانند به صورت خط مستقیم یا به صورت کمان باشند که این کمان‌ها بخشی از شاخه‌های هذلولی بین سلول‌های همسایه هستند و همچنین سلول ورونوی ایجاد شده به ازای حسگرها لزوماً محدب نیست.

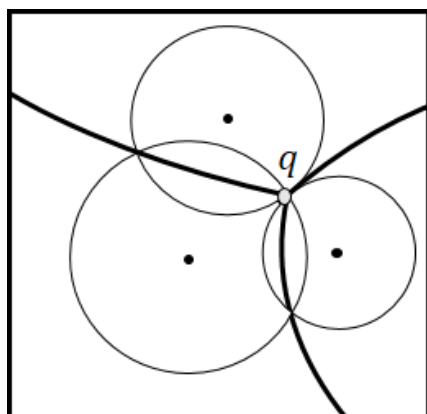
اصل ۱: اگر دو حسگر u و v در یک محیط با هم تقاطع داشته باشند، آن‌گاه نقاط تقاطع این دو حسگر روی یال ورونوی وزندار جمعی به دست آمده از این دو حسگر قرار می‌گیرد.

لم ۱: اگر دو حسگر u و v با هم تقاطع داشته باشند و حسگر w دیگری مانند w طوری در محیط قرار گرفته باشد که حداقل یکی از نقاط تقاطع حسگر u و v را پوشش دهد، آن‌گاه این نقاط تقاطع روی یال‌های ورونوی وزندار جمعی به دست آمده از این سه حسگر قرار نمی‌گیرند، ولی اگر حسگر w به این نقاط تقاطع مماس باشد آن‌گاه این نقاط جزء رأس‌های ورونوی می‌باشند.

اثبات: فرض کنیم حداقل یکی از نقاط تقاطع دو حسگر u و v توسط حسگر w پوشش داده شده باشد یعنی $d(c_w, b_{u,v}) < r_w$ (شکل (۳)). با توجه به این که $b_{u,v}$ روی مرز دو حسگر u و v قرار دارند پس فاصله این نقاط از مرز دو حسگر u و v صفر می‌باشد. هنگامی که حداقل یکی از نقاط تقاطع حسگر u و v داخل حسگر w قرار می‌گیرند حتماً فاصله این نقاط از مرز حسگر w بزرگتر از صفر می‌باشد، یعنی $d(c_w, b_{u,v}) > 0$ ، بنابراین، طبق ویژگی نمودار ورونوی وزندار جمعی نقاطی از $b_{u,v}$ که توسط حسگر w پوشش داده شده‌اند روی یال‌های ورونوی به دست آمده از سه حسگر u و v و w قرار نمی‌گیرند و درون سلول ورونوی حسگر w قرار می‌گیرند. ولی هنگامی که حداقل یکی از نقاط تقاطع حسگر u و v روی مرز حسگر w قرار می‌گیرد (شکل (۴))، آن‌گاه $d(c_w, b_{u,v}) = 0$



شکل (۵). نقطه q جزء مجموعه رئوس ورونوی می‌باشد ولی جزء مجموعه نقاط مرزی حفره‌ها نمی‌باشد.



شکل (۶). نقطه q جزء مجموعه رئوس ورونوی و نقاط مرزی حفره‌ها می‌باشد.

پیچیدگی ساخت نمودار ورونوی وزندار جمعی به ازای مجموعه‌ای از n حسگر موجود در محیط از مرتبه $O(n \log^2 n)$ می‌باشد [۲۷] و با توجه به اینکه بررسی شرایط فوق دارای پیچیدگی زمانی $O(n)$ می‌باشد. بنابراین، پیچیدگی کل به دست آوردن مجموعه نقاط مرزی حفره‌ها در یک محیط از مرتبه $O(n \log^2 n)$ می‌باشد.

در مرحله بعدی هدف ما مشخص کردن تعداد حفره‌ها و گزارش یال‌های مرزی هر حفره به صورت دقیق است. که تعداد کل حفره‌های موجود در محیط را با مجموعه $H = \{H_1, H_2, \dots, H_{|H|}\}$ نشان می‌دهیم که $|H|$ نشان‌دهنده تعداد حفره‌ها در محیط می‌باشد. به بیان دیگر هدف ما گزارش تمامی یال‌های مرزی تشکیل دهنده حفره‌ها با استفاده از نقاط مرزی

اثبات: با توجه به این که نمودار ورونوی وزندار جمعی از n حسگر ساخته شده است. بنابراین، تعداد رأس‌ها و تعداد یال‌های ورونوی موجود در محیط از مرتبه $O(n)$ است و می‌دانیم که هر یال ورونوی توسط دو حسگر دلخواه مانند s_i و s_j ساخته شده است و با توجه به اصل ۱ و لم ۱ تمامی نقاط مرزی حفره‌ها از تقاطع حسگرها به دست می‌آیند و این نقاط روی یال‌های ورونوی قرار می‌گیرند و یا جزء مجموعه رئوس ورونوی می‌باشند. بنابراین، تعداد نقاط مرزی حفره‌ها در محیط از مرتبه $O(n)$ می‌باشد.

