DOR: 20.1001.1.2538161.1402.8.4.3.1

DOI : 10.22034/csj.2024.192413

زمستان ۱۴۰۲، دوره ۸، شماره ۴، پیاپی ۳۱

دریافت مقاله: ۱۴۰۲/۰۸/۰۷

پذیرش مقاله: ۱۴۰۲/۱۱/۱۱

نوع مقاله: پژوهشی

تور نگهبان چندگانه در چندضلعی پلکانی در حالت حداقل مجموع

رحمت قاسمی
دانشکده مهندسی کامپیوتر و مکاترونیک، واحد علوم و تحقیقات، دانشگاه آزاد اسلامی، تهران، ایران
پست الکترونیکی: rahmat.ghasemi@srbiau.ac.ir

علیرضا باقری*
دانشیاردانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)، تهران، ایران
پست الکترونیکی: ar_bagheri@aut.ac.ir

فاطمه کشاورز کوهجردی
دانشکده علوم پایه، گروه علوم کامپیوتر، دانشگاه شاهد، تهران، ایران
پست الکترونیکی: f.keshavarz@shahed.ac.ir

چکیده

در این مقاله، ما مسئلهٔ تورنگهبان چندگانه را در یک چندضلعی پلکانی بررسی کردیم. مسئله تورنگهبان یکی از مسائل مهم در حوزه هندسه محاسباتی است و یک نسخه دیگر از مسئله موزه هنری است. هدف در این مسئله، یافتن یک یا چند تور بسته برای یک یا چند نگهبان به منظور رویت تمامی‌ناحیه داخل چندضلعی است. ما مسئله را در حالت ثابت در نظر گرفتیم، به این معنی که نقاط شروع نگهبانان داخل چندضلعی پلکانی از پیش تعیین شده‌اند. دو معیار بهینه‌سازی در این مسئله حداقل مجموع، یعنی کمینه کردن مجموع طول تورها، و حداقل حداکثر، یعنی کمینه کردن ماکزیمم طول تورها، است. در این مقاله، یک الگوریتم مبتنی بر برنامه‌سازی پویا ارائه شده است که جواب بهینه مسئله را با معیار حداقل مجموع محاسبه می‌کند. الگوریتم پویا در هر دو حالت بدون دامینیت و با دامینیت دارای پیچیدگی زمانی $O(n^2 \cdot \log m)$ است. همچنین، این الگوریتم در هر دو حالت دارای پیچیدگی مکانی $O(n)$ است، که در آن m تعداد نگهبان‌ها و n تعداد رئوس چندضلعی داده شده است.

کلیدواژه‌ها: موزه هنر، تورنگهبان چندگانه، چندضلعی متعامد، برنامه‌سازی پویا، چندضلعی پلکانی.

۱- مقدمه

مسئلهٔ رویت‌پذیری[1] یک موضوع مهم تحقیقاتی در هندسه محاسباتی است. نقطه a نقطه b را در یک چندضلعی مانند P رویت می‌کند اگر پاره‌خطی که این دو نقطه را به هم وصل می‌کند اضلاع چندضلعی P را قطع نکند. به عبارت دیگر خط واصل این دو نقطه a و b به طور کامل توسط چندضلعی P محصور شده باشد [۱]. در مسئله‌ای مشابه، مسئلهٔ موزه هنر در سال ۱۹۷۳ توسط ویکتور کلی[2] در یک محاوره با واسک چوتال[3] مطرح شد

1-Visibility problem
2-Victor Klee
3-Vasek Chvátal

* نویسندهٔ مسئول

[۱] که چه تعداد نگهبان ثابت برای دیده‌بانی یک موزه به شکل چندضلعی مورد نیاز است. چوتال ثابت کرد که حداکثر $\lfloor\frac{n}{3}\rfloor$ نگهبان ثابت کافی است تا یک چندضلعی ساده با n رأس را نگهبانی کنیم [۲]. همچنین فیسک[4] یک اثبات بر اساس رنگ‌آمیزی گراف‌ها برای این موضوع ارایه کرد [۳]. لی[5] نشان داد که مسئلهٔ پیدا کردن حداقل تعداد نگهبان برای یک چندضلعی ساده ان-پی سخت[6] است [٤].

در مسئلهٔ تور نگهبان کلاسیک هدف پیدا کردن یک مسیر بسته مانند r درون چندضلعی P است به طوری که همه نقاط درون چندضلعی P توسط برخی از نقاط روی r قابل مشاهده باشند. مسئلهٔ تور نگهبان کاربردهای متنوعی دارد. از جمله کاربردهای آن می‌توان به امنیت، نظارت بر محیط‌های جغرافیایی، بهینه‌سازی انرژی و زمان اشاره کرد. مسئلهٔ تور نگهبان برای حالت بیشتر از یک نگهبان نیز تعریف می‌شود. در حالتی که دو یا چند نگهبان وجود داشته باشند هر نقطه در داخل چندضلعی باید حداقل از یک نقطه بر روی یکی از تورهای نگهبانی قابل رویت باشند. معیارهای مورد انتظار در حالت چندگانه می‌تواند حداقل کردن طول مجموع تورها (حداقل مجموع[7]) یا حداقل کردن طول تور ماکزیمم (حداقل حداکثر[8]) باشد [٥]. تور نگهبان یک مسئلهٔ دنیای واقعی در رباتیک را مدل‌سازی می‌کند. در این مسائل یک ربات یا دوربین متحرک باید یک بازرسی از محیط اطراف خود که هندسه آن داده شده است، داشته باشد [٦].

کارلسون و همکاران [۷] مساله تورنگهبان چندگانه را به عنوان حالت کلی مسئله مورد بررسی قرار دادند. این مسئله دارای دو حالت اصلی است: تور نگهبان شناور و تور نگهبان ثابت. در هر دو حالت تعدادی نگهبان متحرک (به عنوان مثال m) وجود دارند. در حالت شناور نقطه شروع نگهبانان مشخص نیست ولی در حالت ثابت نقاط شروع نگهبانان مشخص است. هدف اصلی پیدا کردن تورهایی برای همه نگهبانان است به گونه‌ای که همه نقاط درون چندضلعی حداقل توسط یکی از تورها قابل مشاهده باشد به طوری که مجموع طول تمام تورها (معیار حداقل مجموع) یا ماکزیمم طول تورها (معیار حداقل حداکثر)، کمینه باشد. آنها ثابت کردند که این مسئله در چندضلعی‌های ساده ان‌پی‌سخت است. سپس الگوریتمی با پیچیدگی زمانی چندجمله‌ای برای چندضلعی‌های هیستوگرام ارائه کردند. نیلسون و وود یک الگوریتم با پیچیدگی زمانی $O(n^2 \cdot m)$ و پیچیدگی مکانی $O(n^2)$ برای چندضلعی‌های حلزونی با نقاط شروع نامشخص ارائه کردند [۸، ۹].

