

الگوریتمی سریع برای پوشش دید مستطیلی چندضلعی‌های متعامد ساده با حداقل تعداد r-Star ها

کیوان برنا؛ دانشگاه خوارزمی، دانشکده علوم ریاضی و کامپیوتر

پذیرش ۹۶/۱۱/۲۸

دریافت ۹۶/۰۷/۰۱

چکیده

این مقاله الگوریتمی برای پوشش دید چندضلعی‌های متعامد ساده با حداقل تعداد نگهبانان به دست می‌دهد. در واقع حداقل تعداد نگهبانان را برای چندضلعی‌های ساده (بدون حفره) متعامد برای همه حالت‌ها بررسی کرده و قادر هستیم که برای هر یک از نگهبانان نیز محدوده مستطیل شکلی را بیابیم. به عبارت دیگر مسئله پوشش چندضلعی‌های متعامد ساده با حداقل r-Star ها را بررسی می‌کنیم. در هر چندضلعی متعامد P دو نقطه p و q، نسبت به هم r-visible هستند اگر و تنها اگر آن دو نقطه را دو گوشه مخالف مستطیلی در نظر بگیریم، تمام مستطیل درون چندضلعی P قرار داشته باشد. حال یک چندضلعی P را یک r-Star گوئیم اگر یک نقطه p در آن وجود داشته باشد به طوری که هر نقطه q عضو چندضلعی، از p، r-visible باشد. در این مقاله الگوریتمی را پیشنهاد می‌کنیم که روی همه چندضلعی‌های ساده متعامد کاربرد دارد و قادر است حداقل تعداد نگهبانان را در جای خود مستقر کند. این الگوریتم با استفاده از روشی به نام مستطیل‌بندی (تقسیم چندضلعی متعامد به تعدادی مستطیل)، تعدادی از r-Star ها را افزایش کرده و به پردازش آن‌ها برای درج نگهبانان در محل خود برای رسیدن به هدف، که حداقل تعداد نگهبانان است می‌پردازد. الگوریتم پیشنهادی ما قادر است تا در زمان $O(n^5)$ حداقل تعداد نگهبانان را به همراه محدوده مستطیل شکلی برای آن‌ها تعیین کند در حالی که مرتبه اجرایی بهترین الگوریتم‌های موجود قبلی $O(n^{17} \text{ poly} - \log n)$ بوده است. از دیگر مزایای این الگوریتم می‌توان به نداشتن محدودیت در چندضلعی‌های متعامد ساده اشاره کرد.

واژه‌های کلیدی: چندضلعی متعامد (راست گوشه)، مستطیل‌بندی، r-Star، r-visible، افزایش r-Star ها.
رده بندی موضوعی: 68U05, 52B55.

۱. مقدمه

مسائل گالری هنر مسئله‌ای پژوهشی بزرگ و پرکاربرد در هندسه محاسباتی و ترکیبی است [۷]، [۸]. به دست آوردن حداقل تعداد نگهبان برای تماشای یک چندضلعی داده شده NP-Complete است [۱]. این مسئله را Keil و Worman بررسی کرده‌اند [10] که یک الگوریتم با مرتبه زمانی $O(n^{17} \text{ poly} - \log n)$ به طوری که n اندازه چندضلعی داده شده است توصیف شده است و از روش افزایش چندضلعی P به چندضلعی‌های کوچک‌تر بهره می‌گیرد. بدین صورت که یک افزایش از چندضلعی P مجموعه‌ای از تعدادی چندضلعی است که اجتماع هندسی آن‌ها دقیقاً برابر P است. حال که مسائل نگهبانی روی چندضلعی‌های متعامد مطرح شده است، (یعنی چندضلعی‌هایی که اضلاع آن افقی یا عمودی‌اند، از این‌رو، یال‌هایش می‌توانند به صورت یال‌های S، N، E و W نام‌گذاری شوند) باعث شده تا بتوان برای این‌گونه مسائل الگوریتم‌هایی برای پوشش دید بهینه ارائه کرد.