یادآوری می‌کنیم تمامی نقاط تقاطع حسگرها: ۱- درون سلول‌های ورونوی یا، ۲- روی یال‌های ورونوی و یا، ۳- جزء مجموعه رئوس ورونوی می‌باشند. در حالت ۱، با توجه به لم ۱ نقاط تقاطع حداقل توسط حسگری که درون آن سلول ورونوی قرار دارد، پوشش داده شده است. بنابراین، در این حالت نقاط تقاطع جزء مجموعه نقاط مرزی حفره‌ها نمی‌باشند. در حالت ۲، براساس اصل ۱ و لم ۱ نقاط تقاطع جزء نقاط مرزی حفره‌ها محسوب می‌شوند. در حالت ۳، مجموعه نقاط رئوس از تقاطع حسگرها یا از امتداد یال‌های ورونوی به دست آمده‌اند. برای حالتی که مجموعه نقاط رئوس از امتداد یال‌های ورونوی به دست آمده‌اند این نقاط را بررسی نمی‌کنیم چون هدف ما به دست آوردن نقاط مرزی حفره‌ها است و این نقاط جزء نقاط مرزی حفره‌ها محسوب نمی‌شوند چرا که از تقاطع حسگرها به دست نیامده‌اند. برای حالتی که یک رأس دلخواه ورونوی از تقاطع تعدادی حسگر به دست آمده باشد، ابتدا معادله هر کدام از یال‌های گذرنده از رأس موردنظر را با حسگرهای تشکیل‌دهنده این یال‌ها مقایسه می‌کنیم. با توجه به این که می‌دانیم هر یال ورونوی توسط دو حسگر تشکیل شده است، اگر هر کدام از یال‌ها در دو نقطه مجزا با حسگرهای تشکیل‌دهنده آن یال اشتراک داشته باشند، به بیان دقیق‌تر اگر معادله یال تشکیل شده از دو حسگر با معادله دو حسگر در دو نقطه مجزا با هم برابر باشند و این ویژگی برای تمامی یال‌های گذرنده از این رأس برقرار باشد رأس موردنظر جزء مجموعه نقاط مرزی حفره‌ها نمی‌باشد (شکل (۵))، در غیر این صورت اگر حداقل یکی از این یال‌ها ویژگی بیان شده را نداشته باشند، رأس موردنظر جزء مجموعه نقاط مرزی حفره‌ها می‌باشد (شکل (۶)). با توجه به اینکه تعداد کل سلول‌ها و همچنین رأس‌های ورونوی موجود در محیط از مرتبه $O(n)$ است. بنابراین، پیچیدگی این مرحله بعد از محاسبه نمودار ورونوی وزندار جمعی از مرتبه $O(n)$ است.

در مرحله قبل توانستیم مجموعه یال‌هایی که حفره‌های موجود در محیط را تشکیل می‌دهند به دست آوریم. در این مرحله هدف آن است که تشخیص دهیم هر یال مربوط به کدام حفره موجود در محیط می‌باشد. به بیان دیگر هدف ما پیدا کردن تمامی یال‌هایی می‌باشد که حفره H_i را تشکیل می‌دهند (شکل ۷). با توجه به این که مجموعه یال‌هایی که حفره‌ها را تشکیل می‌دهند در اختیار داریم و همچنین نقطه انتهایی هر یال، نقطه ابتدایی یال بعدی می‌باشد. می‌توانیم این یال‌ها را بر اساس نقاط ابتدایی طبق مختصات x مرتب کنیم سپس با شروع از یک یال دلخواه مانند e_{ij} ، یال e_{ij+1} را طوری انتخاب کنیم که نقطه انتهایی یال e_{ij} برابر نقطه ابتدایی یال e_{ij+1} باشد که برای یافتن یال e_{ij+1} از جستجوی دودویی استفاده می‌کنیم. متذکر می‌شویم که یال e_{ij} می‌تواند با چند یال دیگر تقاطع داشته باشد در این حالت یالی را به عنوان یال e_{ij+1} در نظر می‌گیریم که جهت حرکت (ساعتگرد یا پادساعتگرد) آن در جهت حرکت یال e_{ij} باشد و همچنین یال e_{ij+1} روی مرز حسگری که یال e_{ij} روی آن قرار گرفته است، نباشد. این روند را تا جایی انجام می‌دهیم که به نقطه ابتدایی یال اولی برسیم و تمامی یال‌ها را دقیقاً یک بار ملاقات کنیم. با توجه به این که به دست آوردن هر یال e_{ij+1} بر اساس جستجوی دودویی از مرتبه $O(\log n)$ است و در کل تعداد یال‌ها $O(n)$ است، پس پیچیدگی کل به دست آوردن یال‌های مرزی تمامی حفره‌ها در یک محیط، از مرتبه $O(n \log n)$ می‌باشد. برای به دست آوردن مساحت دقیق حفره‌ها هم می‌توان هر حفره را بر اساس مجموعه نقاط مرزی مثلث‌بندی کرد و به ازای هر مثلث ایجاد شده، مساحت ناحیه تشکیل شده از قوس یال و خط مستقیم بین دو نقطه ابتدایی و انتهایی قوس یال را از مساحت کل مثلث کم کرد، تا مساحت دقیق حفره در هر مثلث به دست آید و در انتها باید مساحت باقیمانده مثلث‌ها را با هم جمع کرد تا مساحت دقیق هر حفره در محیط به دست آید. با توجه به این که مثلث‌بندی محیط به ازای $O(n)$ نقطه مرزی حفره‌ها دارای پیچیدگی $O(n \log n)$ است و محاسبه مساحت حفره در هر مثلث دارای پیچیدگی زمانی ثابتی می‌باشد. بنابراین پیچیدگی کل به دست آوردن مساحت دقیق حفره‌ها در محیط، از مرتبه زمانی $O(n \log n)$ می‌باشد.