نیلسون و شویرر [۱۰] مسئله را در چندضلعی‌های هیستوگرام به شکل دیگری مطرح کردند. در این حالت، هدف اصلی کمینه کردن طول بزرگترین تور (معیار حداقل-حداکثر) در بین m تور نگهبان (معیار حداقل حداکثر) است. آن‌ها یک الگوریتم با پیچیدگی زمانی $O(n^2 \cdot \log n)$ برای این حالت ارائه کردند. میشل و همکاران [۱۱] معیار حداقل حداکثر را در چندضلعی‌های متعامد یکنوا بررسی کردند و یک الگوریتم با پیچیدگی زمانی $O(n^4 \cdot m)$ برای آن ارائه دادند. آن‌ها نشان دادند که این مسئله برای $m = 2$، در حالت کلی ان‌پی‌سخت است. نیلسون و پکر [۵] یک الگوریتم تقریبی برای حالت دو نگهبان در چندضلعی‌های ساده ارائه کردند. پکر [۱۲] نیز ابتکارهایی را برای مسئلهٔ تورنگهبان چندگانه در حالت حداقل مجموع و حداقل حداکثر در چندضلعی‌های حفره‌دار یا بدون حفره مطرح کرد.

فرض کنید $x(p)$ و $y(p)$ به ترتیب مختصات x و y یک نقطه مانند p باشد. به ازای دو نقطه p و q می‌گوییم نقطه q توسط نقطه p دامینیت شده است، اگر $x(p) \geq x(q)$ و $y(p) \leq y(q)$ باشد. ما مسئلهٔ تور نگهبان را در دو حالت بررسی می‌کنیم. در حالت اول، فرض می‌شود که نقاط شروع نگهبان‌ها هیچ کدام یکدیگر را دامینت نمی‌کنند، در حالت دوم فرض می‌شود که برخی از نقاط شروع نگهبان‌ها توسط نقاط شروع دامینت می‌شوند.

در این مقاله، دو الگوریتم برای حل مسئله حداقل مجموع

4-Fisk
5-Lee
6-NP-Hard
7-min-sum
8-min-max

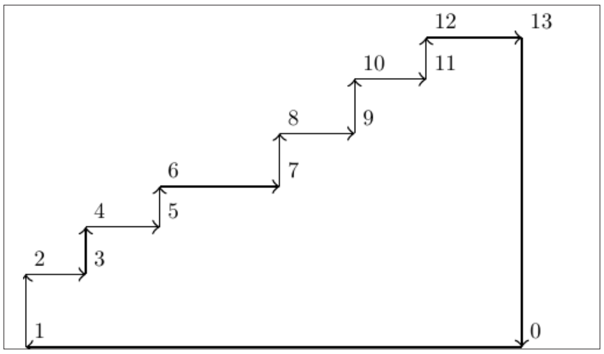
تورها در چندضلعی پلکانی با m نگهبان و نقاط شروع داده شده درون آن ارائه شده است. الگوریتم اول، در حالت بدون دامینیت و الگوریتم دوم، در حالت با دامینیت، هر دو با پیچیدگی زمانی $O(n^2 \cdot \log m)$ و پیچیدگی مکانی $O(n)$ می‌باشند. این الگوریتم‌ها به منظور محاسبه مجموع کمینه طول تورها با استفاده از نقاط شروع نگهبانان در داخل چندضلعی پلکانی با n رأس و m نقطه شروع ارائه شده است. جواب مسئله در چندضلعی پلکانی، در حالت شناور، واضح است. در این حالت، تنها کافی است یکی از نگهبانان را در هسته چندضلعی قرار دهیم. هسته چندضلعی، مجموعه نقاطی از چندضلعی است که تمام نقاط چندضلعی از هر یک از آنها قابل رویت است.

مسئلهٔ تور نگهبان چندگانه در چندضلعی ساده از نوع ان‌پی‌سخت است. با این حال در انواع خاصی از چندضلعی‌ها می‌توان یک راه حل چندجمله‌ای برای این مسئله انتظار داشت. زمان اجرای الگوریتم‌های ارایه شده به شکل چندضلعی و همچنین معیار مدنظر برای بهینه‌سازی بستگی دارد. باقری و همکاران [۱۳] الگوریتمی با پیچیدگی زمانی $O(n^2 \cdot \min\{m,n\})$ و پیچیدگی مکانی $O(n \cdot \min\{m,n\})$ برای این مسئله در یک چندضلعی پلکانی با n رأس و m نقطه شروع که ممکن است دامینیت داشته باشند، برای معیار حداقل مجموع ارایه کردند. در این مقاله، ما مسئله را در دو حالت با دامینیت و بدون دامینیت نقاط شروع بررسی کردیم. الگوریتم ارایه شده در این مقاله در مقایسه با الگوریتم ارائه شده در [۱۳]، هم از لحاظ پیچیدگی زمانی و هم از لحاظ پیچیدگی مکانی بهبود داشته است. همچنین ثابت کردیم که در معیار حداقل مجموع، نگهبانان در نقاط شروعی که دامینیت شده‌اند در جواب بهینه بدون حرکت خواهند بود. استفاده از روش برنامه سازی پویا و مدل‌سازی مسئله از دیگر نوآوری‌های این پژوهش است. از روش‌های دیگر مانند روش تقسیم و حل نیز می‌توان استفاده کرد، ولی پیچیدگی زمانی آن بیشتر است. علاوه بر این، ما یک الگوریتم برای گروه‌بندی نقاط شروع در حالت دامینیت ارایه کردیم.

ساختار مقاله به صورت زیر است. در بخش ۲، تعاریف اولیه و خصوصیات تورنگهبان در چندضلعی‌های پلکانی بیان می‌شود. در بخش ۳، یک الگوریتم برای مسئلهٔ تور نگهبان چندگانه در چندضلعی پلکانی در حالت بدون دامینیت ارائه می‌شود. در بخش ٤، یک الگوریتم برای مسئلهٔ تور نگهبان چندگانه در چندضلعی پلکانی در حالت با دامینیت ارایه می‌شود. در بخش ٥، کاربردهای مسئلهٔ تور نگهبان بیان می‌شود. در نهایت در بخش ٦، نتایج پژوهش، نتیجه‌گیری و کارهای آینده بیان خواهد شد.