*نویسنده مسئول borna@khu.ac.ir

در این مقاله الگوریتمی برای پوشش دید چندضلعی‌های متعامد با حداقل تعداد نگهبان با میدان دید r یا همان r -visibility ارائه می‌شود که ایده پیشنهادی، حداقل تعداد نگهبان را برای چندضلعی‌های ساده متعامد (بدون محدودیت) بررسی می‌کند و قادر است برای هر یک از نگهبانان نیز محدوده مستطیل شکلی تعیین کند. در الگوریتم پیشنهادی این مقاله باید ابتدا چندضلعی متعامد را به چندین چندضلعی متعامد کوچک‌تر که هر کدام یک r -Star است تقسیم کنیم که باعث به وجود آمدن روشی به نام مستطیل‌بندی می‌شود. مستطیل‌بندی به نوبه خود دارای مراحل و قسمت‌های مختلفی است که در بخش ۴-۵ شرح داده می‌شود. به وضوح مسئله تعیین حداقل مجموعه‌ای از نگهبانان با قابلیت میدان دید r برای تماشای یک چندضلعی متعامد ساده، برابری می‌کند با تعیین حداقل پوشش همان چندضلعی به وسیله r -Star ها. در بخش بعدی خلاصه‌ای از کارهای پیشین را ملاحظه می‌کنید. در بخش سوم تعاریف و اصطلاحات مربوط به این مقاله از قبیل فرورفتگی، انواع یال‌ها، میدان دید r -Star، r و غیره بیان می‌شود. در بخش چهارم به شرح کامل الگوریتم پیشنهادی می‌پردازیم. در بخش پنجم مقایسه الگوریتم پیشنهادی با کارهای انجام شده و بخش ششم و پایانی به نتیجه‌گیری اختصاص داده شده است.

۲. کارهای پیشین

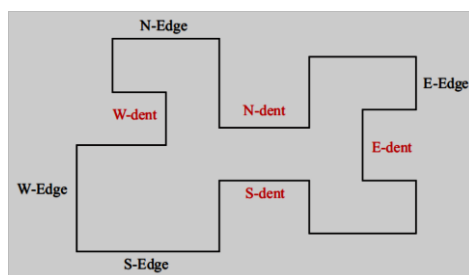
در مقاله‌های مشابه با این موضوع کارهایی صورت گرفته که هر کدام فاکتور خاصی را برای رسیدن به هدف در نظر داشته‌اند. برای $0 \leq k \leq 4$ یک چندضلعی متعامد در رده k - است هرگاه فرورفتگی‌های آن حداکثر در k جهت باشد. چندضلعی‌های رده ۲ می‌توانند به دو رده $2a$ و $2b$ تقسیم شوند. فرورفتگی‌های رده $2a$ در راستاهای موازی (شمال و جنوب) یا (شرق و غرب) است در حالی که فرورفتگی‌های رده $2b$ به صورت متعامد است. توجه کنید که چندضلعی‌های متعامد واقع در رده $2a$ یکنوا هستند.

مسئله پوشش دید مستطیلی را Leonidas Palios و Petros Tzimas برای چندضلعی‌های متعامد ساده رده-۳ پوشش داده‌اند [12]، که در مرتبه زمانی $O(n+k \log k)$ صورت می‌گیرد. Keil یک الگوریتم با مرتبه زمانی $O(n^2)$ را برای پوشش یک چندضلعی متعامد رده- $2a$ به وسیله r -Starها توصیف کرده است [4]. پیچیدگی زمانی پوشش چندضلعی متعامد رده $2a$ با r -Starها را Keil، Gewail و Ntalos بهبود داده‌اند [3]، که یک الگوریتم با مرتبه زمانی $O(n)$ برای گزارش محل‌های یک مجموعه از حداقل تعداد نگهبانان را می‌دهد. یک بهبود از این الگوریتم در [۵] داده شده است که Keil و Worman به مسئله پوشش چندضلعی‌های متعامد عمومی با r -Starها پرداخته‌اند و از رویکرد تئوری گراف استفاده شده در [۶] بهره گرفته است (برای پوشش‌های s -Star)، تا یک الگوریتم با مرتبه زمانی $O(n^{17} \text{ poly} - \log n)$ شرح داده شود [۱۰]. در اکثر مقاله‌های ذکر شده، مسئله پوشش r -Star برای چندضلعی‌های متعامد رده‌های خاصی بررسی شده است ولی الگوریتم پیشنهادی در این مقاله همه چندضلعی‌های متعامد ساده، شامل هر رده‌ای را قادر است پوشش دهد و حداقل تعداد نگهبانان با میدان دید r را با محدوده قرارگیری مناسب پیدا کند.