بنابراین در کل می‌توان نتیجه گرفت در یک محیط بدون مانع شامل مجموعه‌ای از n حسگر ثابت، اگر هدف به دست آوردن حفره‌های موجود در محیط باشد، می‌توان با ساخت نمودار

می‌باشد. که محیط شامل دو نوع حفره است: ۱- حفره بسته^۱ حفره‌ای است که مرز آن بوسیله مرز حسگرها تشکیل شده است. ۲- حفره باز^۲ حفره‌ای است که مرز آن بوسیله مرز حسگرها و محیط (و یا موانع) تشکیل شده است. بنابراین، یال‌های حفره به صورت کمان هستند که همان مرزهای حسگرها می‌باشند و یا به صورت خط مستقیم هستند که مرز محیط و موانع می‌باشند.

در این مرحله به ازای هر سلول ورونوی، نقاط تقاطع روی یال‌ها و رأس‌های ورونوی که جزء نقاط مرزی حفره‌ها هستند را در یک مجموعه قرار می‌دهیم. سپس یال‌هایی که حفره‌ها را ایجاد می‌کنند به دست می‌آوریم. برای تشخیص یال‌هایی که حفره‌ها را تشکیل می‌دهند باید نقاط ابتدایی و انتهایی این یال‌ها را که قبلاً در مجموعه‌های مجزا قرار داده‌ایم به دست آوریم که برای رسیدن به این هدف ابتدا نقاط تقاطع درون هر مجموعه را به صورت ساعتگرد مرتب کرده و یال به دست آمده از اتصال دو نقطه متوالی در صورتی که توسط هیچ حسگر دیگری پوشش داده نشده باشد به عنوان یال حفره در نظر گرفته می‌شود. به بیان دیگر فرض کنید نقاط مرزی روی حسگر s_i که به صورت ساعتگرد مرتب شده است را با $h_i^{h_i}, \dots, h_i^1, h_i^2, \dots, h_i^{h_i}$ نشان می‌دهیم هدف ما تشخیص یال دلخواهی مانند $h_i^j h_i^{j+1}$ است که توسط هیچ حسگر دیگری پوشش داده نشده باشد. فرض کنید نقطه h_i^j و h_i^{j+1} از تقاطع حسگرها به دست آمده‌اند و جزء مجموعه نقاط مرزی حفره‌ها می‌باشند. در صورتی یال $h_i^j h_i^{j+1}$ جزء مجموعه یال‌های حفره‌ها می‌باشد که تمامی نقاط روی این یال توسط حسگرهایی که h_i^j و h_i^{j+1} از تقاطع آنها به دست آمده‌اند، پوشش داده نشده باشند. به بیان دقیق‌تر اگر یک نقطه روی یال $h_i^j h_i^{j+1}$ در نظر بگیریم و بررسی کنیم که این نقطه توسط حسگرهایی که h_i^j و h_i^{j+1} از تقاطع آنها به دست آمده‌اند پوشش داده شده است یا خیر، می‌توان در مورد یال‌های مرزی حفره‌ها بحث کرد. این روند را برای هر حسگر و به ازای تمامی نقاط مرزی تا جایی انجام می‌دهیم که به نقطه شروع برسیم. بر اساس قضیه ۱ تعداد نقاط مرزی حفره‌ها از مرتبه $O(n)$ است در نتیجه تعداد یال‌های حفره‌ها نیز از مرتبه $O(n)$ است و تشخیص اینکه دو نقطه متوالی، تشکیل یال حفره را می‌دهند دارای مرتبه زمانی ثابتی می‌باشد. بنابراین پیچیدگی کل به دست آوردن یال‌های حفره‌ها بر اساس مرتب‌سازی نقاط مرزی حفره‌ها، از مرتبه $O(n \log n)$ می‌باشد.

تمام نقاط مرزی (h_i^j) که روی آن حسگر قرار دارند را به صورت ساعتگرد در یک مجموعه مرتب می‌کنیم.

۵- برای هر h_i^j از هر مجموعه مرتب شده:

اگر $d(e_{ij}, s_i) < r_i$

آنگاه $e_{ij} \in H$

در غیر این صورت:

$e_{ij} \notin H$

۶- برای مجموعه یال‌های H ، با انتخاب یال شروع و پیمایش

روی یال‌ها، یال‌های مرزی هر حفره به دست می‌آید.

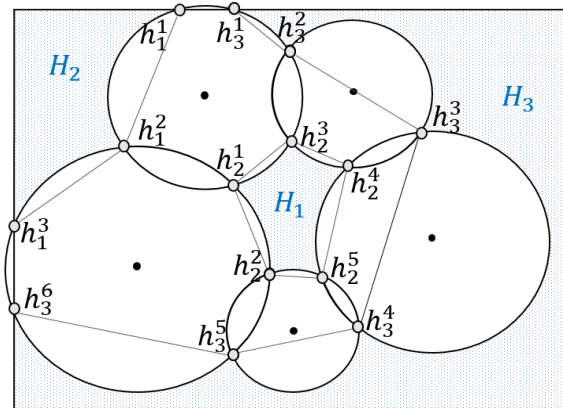
۷- هر حفره بر اساس نقاط مرزی مثلث‌بندی می‌شود و مساحت

دقیق هر حفره در محیط به دست می‌آید.