۲- خصوصیات تورنگهبان در چندضلعی‌های پلکانی

یک چندضلعی مستطیلی همان‌طور که در [١٤] تعریف شده است، یک چندضلعی پلکانی است، اگر قسمت پایین آن شامل یک لبه افقی به نام قاعده و یک لبه عمودی به نام دیواره باشد. شکل ۱، یک چندضلعی پلکانی را نمایش می‌دهد. رأس بین قاعده و دیواره، رأس مبدأ[۹] نامیده می‌شود و قسمت بالای آن شامل لبه‌های افقی و عمودی است که این قسمت از چندضلعی پلکانی را زنجیره مورب می‌نامیم. یک رأس از چندضلعی، رأس محدب است اگر زاویه داخلی آن کوچک‌تر از ۱۸۰ درجه باشد. در این مقاله، تمام رئوس چندضلعی را در جهت ساعتگرد شماره‌گذاری می‌کنیم. رأس مبدا با عدد صفر شماره‌گذاری می‌شود. تمام رئوس با شماره زوج را کنج می‌نامیم.

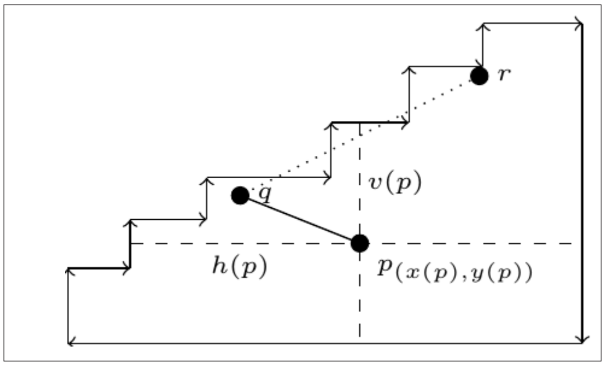


شکل ۱: یک چندضلعی پلکانی با ۱۴ رأس. رئوس در جهت ساعتگرد شماره‌گذاری شده اند. رئوس با شماره زوج محدب است و در اینجا کنج نامیده می‌شوند. قاعده افقی در پایین و لبه عمودی یا دیوار در سمت راست واقع است. رأس شماره صفر رأس مبدأ نامیده می‌شود.

9-Origin

فرض کنید S مجموعه‌ای شامل m نقطه درون چندضلعی P باشد، این نقاط را به عنوان نقاط شروع نگهبانان درون چندضلعی در نظر می‌گیریم. می‌گوییم نقطه p نقطه q را مشاهده(رویت) می‌کند اگر پاره‌خط $[p,q]$ به طور کامل درون چندضلعی P باشد در شکل ۲، نقطه p از نقطه q قابل رویت است ولی نقطه q از نقطه r قابل رویت نیست. در برخی موارد ممکن است برخی از نقاط این پاره خط یا همه نقاط آن روی یال‌های چندضلعی قرار داشته باشند. پاره‌خط‌های افقی و عمودی که از نقطه p عبور می‌کند و درون چندضلعی P قرار دارند را به ترتیب با $h(p)$ و $v(p)$ مشخص می‌کنیم که در شکل ۲ با خطوط خط چین به تصویر کشیده شده است.

به ازای دو نقطه p و q می‌گوییم نقطه q توسط نقطه p دامینیت شده است، اگر $x(p) \geq x(q)$ و $y(p) \leq y(q)$ باشد، یعنی q در گوشه بالا و چپ p قرار داشته باشد (شکل ۲ را ببینید). باید به این نکته توجه داشت که، اگر نقطه q توسط نقطه p دامینیت شده باشد، آنگاه هر نقطه‌ای از چندضلعی که توسط q دیده می‌شود، توسط p نیز دیده می‌شود. لذا هر نگهبان، برای نگهبانی، از نقطه شروع خود می‌تواند به سمت راست و پایین برود، چون موقعیت پایین‌ترین و راست‌ترین در رؤیت آن نگهبان مانند w همه موقعیت‌های دیگر w را تسلط دارد. از این‌رو قضیه زیر را داریم.



شکل ۲: نقطه p نقطه q را مشاهده می‌کند ولی نقطه q نقطه r را مشاهده نمی‌کند.

همچنین نقطه q توسط نقطه q دامینیت شده است و نقطه r توسط نقطه p دامینیت نشده است.

قضیه ۱: یک تورنگهبان با نقطه شروع s و نقطه انتهایی در مختصات (x,y) می‌تواند با پاره خط $[s,(x,y)]$ جایگزین شود. بدون این که مقدار حداقل مجموع در جواب افزایش یابد.

اثبات: از آنجایی که همه تورهای مربوط به نگهبانان به صورت پاره‌خط هستند، یک نگهبان مسیر رفت و برگشتی را روی پاره‌خط مربوطه طی می‌کند. در ادامه فقط طول پاره‌خط (طول مسیر در یک جهت) را محاسبه می‌کنیم و طول تور دو برابر طول پاره‌خطی هست که محاسبه شده است.

قضیه ۲: اگر s و s' دو نقطه شروع باشند به طوری که s بر s' مسلط باشد و w و w' به ترتیب دو تورنگهبان مربوط به نقاط s و s' باشند، آنگاه w' در پاسخ بهینه حداقل مجموع دارای طول صفر خواهد بود.

اثبات: هر نقطه‌ای که توسط s' مشاهده می‌شود، نیز برای s قابل مشاهده است. اگر p یک نقطه ای باشد که هر دو s' و s آن را نمی‌بینند، اما توری که از s' آغاز می‌شود، نقطه p را مشاهده کند، تورنگهبان باید از خط افقی (و همچنین عمودی) که از s عبور می‌کند، عبور نماید. با این حال، فاصله از s به نقطه‌ای که p را مشاهده می‌کند، باید کمتر از فاصله از s' به همان نقطه باشد.

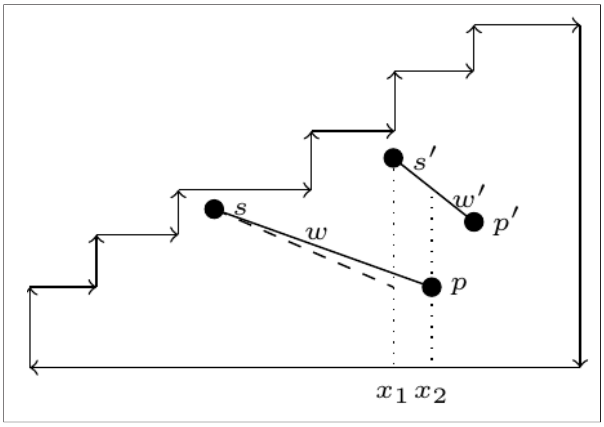
بنابراین با توجه به قضیه ۲، فرض کنید S مجموعه نقاط شروعی باشند که در تسلط هیچ نقطه دیگری نباشند. مجموعه S از یک ترتیب کلی[10] پیروی می‌کند و فرض کنید که این نقاط از پایین و چپ به بالا و راست مرتب شده‌اند، یعنی $s_1 < s_2 < \cdots s_{m-1} < s_m$. اشتراک بین تصویر دو تورنگهبان w و w بر روی قاعده افقی را همپوشان-افقی و اشتراک تصاویر تورهای نگهبان با دیواره عمودی را همپوشان-عمودی می‌نامیم.