۳. تعاریف پایه در الگوریتم پیشنهادی

در این مقاله پردازش روی چندضلعی‌های متعامد یا راست گوشه انجام می‌شود پس فقط چهار نوع یال داریم. اگر چندضلعی مورد نظر را به صورت گونیا نسبت به صفحه مورد پردازش فرض کنید یال شمالی یا همان یال N ، یالی است

که داخل چندضلعی پایین آن قرار دارد. یال جنوبی یا S که درون چندضلعی بالای آن قرار دارد. یال شرقی یا E درون چندضلعی را در سمت چپ خود دارد و یال غربی یا W داخل چندضلعی در سمت راستش قرار می‌گیرد (شکل ۱). هم‌چنین چهار نوع فرورفتگی وجود دارد که عبارتند از فرورفتگی شمالی (N-dent)، فرورفتگی جنوبی (S-dent)، فرورفتگی شرقی (E-dent) و فرورفتگی غربی (W-dent) که در شکل ۱ نشان داده شده است. در الگوریتم پیشنهادی پس از مستطیل‌بندی یال‌ها و رؤس جدیدی در چندضلعی متعامد P ایجاد می‌شوند که آن‌ها را یال‌های فرعی و رؤس فرعی می‌نامیم. یال‌های فرعی به‌صورت یال‌های فرعی افقی و عمودی شناخته می‌شوند و ممکن است در مواردی برخی از یال‌های فرعی افقی را یال‌های شمالی فرعی بنامیم و رؤس فرعی نیز از تقاطع یال‌های فرعی با یال‌های اصلی یا فرعی ایجاد می‌شوند که تمام این نقاط تقاطع را رؤس فرعی جنوبی می‌نامیم. چندضلعی‌های متعامد می‌توانند بر حسب نوع فرورفتگی‌ها طبقه‌بندی شوند [۲] که یک چندضلعی متعامد رده k (class-k) به طوری که $1 \leq k \leq 4$ ، به این‌صورت تعریف می‌شود که فرورفتگی‌هایش حداکثر در k جهت مختلف باشند. چندضلعی‌های رده ۲ می‌توانند مجدداً رده‌بندی شوند. در رده‌های $2a$ به طوری که ۲ جهت فرورفتگی موازی باشند (یعنی N و S یا E و W) و رده $2b$ به طوری که ۲ جهت فرورفتگی عمود بر یک‌دیگر باشند.



شکل ۱. یک چندضلعی متعامد ساده با انواع یال‌ها و فرورفتگی‌ها

میدان دید نگهبانان را به‌صورت r -visibility یعنی میدان دید مستطیلی که این‌گونه تعریف می‌کنیم: (شکل ۲ ب) در هر چندضلعی متعامد P دو نقطه p و q نسبت به هم r -visible هستند اگر و فقط اگر آن دو نقطه را دو گوشه مخالف مستطیلی در نظر بگیریم تمام مستطیل درون چندضلعی P قرار داشته باشد. حال یک چندضلعی P را یک r -Star گوئیم اگر یک نقطه p در آن وجود داشته باشد به طوری که هر نقطه q عضو چندضلعی، از p، r -visible باشد. شکل ۲ ب در واقع یک r -Star است و هسته آن که همان مستطیل نگهبان است که با یک مستطیل خط‌چین نشان داده شده است. در مسئله r -Star فرض می‌کنیم که هر نگهبان میدان دید ۳۶۰ درجه دارد.