۵- تشخیص حفره در یک محیط با مانع

در قسمت قبل، مسئله تشخیص حفره را در حالتی که محیط بدون مانع بود بررسی کردیم و الگوریتم کارآیی برای آن ارائه دادیم. در ادامه این مسئله را زمانی که محیط شامل مجموعه‌ای از موانع است بررسی می‌کنیم. در حالتی که محیط شامل موانع است در این صورت ما قادر نیستیم در ناحیه‌هایی که توسط موانع احاطه شده است حسگر قرار دهیم بنابراین در این قسمت هدف بر آن است که در صورت وجود موانع در محیط، الگوریتمی ارائه دهیم تا توانایی تشخیص حفره را به صورت دقیق داشته باشد. بنابراین مسئله تشخیص حفره را در حالتی که محیط شامل مجموعه موانع $O = \{o_1, o_2, \dots, o_l\}$ است که هر مانع به صورت یک چند ضلعی نامحدب می‌باشد. بسیاری از الگوریتم‌های پیشین [۱، ۳، ۱۳، ۲۱] برای حالتی که محیط بدون مانع باشد حفره‌های موجود در محیط را به درستی گزارش می‌کنند و دارای پیچیدگی زمانی بالایی هستند. اما برای حالتی که محیط شامل مانع باشد این الگوریتم‌ها غیر قابل کاربرد می‌باشند. همچنین بیشتر این الگوریتم‌ها برای حالتی که گرفته شده‌اند که شعاع حسگرها یکسان است و جوابی برای حالتی که شعاع حسگرها متفاوت باشد نمی‌دهند. در نتیجه ما در ادامه الگوریتم کارآیی را برای تشخیص حفره‌های موجود در یک محیط با مانع، زمانی که شعاع حسگرها متفاوت است، ارائه می‌کنیم و بهینگی الگوریتم ارائه شده را برای این مسئله اثبات می‌کنیم.

ورونوی وزندار جمعی به ازای حسگرهای موجود در محیط و با استفاده از ویژگی‌های نمودار ورونوی، تعداد حفره‌های موجود در محیط و همچنین یال‌های مرزی هر حفره را به صورت دقیق در زمان $O(n \log^2 n)$ گزارش کرد.



شکل (۷): حفره بسته، H_2 و H_1 حفره باز، h_i^j : زمین نقطه مرزی حفره نام.

شبه کد الگوریتم مسئله تشخیص و گزارش حفره‌ها در حالت بدون مانع به صورت زیر می‌باشد:

ALG1 HDAO(S, W)

ورودی: یک محیط W شامل

$S = \{s_1, s_2, \dots, s_n\}$ که نشان‌دهنده حسگرهای ثابت است و $s_i = (c_i, r_i)$ و $c_i = (x_i, y_i)$ مرکز و r_i شعاع حسگر s_i می‌باشد.

خروجی: $H = \{H_1, H_2, \dots, H_{|H|}\}$ که نشان‌دهنده مجموعه حفره‌های موجود در محیط است و $H_i = \{e_{i1}, e_{i2}, \dots, e_{i|h_i|}\}$

۱- نمودار ورونوی وزندار جمعی به ازای مجموعه حسگرهای S ساخته می‌شود.

۲- نقاط تقاطع حسگرهایی که روی یال‌های ورونوی قرار می‌گیرند جزء مجموعه نقاط مرزی حفره‌ها (B) می‌باشند.

۳- برای مجموعه رأس‌های ورونوی $V = \{ver_1, \dots, ver_{|V|}\}$ از تقاطع حسگرها به دست آمده‌اند:

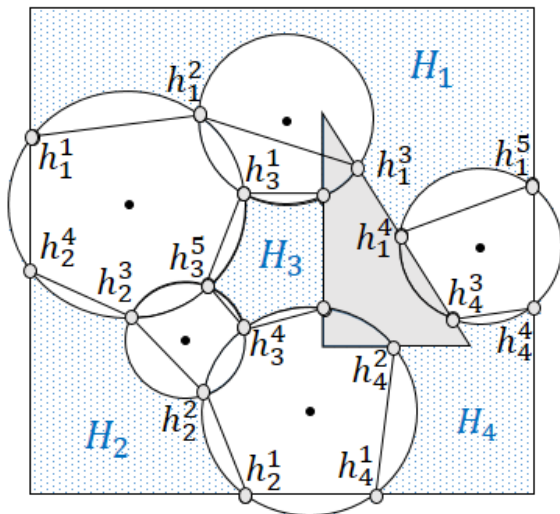
اگر $d(ver_j, s_i) < r_i$

آنگاه $ver_j \in B$

در غیر این صورت:

$ver_j \notin B$

۴- برای هر حسگری که مرکز سلول ورونوی می‌باشد:



شکل (۸). H_1 و H_2 و H_4 حفره باز و H_3 بسته، h_i^j : زمین نقطه مرزی حفره نام.

ALG2 HDPO(O, S, W)

ورودی: یک محیط W شامل

مجموعه $O = \{o_1, o_2, \dots, o_l\}$ که نشان‌دهنده موانع موجود در محیط و $S = \{s_1, s_2, \dots, s_n\}$ که نشان‌دهنده مجموعه حسگرهای ثابت است و $s_i = (c_i, r_i)$ و $c_i = (x_i, y_i)$ مرکز و r_i شعاع حسگر s_i می‌باشد.

خروجی: $H = \{H_1, H_2, \dots, H_{|H|}\}$ که نشان‌دهنده مجموعه حفره‌های موجود در محیط است و $H_i = \{e_{i1}, e_{i2}, \dots, e_{i|h_i|}\}$

$$H' \leftarrow HDAO(O, S, W) \quad -1$$

-2 برای هر $i \leq l$:

نقطه برخورد هر مانع o_i را با مجموعه H' به دست می‌آوریم و حفره‌ها را به روز می‌کنیم.

-3 مجموعه H را به عنوان خروجی الگوریتم گزارش می‌کنیم.

با توجه به این که تعداد رئوس حفره‌های موجود در محیط از مرتبه $O(n)$ می‌باشد و تعداد رئوس موانع m می‌باشد که در بدترین حالت ممکن است تمامی موانع با هم و با حفره‌ها برخورد داشته باشند در نتیجه پیچیدگی الگوریتم تشخیص حفره در حالت با موانع $O(n \log^2 n + nm^2)$ می‌باشد. در ادامه ثابت می‌کنیم که پیچیدگی الگوریتم HDPO یک حد پایین برای مسئله تشخیص حفره همراه با گزارش یال‌های مرزی حفره‌ها می‌باشد.