قضیه ۳: فرض کنید مجموعه W شامل تورهای نگهبان بهینه باشد و w و w' دو تورنگهبان در این مجموعه باشند، که نقاط شروع مربوط به آنها به ترتیب s و s' باشند. اگر تورهای w و w' طول صفر نداشته باشند، آنگاه این دو

10-Total order

تورنگهبان همپوشان- افقی و همپوشان- عمودی نخواهند داشت.

اثبات: بدون از دست دادن کلیت مسئله، فرض کنید که $s < s'$ باشد و همپوشان- افقی w و w' غیرتهی باشد؛ همانطورکه در شکل ۳ نشان داده شده است. فرض کنید w و w' مجزا از هم باشند (در غیر این‌صورت می‌توان تورها را کوتاه کرد). فرض کنید که p انتهای w و p' انتهای w' باشد. دقت کنید که $x(p) < x(p')$ است در غیر این صورت p بر تور w' مسلط خواهد بود. فرض کنید که همپوشانی در بازه $[x_1, x_2]$ باشد، پس $x(s') = x_1$ و می‌توان w را با $[s, (x(s'), y(p))]$ جایگزین و کوتاه کرد. پاره خط عمودی $[s', (x(s'), y(p))]$ به طور کامل درون P قرار دارد. بنابراین، هیچ کنج محدبی که تورنگهبان w قبل از این دیده بوده است، توسط تور نگهبان‌های جدید w و w' دیده نمی‌شود. با توجه به تقارن، می‌توان نشان داد که این قضیه برای همپوشانی- عمودی نیز اعمال می‌شود. ■



شکل ۳: همپوشانی افقی تورهای W و w' در جواب بهینه و جایگزینی w با $[s,(x(s'), y(p))]$ و کوتاه شدن w که با خط چین نمایش داده شده است.

فرض کنید که P یک چندضلعی پلکانی باشد و فرض کنید که C کنج‌های محدب روی زنجیره مورب چندضلعی باشند. کنج‌های موجود در C را از پایین و چپ به بالا و راست به صورت $c_1 \dots c_{\tilde{n}}$ شماره گذاری می‌کنیم و $\tilde{n} = \frac{n-2}{2}$ است.

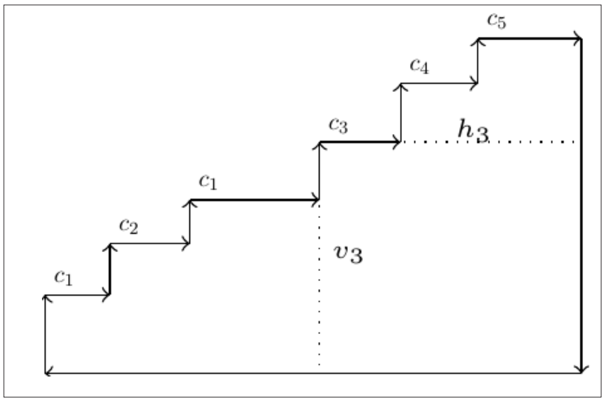
برای یک کنج مانند c_k فرض می‌کنیم که $h_k = h(c_k)$ امتداد لبه افقی و $v_k = v(c_k)$ امتداد لبه عمودی در c_k باشد؛ همان‌طور که در شکل ٤ نشان داده شده است. کنج بعد از یک کنج مانند c در چندضلعی را با $next(c)$ نشان می‌دهیم و فرض می‌کنیم که $next(c_{\tilde{n}}) = c_0$ باشد. اگر c و c' به ترتیب اولین و آخرین کنج‌های یک زیر چندضلعی پلکانی باشند، برش‌های h_c و $v_{c'}$ برش‌های ضروری چندضلعی نامیده می‌شوند. تلاقی این دو برش را با $p_{c,c'}$ نشان می‌دهیم.

قضیه ۴: فرض کنید W یک مجموعه از تورهای نگهبان در جواب بهینه باشند. برای هر کنج محدب c_k در S که دیده نمی‌شود، یا h_k یا v_k ملاقات می‌شود و هیچ امتدادی دو بار ملاقات نمی‌شود و هر تورنگهبان روی یکی از امتدادها پایان می‌یابد.

اثبات: ابتدا نشان می‌دهیم که هر تورنگهبان روی یک امتداد پایان می‌یابد. بدون کاستن از کلیت مسئله، فرض کنید که تورنگهبان w امتداد v_k را در $v_k^{\times}$ قطع کند و این امتداد پایانی مربوط به آن تور باشد. فرض کنید q آخرین نقطه‌ی این تور باشد. وقتی که پاره‌خط $[v_k^{\times}, q]$ را طی می‌کنیم، w هیچ کنج محدب غیر قابل مشاهده اخیر را که در $v_k^{\times}$ ندیده است را نخواهد دید. بنابراین w می‌تواند با پاره خط $[s, v_k^{\times}]$ جایگزین شود که خلاف فرض بهینه بودن است.

حال فرض کنید که برای هر c_k که در مجموعه S دیده نشده است، یا h_k یا v_k ملاقات می‌شود. فرض کنید که تورنگهبان w امتداد v_k را ملاقات کرده است و همان جا متوقف شده باشد. بنابراین w می‌تواند تمام مستطیل‌های بین c_k و رأس مبدأ را مشاهده کند، اما c_{k+1} را نخواهد دید. فرض کنید c_{k+1} توسط w' دیده شده باشد. اگر w' در امتداد h_k حرکت کند، آنگاه هیچ چیزی که w هنوز ندیده باشد را نخواهد دید. برای این که دیدن کنج‌های محدب c_j، $j < k$که توسط w دیده نمی‌شوند، w' باید به سمت پایین حرکت کند تا به نقطه‌ای زیر $v_k^{\times}$ برسد. در این صورت

w و w' همپوشـــان- عمـــودی غیر- تهی خواهند داشـــت که با قضیه ۳ در تناقض اســـت. در نهایـــت اگر یک امتداد دوبار ملاقات شود، لازم است دو نگهبان همپوشان-افقی یا عمودی غیرتهی داشـــته باشند که باز هم با قضیه ۳ در تناقض است.



شکل ۴: شمارش کنج‌های محدب روی زنجیره مورب ($\tilde{n} = 6$) و امتدادهای افقی و عمودی از کنج c_3 که با خطوط نقطه چین نمایش داده شده است.