۴. الگوریتم پیشنهادی

در این مقاله ایده جدیدی بر مبنای مستطیل‌بندی چندضلعی متعامد مطرح می‌شود که باعث ایجاد یال‌های جدیدی در آن می‌شود. در واقع چندضلعی متعامد اولیه P را با روشی موسوم به مستطیل‌بندی که در بخش ۴-۵ توضیح داده خواهد شد به تعدادی مستطیل افراز می‌کنیم به طوری که اجتماع هندسی آن‌ها دقیقاً برابر چندضلعی اولیه P است. از این رو یال‌ها را بعد از مستطیل‌بندی در دو گروه یال‌های اصلی و یال‌های فرعی قرار داده که در شکل‌ها، یال‌های فرعی را با خط‌چین نمایش می‌دهیم. بدیهی است که یال اشتراکی بین دو مستطیل افراز شده از نوع فرعی بوده و مانع دید نگهبان نخواهد شد. ابتدا طبق روشی که در بخش ۴-۵ این مقاله گفته خواهد شد چندضلعی اولیه را

مستطیل‌بندی کرده و سپس طبق روش افراز چندضلعی متعامد که در زیر بخش بعدی آمده است، چندضلعی اولیه P را در صورت امکان به دو یا چند، چندضلعی متعامد تقسیم می‌کنیم که هر کدام از آن‌ها یک r -Star خواهد بود. سپس مستطیل‌های بن‌بست را تعیین کرده و برای هر کدام یک نگهبان افزایشی در محدوده مورد نظر قرار می‌دهیم و در مرحله بعد، از بالاترین r -Star کار پیمایش را آغاز می‌کنیم و نسبت به نگهبانان درج شده بررسی می‌شوند و در صورت نیاز r -Starها و مستطیل نگهبانان بروز و یا نگهبانان جدید با روش تعیین محدوده نگهبان (مستطیل نگهبان) درخواست می‌شوند و در محدوده مورد نظر قرار می‌گیرند و با پردازش r -Star یک سطح پایین‌تر نسبت به مستطیل‌های نگهبان آغازین در ابتدا و سپس نسبت به مستطیل‌های نگهبان r -Star بررسی این‌که آیا دیده می‌شوند یا خیر صورت می‌گیرد. با توجه به توضیحاتی که در زیر بخش ۳-۴ به‌طور مفصل آمده است کار را ادامه داده و r -Star اول را تا جای ممکن بسط داده و این کار را برای بقیه r -Starها نیز تکرار خواهیم کرد تا حداقل تعداد نگهبانان یا بهتر بگوییم حداقل تعداد r -Starها را مشخص کنیم. در این مقاله در واقع به دنبال حداقل تعداد r -Star هستیم.

۴-۱. افراز چندضلعی متعامد اولیه مستطیل‌بندی شده به چندضلعی‌های متعامد r -Star

با استفاده از مختصات دو رأس یال‌های شمالی و مختصات یال‌ها و رئوس فرعی که در مرحله مستطیل‌بندی ذخیره شده است افراز r -Starها را از بالاترین یال شمالی آغاز می‌کنیم. یال‌های شمالی را نسبت به مؤلفه y از بالاترین به پایین‌ترین، اولویت‌بندی کرده و در یک لیست قرار می‌دهیم. به‌عنوان مثال بالاترین یال شمالی اولویت ۱ و یال شمالی بعدی اولویت ۲ و به‌همین ترتیب برای سایر یال‌های شمالی اولویت‌بندی را انجام می‌دهیم. سپس کار تشخیص و ذخیره مستطیل‌ها و افراز r -Starها را آغاز می‌کنیم. برای این کار دو رأس از یال شمالی مذکور را به‌عنوان دو رأس از مستطیل ابتدایی در نظر گرفته و رأس سمت چپ آن را با A و رأس سمت راست را با B نشان می‌دهیم. سپس طبق مرحله‌ای که در زیر آمده رأس سوم و چهارم مستطیل را می‌یابیم:

۱. رأسی از لیست رئوس جنوبی که از درجه ۲ باشد و مؤلفه x آن برابر با مؤلفه x رأس A و مؤلفه y آن کوچک‌تر از مؤلفه y رأس A و بیشینه باشد را به‌عنوان C (رأس سوم) تعیین می‌کنیم.
۲. رأس چهارم نیز رأسی با مؤلفه x رأس B و مؤلفه y رأس C است که با D نمایش می‌دهیم.
۳. $ABCD$ را به‌عنوان یک مستطیل از r -Star جاری ذخیره می‌کنیم.