در این قسمت مانند رویکردی که در حالت بدون موانع اتخاذ کردیم، با فرض اینکه هیچ مانعی در محیط وجود ندارد ابتدا حفره‌های موجود در محیط را با استفاده از الگوریتم HDAO (Hole Detection for Absence of the Obstacles) به دست می‌آوریم. با توجه به اینکه مرز حفره‌ها در حالت بدون مانع توسط حسگرها و مرز محیط ساخته می‌شوند در نتیجه حفره‌ها به شکل چندضلعی‌های ساده هستند که مرز این چندضلعی‌ها می‌توانند به صورت کمان یا خطوط مستقیم می‌باشند بنابراین در ادامه منظور ما از حفره همان چندضلعی ساده می‌باشد. با توجه به قضیه ۱ حداکثر تعداد نقاط مرزی حفره‌ها از مرتبه $O(n)$ می‌باشد در نتیجه حداکثر تعداد رئوس چندضلعی‌های ایجاد شده توسط حفره‌های موجود در محیط در حالت بدون مانع از مرتبه $O(n)$ می‌باشد. در مرحله بعد با یک رویکرد افزایشی، موانع را به محیط اضافه می‌کنیم. در این مرحله بعد از اضافه کردن هر مانع به محیط ابتدا نقاط تقاطع هر مانع با حفره‌ها را محاسبه کرده و بر اساس این نقاط تقاطع حفره‌ها را بروز می‌کنیم (شکل ۸). به بیان دقیق‌تر ممکن است در روند اضافه شدن هر مانع به محیط ۱- حفره جدیدی ایجاد شود و یا ۲- یک حفره دلخواه حذف شود. ما این روند افزایشی را برای تمام موانع موجود در محیط انجام داده و در هر مرحله بعد از اضافه کردن هر مانع به محیط، با توجه به در اختیار داشتن اطلاعات مرز موانع و محیط، مساحت دقیق حفره‌ها و نقاط مرزی هر حفره را بروز می‌کنیم. بنابراین، به عنوان ورودی الگوریتم HDPO (Hole Detection for Presence of the Obstacles) مختصات مراکز مجموعه حسگرهای S و اطلاعات مرز موانع و محیط را داریم که الگوریتم HDPO ابتدا مانند الگوریتم HDAO مجموعه حفره‌های موجود در محیط را با توجه به محل قرارگیری مجموعه حسگرهای S و همچنین با فرض اینکه محیط شامل هیچ مانعی نباشد، تشخیص می‌دهد که حفره‌های موجود در محیط را با H' نمایش می‌دهیم. در مرحله بعد الگوریتم HDPO برخورد هر مانع o_i را که به شکل یک چندضلعی نامحدب می‌باشد با مجموعه H' که به شکل چندضلعی‌های ساده که یال‌های این چندضلعی‌ها به صورت کمان و یا خطوط مستقیم می‌باشد، محاسبه می‌کند و با توجه به نقاط تقاطع مجموعه O و H' مختصات مجموعه حفره‌های ایجاد شده را که با نشان H می‌دهیم در خروجی گزارش می‌کند. در ادامه شبه کد الگوریتم HDPO را بیان می‌کنیم.

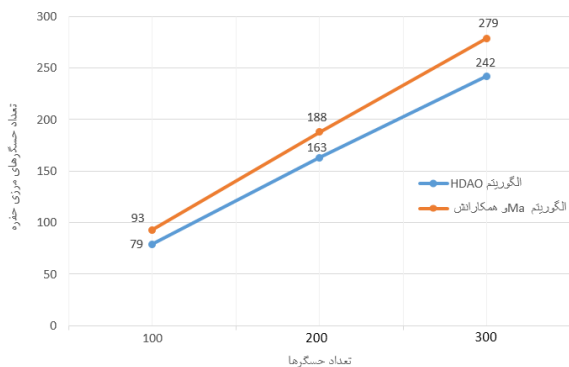
صورت چندضلعی‌های نامحدب می‌باشد.

۶-۲- نتایج شبیه‌سازی

۶-۲-۱. شبیه‌سازی الگوریتم در حالت بدون مانع

با توجه به اینکه الگوریتم‌های پیشین مسئله تشخیص حفره بیشتر در حالتی بررسی شدند که شعاع حسگرها یکسان و محیط بدون مانع است و این الگوریتم‌ها قابل بسط به حالتی که شعاع حسگرها متفاوت و محیط شامل مانع باشد، نیستند. بنابراین، ابتدا الگوریتم *HDAO* را برای حالت اول (محیط بدون مانع و شعاع حسگرها یکسان) شبیه‌سازی کرده و نتایج حاصل را با الگوریتم ارائه شده توسط *Ma* و همکارانش [۲۲] مقایسه می‌کنیم.

با توجه به شکل (۹)، الگوریتم *HDAO* حسگرهای مرزی حفره را که شعاع هر حسگر ۵ متر است به صورت دقیق گزارش می‌کند در حالی که الگوریتم ارائه شده توسط *Ma* و همکارانش [۲۲] با خطا همراه است.