۳- الگوریتـــم پویای مســـئله در حالت بدون دامینیت

در این بخش، یک راه‌حل برای مسئلهٔ تور نگهبان روی چندضلعـــی پلکانـــی P را در حالتی کـــه نگهبانان دامینیت نداشته باشـــند ارائه می‌دهیم. همچنین در این حالت فرض می‌کنیم که هیچ دو کنج متوالی توســـط دو نگهبان متفاوت نگهبانی نشود. این فرض به این دلیل است که اگر این حالت را در نظر بگیریم، بخشـــی از چندضلعی را می‌توان توسط دو نگهبان به طور همزمان نگهبانی کرد. این وضعیت را در حالت با دامینیت پوشـــش می‌دهیم و نگهبانانی که کنج‌های متوالی را نگهبانی کنند را در یک گروه قرار می‌دهیم. ما کنج c_s را برای یک نقطه شروع به این صورت تعریف می‌کنیم: بالاتریـــن کنجی کـــه از نقطه s قابل دیدن اســـت و نمایش ریاضی آن در معادله شماره (۱) مشخص شده است.

$$c_s = \max_{1 \le i \le m} \{c_i \mid \text{دیده می‌شود توسط } s \ c_i\} \qquad (۱)$$

فرض کنید $\mathcal{L}(c)$ مسئلهٔ تورنگهبان چندگانه روی یک زیرچندضلعی پلکانی باشد که c کنجی از چندضلعی اصلی P و اولین کنج برای زیرچندضلعی است و $c_{\tilde{n}}$ آخرین کنج زیرچندضلعی است. این زیرچندضلعی به گونه‌ای نگهبانی می‌شود که یکی از نگهبانان درون زیرچندضلعی، کنج c و کنجی مانند c' که در سمت راست نگهبان قرار دارد و دیده نشده است (یا برابر c_s باشد) را نگهبانی کند و بقیه کنج‌های بعد از c' توسط سایر نگهبانان در زیرچندضلعی باقی‌مانده نگهبانی شوند که منجر به تولید زیرمسئلهٔ $\mathcal{L}(next(c'))$ می‌شـــود. بنابراین، الگوریتم باید مناســـب‌ترین نگهبان را برای یک کنج مانند c' که چندضلعی را تفکیک می‌کند، پیدا کند تا حداقل مجموع بهینه را تولید نماید. معادله بازگشتی $\mathcal{L}(c)$ را در رابطه شماره (۲) ارائه کردیم.

(۱)

$$\mathcal{L}(c) = \begin{cases} 0, & if\ c = c_0 \\ +\infty, & if\ c \neq c_0\ and\ \tilde{S}_{c.c'} = \emptyset \\ \min\limits_{\substack{c' \in [c, c_{\tilde{n}}] \\ \tilde{S}_{c,c'} \neq \emptyset, find\ s^* \in \tilde{S}_{c,c'} \\ c' = c_{s^*}\ or\ c'\ is\ unseen\ and\ x(c') > x(s^*)}} \{ms(s^*.c, c') + \mathcal{L}(next(c'))\}. & otherwise \end{cases}$$

که $\tilde{S}_{c,c'}$ یک لیســـت از نقاط شـــروع دامینیت نشده در زیرچندضلعـــی محصور بین v_c و $h_{c'}$ اســـت. فرض کنید $p_{c,c'}$ نقطه تلاقی برش‌های h_c و $v_{c'}$ باشـــد. s^* نزدیکترین نقطه به $p_{c,c'}$ که گوشـــه ســـمت چپ و بالای هسته است. همچنیـــن $ms(s^*, c, c')$ فاصله مورب نقطه s^* از هســـته اســـت که در ادامه توضیح داده می‌شود. اگر s^* در سمت چپ و بالای $p_{c,c'}$ باشد مقدار $ms(s^*, c, c')$ برابر فاصله s^* و $p_{c,c'}$ اســـت. اگر s^* در ســـمت راســـت و بالای $p_{c,c'}$ باشـــد مقدار $ms(s^*, c, c')$ برابر فاصلـــه s^* و برش افقی h_c اســـت. اگر s^* در ســـمت چپ و پایین $p_{c,c'}$ باشد مقدار $ms(s^*, c, c')$ برابر فاصله s^* و برش عمودی $v_{c'}$ است و در غیر این صورت اگر s^* در ســـمت راست و پایین $p_{c.c'}$ باشـــد $ms(s^*, c, c')$ برابر صفر اســـت. برای یافتن s^* از نمودار ورونوی [۱۵] نقاط شروع استفاده می‌کنیم. این کار در ابتدای الگوریتم انجام می‌شـــود و به یک زمان از درجه

$O(m \cdot \log m)$ و حافظه از درجه $O(m)$ احتیاج دارد [۱۵]. جستجو در ساختار ورونوی برای پیدا کردن نزدیک‌ترین نقطه به زمان $O(\log m)$ نیاز دارد [۱۵].

اگر مقدار c طوری باشد که با یک زیرچندضلعی تهی روبه‌رو شویم، مقدار $\mathcal{L}(c)$ برابر صفر است و اگر زیرچندضلعی حاوی هیچ نقطه شروعی نباشد مقدار $\mathcal{L}(c)$ برابر بی‌نهایت است. در غیراین‌صورت با ثابت نگه داشتن برش افقی از کنج c، کنج c' را که از c شروع می‌شود و برش عمودی را مشخص می‌کند، شمارش می‌کنیم. اگر بازه‌ای که توسط c و c' محدود شده است حاوی نقاط شروع باشد ($\tilde{S}_{c,c'} \neq \emptyset$)، آنگاه نزدیک‌ترین نقطه به هسته را توسط ساختار ورونوی پیدا می‌کنیم و اگر این نقطه در بازه موردنظر باشد و کنج c' یک کنج دیده نشده در سمت راست s^* باشد یا کنج c' آخرین کنجی باشد که از نقطه s^* رویت می‌شود، آنگاه این تفکیک قابل قبول بوده و الگوریتم ادامه پیدا می‌کند. اگر نزدیک‌ترین نقطه در بازه مورد نظر نباشد، یعنی پاسخ بهینه در بازه دیگری قرار دارد و این بازه نیاز به بررسی ندارد.

قضیه ۵: الگوریتم ارایه شده در معادله (۲) یک جواب بهینه برای مسئلهٔ تورنگهبان چندگانه در چندضلعی پلکانی ارائه می‌دهد. این الگوریتم با پیچیدگی زمانی $O(n^2 \cdot \log m)$ و پیچیدگی مکانی $O(n)$ عملیات محاسباتی را انجام می‌دهد.