شکل (۹): گزارش حسگرهای مرزی حفره توسط الگوریتم *HDAO* و الگوریتم *Ma* و همکارانش

در ادامه، شبیه‌سازی را برای حالتی که محیط بدون مانع است انجام می‌دهیم. با توجه به شکل (۱۰) بعد از استقرار مجموعه‌ای از حسگرهای ثابت در یک محیط بدون مانع، حفره‌های تشکیل شده در محیط را براساس نقاط مرزی به دست می‌آوریم که ناحیه‌های خاکستری رنگ ناحیه حفره‌های به دست آمده توسط الگوریتم *HDAO* می‌باشد. نتایج حاصل از شبیه‌سازی نشان می‌دهد که الگوریتم *HDPO* و *HDAO* حفره‌های موجود در محیط را در حالتی که چه تراکم حسگرها در محیط کم و چه زیاد باشد به صورت کامل و دقیق تشخیص می‌دهد. همچنین شکل حفره‌های (باز و بسته) ایجاد شده در محیط تأثیری در جواب الگوریتم ندارد یعنی الگوریتم به ازای هر حفره با شکل‌های محدب و نامحدب درست کار می‌کند.

۵-۱. حد پایین الگوریتم‌های قطعی برای تشخیص حفره

در این قسمت هدف بر آن است که نشان دهیم پیچیدگی زمانی هر الگوریتم قطعی برای مسئله تشخیص حفره همراه با گزارش دقیق نقاط مرزی هر حفره حداقل $O(nm^2)$ است که n نشان‌دهنده تعداد حسگرهای موجود در محیط و m نشان‌دهنده تعداد رئوس موانع می‌باشد.

قضیه ۴: در یک محیط با مانع با n حسگر، اگر تعداد رئوس موانع m باشد آنگاه حد پایین پیچیدگی زمانی هر الگوریتم قطعی برای تشخیص حفره‌های موجود در محیط همراه با گزارش یال‌های مرزی هر حفره از مرتبه $O(nm^2)$ می‌باشد.

اثبات. با توجه به توزیع موانع در محیط، در بدترین حالت تمامی یال‌های مانع دلخواهی مانند o_i با تمامی حسگرها و با تمامی یال‌های موانع برخورد دارد بنابراین، برای گزارش حفره‌هایی که از برخورد این مانع با حسگرها ایجاد شده‌اند باید تمامی نقاط تقاطع این مانع را با حسگرها گزارش کنیم که واضح است تعداد این نقاط $O(n)$ می‌باشد و با توجه به این که تعداد رئوس موانع موجود در محیط m می‌باشد و تمامی یال‌های این موانع با یکدیگر و با حسگرها می‌توانند تقاطع داشته باشند بنابراین، واضح است که باید $O(nm^2)$ نقطه را برای تمام حفره‌ها در خروجی هر الگوریتم دلخواه گزارش کنیم. در نتیجه پیچیدگی زمانی هر الگوریتم قطعی برای تشخیص حفره‌ها در یک محیط در بدترین حالت بهتر از $O(n \log^2 n + nm^2)$ نمی‌شود و این پیچیدگی یک حد پایین برای هر الگوریتم قطعی دلخواه می‌باشد. در نتیجه قضیه اثبات شد.

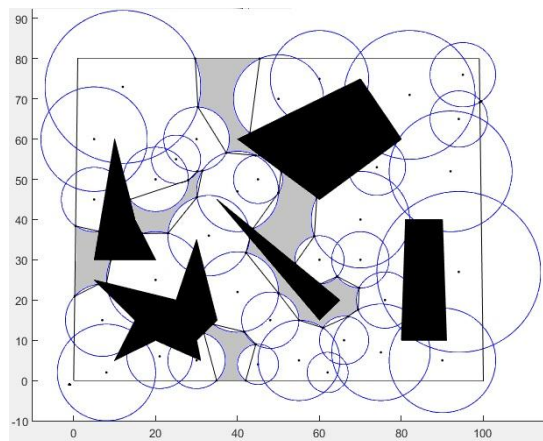
۶- ارزیابی نتایج

با توجه به الگوریتم‌های ارائه داده شده (*HDAO* و *HDPO*) در قسمت‌های قبلی و نتایج به دست آمده، در این بخش عملکرد الگوریتم را از لحاظ پیچیدگی و دقیق و کامل بودن الگوریتم مورد ارزیابی قرار می‌دهیم.

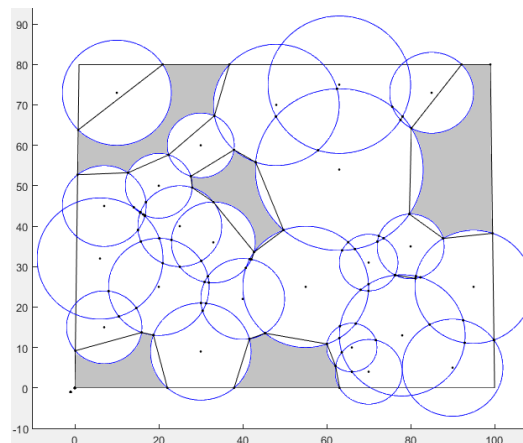
۶-۱- مرحله شبیه‌سازی

الگوریتم‌های *HDAO* و *HDPO* با استفاده از نرم‌افزار MatLab 2015 برای مجموعه‌ای از حسگرها شبیه‌سازی شده است که شعاع حسگرهای موجود در محیط متفاوت می‌باشد. همچنین محیط شامل مجموعه‌ای از موانع است و ابعاد محیط 200×200 متر می‌باشد. حسگرها به صورت تصادفی در محیط قرار داده شده‌اند و شعاع حسگرها بین بازه ۰ تا ۳۰ متر و شکل موانع به

پیچیدگی زمانی $O(n \log^2 n + nm^2)$ است که m نشان دهنده تعداد رئوس موانع موجود در محیط است. نتایج حاصل از شبیه سازی نشان می دهد که الگوریتم HDPO حفره های موجود در محیط را به صورت دقیق تشخیص می دهد.



شکل (۱۱): تشخیص حفره در یک محیط با مانع با استفاده از نقاط مرزی



شکل (۱۰): تشخیص حفره در یک محیط بدون مانع با استفاده از نقاط مرزی

الگوریتم ابتدا با استفاده از رویکرد ارائه شده برای تشخیص نقاط مرزی، به ازای هر حسگر نقاط مرزی را به دست آورده و با اتصال متوالی این نقاط یال های مرزی را به ازای هر حفره به دست می آورد.

مزیت اصلی الگوریتم ارائه داده شده، پیچیدگی پایین و دقیق و کامل بودن الگوریتم در تشخیص حفره های موجود در محیط است. در حالتی که محیط بدون مانع است و شعاع حسگرها متفاوت می باشد طبق شکل (۱۰) الگوریتم دارای پیچیدگی زمانی $O(n \log^2 n)$ می باشد که n تعداد حسگرهای موجود در محیط می باشد در حالی که الگوریتم های پیشین مسئله تشخیص حفره [۲۱،۳] بیشتر در حالتی در نظر گرفته شدند که شعاع حسگرها یکسان است و دارای پیچیدگی زمانی $O(n^3)$ و در [۱] دارای پیچیدگی زمانی $O(n^2)$ می باشند و همچنین الگوریتم های ارائه شده در [۲۱،۳] پس از تشخیص نقاط مرزی به ازای هر حسگر، حفره های موجود در محیط را به درستی تشخیص نمی دهد و با خطا همراه است.

۶-۲-۲- شبیه سازی الگوریتم در حالت با مانع

با توجه به دانش نویسندگان این مقاله و مطالعات انجام شده، مسئله تشخیص حفره در حالتی که شعاع حسگرها متفاوت و محیط شامل مجموعه ای از حسگرها باشد یک بحث جدید است که تاکنون مورد مطالعه قرار نگرفته است.

در این قسمت الگوریتم HDPO (الگوریتم ارائه داده شده در حالت با مانع) را وقتی محیط شامل مجموعه ای از حسگرها و موانع غیرمحدب است در یک محیط ۱۰۰×۸۰ متر شبیه سازی می کنیم. با توجه به شکل (۱۱) که ناحیه های سیاه رنگ نشان دهنده موانع موجود در محیط می باشد، الگوریتم دارای

۷- نتیجه گیری

ما در این مقاله مسئله تشخیص حفره در یک محیط را وقتی شعاع حسگرها متفاوت است با در دست داشتن مختصات مکان هر حسگر، در دو حالت بررسی کردیم. در حالت اول که محیط بدون مانع است به کمک نمودار ورونوی و زنندار جمعی نشان دادیم که تعداد کل نقاط مرزی حفره های موجود در محیط که از تقاطع n حسگر به دست می آیند از مرتبه $O(n)$ می باشد و یک الگوریتم متمرکز با پیچیدگی زمانی $O(n \log^2 n)$ برای تشخیص حفره با گزارش دقیق یال های مرزی هر حفره همراه با مساحت دقیق حفره ها در این حالت ارائه دادیم. در حالت دوم وقتی محیط شامل مانع است نشان دادیم که تعداد کل نقاط مرزی حفره های موجود در محیط از مرتبه $O(nm^2)$ است و این پیچیدگی یک حد پایین برای تمامی الگوریتم های قطعی مسئله تشخیص حفره همراه با گزارش یال های مرزی است و یک الگوریتم متمرکز برای تشخیص حفره همراه با گزارش یال های مرزی ارائه دادیم، که در الگوریتم ارائه داده شده اگر n تعداد حسگرهای موجود در محیط باشد و تعداد رئوس موانع موجود در محیط m باشد، آن گاه پیچیدگی کل الگوریتم تشخیص حفره در حالت با مانع از مرتبه زمانی $O(n \log^2 n + nm^2)$ می باشد. به عنوان کارهای آتی حسگرهای متحرک را نیز در نظر گرفته و بعد از محاسبه حفره در محیط به ازای حسگرهای ثابت و متحرک،

- [14] G. Wang, G. Cao, and T. LaPorta, "Movement-assisted sensor deployment," *Mobile Computing, IEEE Transactions on* 5.6, pp. 640-652, 2006.
- [15] S. Babaie and S. S. Pirahesh, "Hole detection for increasing coverage in wireless sensor network using triangular structure," *arXiv preprint arXiv: 1203.3772*, 2012.
- [16] Fekete, P. Sándor, et al., "Neighborhood-based topology recognition in sensor networks," *Algorithmic Aspects of Wireless Sensor Networks, Springer Berlin Heidelberg*, pp. 123-136, 2004.
- [17] R. Ghrist and A. Muhammad, "Coverage and hole-detection in sensor networks via homology," *Proceedings of the 4th international symposium on Information processing in sensor networks*, IEEE Press, 2005.
- [18] V. De Silva, R. Ghrist, and A. Muhammad, "Blind Swarms for Coverage in 2-D," *Proc. Robotics: Science and Systems*, pp. 335-342, June 2005.
- [19] X. Li, D. K. Hunter, and K. Yang, "WLC12-1: Distributed coordinate-free hole detection and recovery," *Global Telecommunications Conference, GLOBECOM'06, IEEE*, 2006.
- [20] X. Li and D. K. Hunter, "Distributed coordinate-free hole recovery," *Communications Workshops, ICC Workshops' 08. IEEE International Conference on. IEEE*, 2008.
- [21] P. K. Sahoo, J.-Z. Tsai, and H.-L. Ke, "Vector method based coverage hole recovery in wireless sensor networks," *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on. IEEE*, 2010.
- [22] H.-C. Ma, P. K. Sahoo, and Y.-W. Chen, "Computational geometry based distributed coverage hole detection protocol for the wireless sensor networks," *Journal of network and computer applications* 34.5, pp. 1743-1756, 2011.
- [23] C. Zhang, Y. Zhang, and Y. Fang, "Detecting coverage boundary nodes in wireless sensor networks," *Networking, Sensing and Control, 2006. ICNSC'06, Proceedings of the 2006 IEEE International Conference on, IEEE*, 2006.
- [24] K. Bi et al., "Topological hole detection in sensor networks with cooperative neighbors," *Systems and Networks Communications, 2006. ICSNC'06, International Conference on, IEEE*, 2006.
- [25] S. Funke, "Topological hole Detection in wireless sensor networks and its applications," *Proceedings of the 2005 joint workshop on foundations of mobile computing, ACM*, 2005.
- [26] S. Funke and C. Klein, "Hole detection or: how much geometry hides in connectivity," *Proceedings of the twenty-second annual symposium on Computational geometry, ACM*, 2006.
- [27] M. Sharir, "Intersection and closest-pair problems for a set of planar discs," *SIAM J. Comput.*, vol. 14, no. 2, pp. 448-468, 1985.