اثبات: در هر تکرار الگوریتم نزدیک‌ترین نقطه به هسته (s^*) به کمک دیاگرام ورونوی جستجو می‌شود که دارای پیچیدگی زمان $O(\log m)$ است. لذا، یک کنج مانند c' در سمت راست وجود دارد که نگهبان مستقر در نقطه s^* باید برش عمودی آن را نگهبانی کند. تعداد این کنج‌ها از درجه $O(n)$ است. بنابراین هر تکرار الگوریتم از درجه $O(n \cdot \log m)$ است. از طرف دیگر، برای ذخیره جواب $\mathcal{L}(c)$ به تعداد کنج‌های موجود از درجه $O(n)$ حافظه مورد نیاز است. پس الگوریتم فوق دارای پیچیدگی زمان $O(n^2 \cdot \log m)$ و پیچیدگی مکانی $O(n)$ است.■

۴- الگوریتم پویای مسئله در حالت با دامینیت

در این بخش، الگوریتم حل مسئله را در حالتی که نقاط شروع نگهبانان با هم دامینیت داشته باشند، شرح می‌دهیم. بنابر قضیه ۲، طول تورنگهبان مربوط به نگهبانان دامینیت شده در جواب نهایی صفر است. بنابراین همان‌طور که قبلا گفته شد بنابر قضیه ۲، نگهبانان دامینیت شده را می‌توانیم از فضای مسئله حذف کنیم. اما مشکل اصلی در این حالت مشاهده شدن ترتیبی از کنج‌های مجاور هم در چندضلعی توسط دو یا چند نگهبان است. در این حالت، تمام نگهبانانی که کنج‌های مجاور هم را نگهبانی می‌کنند را در یک گروه قرار می‌دهیم. پس در نهایت، تعدادی گروه ایجاد می‌شود که هر گروه شامل تعدادی نقاط شروع مربوط به نگهبانان است.

۴-۱- گروه‌بندی نقاط شروع

در حالت با دامینیت مسئله، لازم است نقاط شروع دامینیت شده را حذف کنیم و نقاط شروعی که کنج‌های متوالی را در ابتدا نگهبانی می‌کنند، در یک گروه قرار دهیم. در نهایت با گروه‌های به دست آمده مسئله را حل می‌کنیم. برای این کار ابتدا m نقطه شروع ورودی که در لیست S قرار دارند را بر اساس مختصات y به صورت صعودی مرتب می‌کنیم. اگر دو نقطه دارای مختصات y یکسانی باشند، نقطه‌ای که مختصات x بیشتری دارد را مقدم می‌کنیم. چون نقطه‌ای که در سمت راست باشد بر نقاط سمت چپ خود تسلط دارد. عمل مرتب سازی دارای پیچیدگی زمانی $O(m \cdot \log m)$ است.

باید مختصات x آخرین نقطه دامینیت نشده را داشته باشیم تا بر اساس آن نقاط دامینیت نشده بعدی را تشخیص دهیم. در ابتدا x را با مقدار $-\infty$ مقدار دهی می‌کنیم. یک نقطه جدید دامینیت نشده است اگر مختصات x آن از x بیشتر باشد. بنابراین اولین نقطه در لیست مرتب شده مکان نگهبانی است که دامینیت نشده است. لیست مرتب شده S را از ابتدا تا انتها بررسی می‌کنیم. اگر نقطه تحت بررسی (مانند s) دامینیت شده باشد، پردازشی

ندارد؛ در غیراین‌صورت اگر نقطه s دامینیت نشده باشد، با پیمایش کنج‌ها، اولین کنجی که توسط s نگهبانی می‌شود را پیدا می‌کنیم و به عنوان مشاهده شده علامت می‌زنیم. اگر کنجی که توسط s نگهبانی می‌شود در مجاورت آخرین کنجی که نقطه قبلی دامینیت نشده، نگهبانی می‌کرد، نباشد، به این معنا است که نقطه s شروع یک گروه جدید از نگهبانان است. در این صورت، با ایجاد یک گروه جدید، نقطه s را به این گروه اضافه می‌کنیم. در غیر این‌صورت نقطه s را به آخرین گروه اضافه می‌کنیم. سپس با یک پیمایش دیگر در ادامه کنج‌ها، تمام کنج‌های بعدی که نقطه s مشاهده می‌کند را به عنوان مشاهده شده علامت‌گذاری می‌کنیم. همچنین مختصات x نقطه s را برای بررسی‌های بعدی در x ذخیره کرده و عملیات را تکرار می‌کنیم تا به انتهای لیست S برسیم. عملیات بررسی نقاط دارای پیچیدگی زمان $O(n+m)$ است. بنابراین، در نهایت، عمل گروه‌بندی پیچیدگی زمانی $O(n+m\cdot\log m)$ را خواهد داشت، که m تعداد نقاط شروع ورودی و n تعداد رئوس چندضلعی پلکانی ورودی است.

۴-۲- تورنگهبان در یک گروه

یک گروه مانند g از نگهبانان را در نظر بگیرید. اولین نگهبان در این گروه را با s_g^+ و آخرین نگهبان در گروه را با s_g^- نشان می‌دهیم. مسئلهٔ تور نگهبان در یک زیرچندضلعی پلکانی که شامل یک گروه از نگهبانان باشد را با $msg(g,s^*,c,c')$ نمایش می‌دهیم که g شماره آن گروه در چندضلعی اصلی و c و c' اولین و آخرین کنج در زیرچندضلعی هستند. h_c و $v_{c'}$ باید توسط این گروه نگهبانی شوند تا کل زیرچندضلعی نگهبانی شود. s^* نزدیک‌ترین نقطه به هسته می‌باشد که قبلا محاسبه شده است.

همان‌طور که در معادله (۳) مشخص شده است یک زیر چندضلعی شامل یک گروه می‌تواند به یکی از دو حالت زیر نگهبانی شود:

۱. نگهبانی توسط یکی از نگهبانان که به هسته نزدیک‌تر است. ما نزدیک‌ترین نگهبان به هسته را با s^* نشان می‌دهیم.

۲. نگهبانی به صورت مشترک توسط دو نگهبان ابتدایی s_g^+ و انتهایی s_g^- در گروه مربوطه. که s_g^+ اگر در پایین برش h_c نباشد، با حرکت عمودی خود، برش h_c را پوشش می‌دهد و s_g^- اگر در سمت راست برش $v_{c'}$ نباشد، با حرکت افقی خود $v_{c'}$ را پوشش می‌دهد.

(۳)

$$msg(g,s^*,c,c') = \min\{ms(s^*,c,c').\mathrm{v}(s_g^+,c) + \mathrm{h}(s_g^-,c')\}.$$

در معادله (۳)، $\mathrm{v}(s_g^+,c)$ فاصله از s_g^+ به برش h_c است. این فاصله وقتی که s_g^+ در پایین h_c باشد برابر صفر است و $h(s_g^-,c')$ فاصه از نقطه s_g^- به برش $v_{c'}$ است. این فاصله وقتی که s_g^- در سمت راست $v_{c'}$ باشد برابر صفر است. برای پیدا کردن نقطه s^* در یک گروه می‌توانیم از ساختار ورونوی نقاط شروع استفاده کنیم. ما فرض می‌کنیم که s^* قبلا محاسبه شده است. بنابراین روال محاسباتی مربوط به معادله (۳) دارای پیچیدگی زمانی $O(1)$.

اگر تعداد نگهبانان را با m نشان دهیم، لازم است یک ساختار ورونوی از روی نقاط تشکیل گردد. برای این کار، الگوریتم اصلی ایجاد نمودار ورونوی را روی m نقطه اجرا می‌کنیم. بنابراین نیازمند یک زمان از درجه $O(m\cdot\log m)$ و یک مکان از درجه $O(n)$برای ایجاد ساختار ورونوی هستیم [۱۵].

۴-۳- الگوریتم حداقل مجموع با دامینیت

فرض کنید $\mathcal{L}^*(c)$ مساله تورنگهبان چندگانه روی یک زیرچندضلعی پلکانی در حالت با دامینیت باشد که c کنجی از چندضلعی اصلی P و اولین کنج برای زیرچندضلعی است و $c_{\tilde{n}}$ آخرین کنج زیرچندضلعی است. این زیرچندضلعی به این صورت نگهبانی می‌شود که یکی از گروه‌های درون زیرچندضلعی، کنج c و کنجی مانند c' که در سمت راست گروه قرار دارد و دیده نشده است (یا برابر $c_{s_g^-}$ باشد) را نگهبانی کند و بقیه کنج‌های بعد از c' توسط سایر گروه‌ها

در زیرچندضلعی باقیمانده نگهبانی شوند که منجر به تولید زیرمسئله $\mathcal{L}^*(next(c'))$ می‌شود. بنابراین مسئله باید مناسب‌ترین c' را برای این تفکیک پیدا کند تا حداقل مجموع بهینه را تولید نماید. معادله بازگشتی $\mathcal{L}^*(c)$ را در رابطه شماره (٤) ارائه کردیم.

در معادله (٤)، $c_{s_g^-}$ بالاترین کنجی را مشخص می‌کند که توسط s_g^- مشاهده می‌شود و از معادله (١) محاسبه می‌شود. الگوریتم پویای مسئله در حالت با دامینیت نیز چندضلعی را به صورت بازگشتی به دو زیرچندضلعی مستقل تقسیم می‌کند. زیرچندضلعی سمت چپ توسط یکی از گروه‌ها نگهبانی می‌شود. سپس کمینه مجموع طول تورها در هر زیرچندضلعی سمت راستی به صورت بازگشتی با گروه‌های موجود در آن محاسبه می‌شود. اگر مقدار c طوری باشد که با یک زیرچندضلعی تهی روبه‌رو شویم مقدار $\mathcal{L}^*(c)$ برابر صفر است و اگر زیرچندضلعی حاوی هیچ نقطه شروعی نباشد مقدار $\mathcal{L}^*(c)$ برابر بی‌نهایت است. در غیر این صورت با ثابت نگه داشتن برش افقی از کنج c، کنج c' را که از c شروع می‌شود و برش عمودی را مشخص می‌کند، شمارش می‌کنیم. اگر بازه‌ای که توسط c و c' محدود شده است حاوی نقاط شروع باشد ($\tilde{S}_{c,c'} \neq \emptyset$)، آنگاه نزدیک‌ترین نقطه به هسته را توسط ساختار ورونوی پیدا می‌کنیم و گروه مربوط به نقطه s^* را می‌یابیم، که در معادله (٤) با $group-of(s^*)$ مشخص شده است.

اگر نزدیک‌ترین نقطه در بازه موردنظر باشد و کنج c' یک کنج دیده‌نشده در سمت راست آخرین نقطه گروه مربوطه باشد یا کنج c' آخرین کنجی باشد که از آخرین نقطه گروه رویت می‌شود، آنگاه این تفکیک قابل قبول بوده و الگوریتم ادامه پیدا می‌کند. اگر نزدیک‌ترین نقطه در بازه موردنظر نباشد، یعنی پاسخ بهینه در بازه دیگری قرار دارد و این بازه نیاز به بررسی ندارد.

$$\mathcal{L}^*(c) = \begin{cases} 0, \quad if\ c = c_0 \\ +\infty, \quad if\ c \neq c_0\ and\ \tilde{S}_{c.c'} = \emptyset \\ \min\limits_{\substack{c' \in [c, c_{\tilde{n}}].\tilde{S}_{c,c'} \neq \emptyset \\ find\ s^* \in \tilde{S}_{c,c'}.g = group-of(s^*) \\ c' = c_{s_g^-}\ or\ c'\ is\ unseen\ and\ x(c') > x(s_g^-)}} \{msg(g, s^*, c, c') + \mathcal{L}^*(next(c'))\}. \quad otherwise \end{cases} \tag{٤}$$

قضیه ۶: الگوریتم ارایه شده در معادله (٤) یک جواب بهینه برای مسئلۀ تورنگهبان چندگانه در چندضلعی پلکانی ارائه می‌دهد. این الگوریتم با پیچیدگی زمانی $O(n^2 \cdot \log m)$ و پیچیدگی مکانی $O(n)$ عملیات محاسباتی را انجام می‌دهد.

اثبات: در هر تکرار الگوریتم، نزدیکترین نقطه به هسته (یعنی s^*) به کمک نمودار ورونوی جستجو می‌شود که دارای پیچیدگی زمان $O(\log m)$ است. این نقطه در یگ گروه مانند g عضویت دارد. برای هر کنج مانند c' بعد از c، یکی از نزدیک‌ترین نگهبانان مستقر در بازه مربوطه یا آخرین نگهبان در گروهی که نزدیک‌ترین نگهبان در آن وجود دارد، باید برش عمودی آن را نگهبانی کند. تعداد این کنج‌ها از درجه $O(n)$ است. بنابراین، با توجه به این که هر جستجو برای یافتن نزدیک‌ترین نقطه به هسته $O(\log m)$ زمان نیاز دارد، هر تکرار الگوریتم از درجه $O(n \cdot \log m)$ است. از طرف دیگر برای ذخیره جواب $\mathcal{L}(c)$ به تعداد کنج‌های موجود، از درجه $O(n)$ حافظه مورد نیاز است. بنابراین، الگوریتم فوق دارای پیچیدگی زمان $O(n^2 \cdot \log m)$ و پیچیدگی مکانی $O(n)$ است.