با جابه‌جایی حسگرهای متحرک سعی می‌کنیم پوشش بیشینه را در یک محیط با مانع ایجاد کنیم.

۸- مراجع

- [1] L. Wei and W. Zhang, "Coverage hole and boundary nodes detection in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 48, pp. 35-43, 2015.
- [2] M. Davoodi, E. Delfaraz, and S. Ghobadi, "Handoff Minimization in Wireless Networks With Group Mobility," *Journal of Electronical and Cyber Defence*, vol. 4, no. 3, Serial No.15, 2016.
- [3] K. Zhiping, H. Yu, and Q. Xiong, "Detection and recovery of coverage holes in wireless sensor networks," *Journal of Networks* 8.4, pp. 822-828, 2013.
- [4] Kalwaghe, N. Samidha, and A. V. Dusane, "Literature Review on Hole Detection and Healing in Wireless Sensor Network," *International Journal of Current Engineering and Technology* 4.6, pp. 4184-4188, 2014.
- [5] A. Ghosh, "Estimating coverage holes and enhancing coverage in mixed sensor networks," *Local Computer Networks, 29th Annual IEEE International Conference on, IEEE*, 2004.
- [6] P. K. Sahoo and W.-C. Liao, "HORA: A Distributed Coverage Hole Repair Algorithm for Wireless Sensor Networks," *Mobile Computing, IEEE Transactions on* 14.7, pp. 1397-1410, 2015.
- [7] P. Corke, R. Peterson, and D. Rus, "Finding holes in sensor networks," In: *Proceedings of the IEEE workshop on omniscient space: robot control*, 2007.
- [8] F. Li, B. Zhang, and J. Zheng, "Geographic hole-bypassing forwarding protocol for wireless sensor networks," *IET Commun*, vol. 5, no. 6, pp. 737-44, 2011.
- [9] J. Kanno, J. G. Buchart, R. R. Selmic, and V. Phoha, "Detecting coverage holes in wireless sensor networks," In: *Proceedings of the 17th mediterranean conference on control & automation, Greece*, June 2009.
- [10] B. Tong and W. Tavanapong, "On discovering sensing coverage holes in large-scale sensor networks," *Tech. Rep. TR 06-03, Computer Science, Iowa State University*, March 2006.
- [11] D. Du, F. Hwang, and S. Fortune, "Voronoi diagrams and Delaunay triangulations," *Euclidean Geometry and Computers*, 1992.
- [12] F. Aurenhammer, "Voronoi diagrams, a survey of a fundamental geometric data structure," *ACM Computing Surveys (CSUR)* 23.3, pp. 345-405, 1991.
- [13] G. Wang, G. Cao, and T. LaPorta, "A bidding protocol for deploying mobile sensors," *Network Protocols, 2003 Proceedings, 11th IEEE International Conference on, IEEE*, 2003.

A Centralized Algorithm Based on Voronoi Diagram for Hole Detection Problem in Wireless Sensor Networks

M. Davoodi Monfared*, E. Delfaraz Pahlevanlo, S. Ghobadi Babi, M. Masoori

*Department of Computer Science and Information Technology- Institute for Advanced Studies in Basic Sciences – Zanjan

(Received: 12/11/2016, Accepted: 13/02/2017)

ABSTRACT

One of the challenging problems in wireless sensor networks is detecting and calculating the area of holes which occur because of several reasons such as accidental death of sensors, explosion in the environment or running out of the energy of the sensors that they all disrupt the task of monitoring wireless sensor networks. When the sensors are distributed randomly in some places like dense forests and rugged terrain, it is possible to detect the holes in the environment. Therefore, in this paper we aim to provide a centralized geometric based algorithm to detect and calculate the area of the holes for a given environment. We investigate hole detection problem for a set of sensors with non-identical sensing radii in both cases: (i) when there is no obstacle in the environment and (ii) when there are some polygonal obstacles in the environment. We propose efficient algorithms for detecting holes in both cases. We calculate the amount of holes and report the boundary edges of the holes using computational geometry approaches. The time complexity of algorithm in case (i) is $O(n \log^2 n)$, and in case (ii) is $O(n \log^2 n + nm^2)$. The results of the simulation show that the provided algorithms exactly detect all holes in a given environment. Also, we compare the algorithm in case (i) with one of state-of-the-art algorithms in the literature. The comparison result shows efficiency of the proposed algorithm.

Keywords: Wireless Sensor Networks, Hole Detection, Weighted Voronoi Diagram

* Corresponding Author Email: mdmonfared@iasbs.ac.ir