۵- کاربردهای مسئلۀ تورنگهبان

مسئلۀ تور نگهبان دارای کاربردهای متنوعی است. از جمله این کاربردها می‌توان به موارد زیر اشاره کرد. در زمینه سیستم‌های نظارتی، مسئلۀ تورنگهبان را می‌توان برای بهینه‌سازی مسیرهای گشت‌زنی پرسنل امنیت و دوربین‌های نظارتی استفاده کرد. به طوری که بیشترین

پوشش منطقه را با کمترین طول گشت‌زنی داشته باشیم. بدین‌ترتیب با صرف هزینه و زمان کمتر، کارآمدی و اثربخشی سیستم‌های نظارتی بیشتر خواهد بود. همچنین مسئلهٔ تور نگهبان در عملیات جستجو و نجات قابل به کارگیری است. در این جا، هدف طی کردن کوتاه‌ترین مسیری است که بیشترین پوشش‌دهی نقاط مهم را داشته باشد به طوری که شانس پیدا کردن افراد یا اشیاء گم شده بیشتر شود. با حل مسئلهٔ تور نگهبان، تیم‌های جستجو می‌توانند مسیرهای خود را برای اطمینان از پوشش دقیق منطقهٔ جستجو برنامه‌ریزی کنند. به علاوه در مدیریت تاسیسات، می‌توان مسیر کارکنان نگهداری یا نظافت را بهینه‌سازی کرد تا اطمینان حاصل کنیم که همه منطقه شامل تاسیسات، به درستی تحت پوشش بوده و نظافت می‌شود. این کار باعث صرفه‌جویی در زمان و انرژی مصرفی می‌شود و کیفیت سرویس نیز بهتر خواهد شد. یکی دیگر از کاربردهای مسئلهٔ تور نگهبان، در رباتیک است [١٦]. در این جا، هدف می‌تواند بهینه‌سازی حرکت ربات‌ها برای پوشش دادن یک ناحیه مشخص در محیط باشد. به طور خاص این مسئله برای ربات‌های نظافت کف، اتوماسیون انبار و ربات‌های اکتشافی مفید خواهد بود.

۶- نتیجه‌گیری و کارهای آینده

در این مقاله، مسئلهٔ تور نگهبان چندگانه ثابت در چندضلعی‌های پلکانی که نوع خاصی از چندضلعی‌های مستطیلی هستند را مورد بررسی قرار دادیم و یک الگوریتم پویا برای حل بهینه این مسئله در معیار حداقل مجموع ارایه کردیم. همچنین نشان دادیم که نگهبانانی که نقطه شروع آنان در تسلط باشد، قابل چشم‌پوشی و حذف از مسئله هستند. زیرا ثابت کردیم که تورنگهبانی آنان در جواب بهینه در حالت حداقل مجموع برابر صفر است. الگوریتم پویای ارایه شده در حالت بدون دامینیت و هم در حالت با دامینیت دارای پیچیدگی زمانی $O(n^2 \cdot \log m)$ است. همچنین الگوریتم‌های ارایه شده در هر دو حالت دارای پیچیدگی مکانی $O(n)$ می‌باشند، که m و n به ترتیب تعداد نگهبانان و تعداد رئوس چندضلعی است. برای کارهای آتی می‌توان این مسئله در معیار حداقل حداکثر را بررسی کرد.

فهرست منابع

[1] H. E. G. a. D. Avis., “A linear algorithm for computing the visibility polygon from a point,” Journal of Algorithms, vol. 2, no. 2, p. 186 – 197, 1981.

[2] V. Chvátal, “A combinatorial theorem in plane geometry. Journal of Combinatorial Theory,” Journal of Combinatorial Theory, Series B, vol. 18, no. 1, p. 39 – 41, 1975.

[3] S. Fisk, “A short proof of chvátal’s watchman theorem,” Journal of Combinatorial Theory, Series B, vol. 24, no. 3, p. 374, 1978.

[4] D.T. Lee and A.K. Lin, “Computational Complexity of Art Gallery Problems,” Springer New York, p. 303–309, 1990.

[5] B.J. Nilsson and E. Packer, “An approximation algorithm for the two-watchman route in a simple polygon,” in In Proceedings of EuroCG 2016, EuroCG, 2016.

[6] J.S.B. Mitchell, “Approximating Watchman Routes,” in Society for Industrial and Applied Mathematics, New Orleans, Louisiana, 2013.

[7] S. Carlsson, B.J. Nilsson, and S. Ntafos, “Optimum guard covers and m-watchmen routes for restricted polygons,” International Journal of Computational Geometry \& Applications, vol. 03, no. 01, pp. 85-105, 1993.

[8] B.J. Nilsson, and D. Wood, “Optimum Watchmen Routes in Spiral Polygons: Extended Abstract,” in Proceedings 2nd Canadian Conference in Computational Geometry, Ottawa, 1990.

[9] B.J. Nilsson, and D. Wood, Watchmen Routes in Spiral Polygons, Technical Report LU-CS-TR:90-55, Dept. of Computer Science, Lund University, 1990.

[10] B. J. Nilsson and S. Schuierer, “Shortest m-watchmen routes for histograms: the minmax case,” in Proceedings ICCI `92: Fourth International Conference on Computing and Information, 1992.

[11] J.S.B. Mitchell and E.L. Wynters, “Watchman routes for multiple guards,” in Proc. 3rd CCCG, 1991.

[12] E. Packer, “Computing Multiple Watchman Routes,” in McGeoch, C.C. (eds) Experimental Algorithms. WEA 2008. Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008.

[13] A. Bagheri, A. Brötzner, F. Farivar, R. Ghasemi, F. Keshavarz-Kohjerdi, E. Krohn, B.J. Nilsson, C. Schmidt, “Minsum m watchmen’s routes in Stiegl polygons,” in XX Spanish Meeting on Computational Geometry, Spain, 2023.

[14] M. Bousquet-Melou, A.J. Guttmann, W.P. Orrick, and A. Rechnitzer, “Inversion relations, reciprocity and polyominoes,” Annals of Combinatorics, vol. 3, p. 223–249, 1999.

[15] M. de Berg, O. Cheong, M. Kreveld, and M. Overmars, Computational Geometry: Algorithms and Applications, 3rd ed. ed., Santa Clara, CA, USA: Springer-Verlag TELOS, 2008.

[16] M. Dunbabin and L. Marques, “Robots for Environmental Monitoring: Significant Advancements and Applications,” IEEE Robotics & Automation Magazine, vol. 19, no. 1, pp. 24-39, 2012.