بدین‌صورت اولین مستطیل به‌همراه چهار یالش ذخیره می‌شود. در مرحله بعد با توجه به این‌که همه مستطیل‌ها (به‌غیر از مستطیل پایانی که تمام ضلع جنوبی آن از نوع یال جنوبی چندضلعی P است) دارای یک یال افقی فرعی در ضلع جنوبی خود هستند، یال افقی فرعی مذکور از مستطیل اول را به‌عنوان یال شمالی مستطیل دوم در نظر گرفته و مراحل بالا را برایش تکرار می‌کنیم. این مراحل را برای هر کدام از یال‌های شمالی چندضلعی P تا رسیدن به مستطیلی که تمام ضلع جنوبی آن از نوع یال جنوبی چندضلعی P است ادامه می‌دهیم. در انتها اجتماع مستطیل‌های پیمایش شده برای هر کدام از یال‌های شمالی را به‌عنوان یک چندضلعی متعامد r -Star، همراه با یال شمالی، یال(های) شرقی، یال(های) غربی و یال(های) جنوبی با اولویت یال شمالی آن ذخیره می‌کنیم.

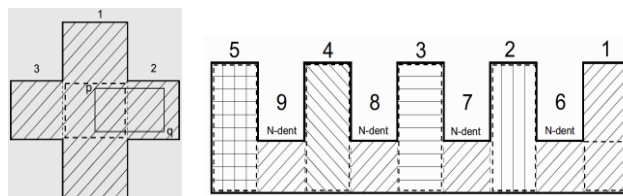
تبصره ۱: اگر در هنگام پیمایش مستطیل‌ها، یال جنوبی مستطیل جاری شامل بیش از یک یال فرعی افقی باشد (این امر زمانی رخ می‌دهد که بعد از مرحله مستطیل‌بندی، تمام حداقل یک یال فرورفتگی جنوبی (S -dent) از P ، قسمتی میانی از ضلع جنوبی مستطیلی را تشکیل دهد) مستطیل و یا مستطیل‌های پیمایش شده تا آن مستطیل را

به همراه مستطیل جاری به عنوان یک r-Star ذخیره کرده که آن را یک r-Star والد نامیده و آن یال‌های فرعی افقی را به همراه تعدادشان ذخیره کرده و با عنوان یال شمالی فرزند به ترتیب از راست به چپ اولویت می‌دهیم. به عنوان مثال اگر در سطح اولویت یال شمالی ۵ باشیم و اولین بار شرایط فوق نمایان شود، راست‌ترین یال فرعی افقی را به ابتدای لیست یال‌های شمالی فرزند و بعدی را به انتهای آن و به همین ترتیب اضافه می‌کنیم. پس از اتمام یال‌های شمالی اصلی و افزاز r-Star ها باید همین مراحل افزاز را برای یال‌های شمالی فرزند تکرار کنیم و r-Star هایی را که r-Star های فرزند می‌نامیم، ثبت کنیم.

نکته: به عنوان یک قرارداد فرضی، از میان چندین یال شمالی در یک سطح، راست‌ترین یال شمالی، بالاترین اولویت را بین آن‌ها به خود اختصاص می‌دهد.

از آن جاکه هر چندضلعی متعامد جدید افزاز شده یک r-Star است و فقط به یک نگهبان برای دیده شدن نیاز دارد ولی هم‌اکنون برای آن‌ها نگهبانی درج نمی‌شود. در مرحله افزاز چندضلعی اولیه به (r-Star) ممکن است مستطیل‌هایی با عنوان بن‌بست ذخیره شوند که در مرحله بعد اولین گزینه‌ها برای دریافت نگهبان هستند. بن بست را به طور مفصل در زیربخش ۲-۴ شرح می‌دهیم.

در شکل ۲ دو چندضلعی متفاوت مستطیل‌بندی شده را می‌بینید که نواحی قابل رویت برای هر نگهبان نیز مشخص شده است. در شکل ۲ آ، چندضلعی اولیه بر طبق الگوریتم پیشنهادی از ۹ ناحیه تشکیل شده است. نواحی ۶، ۷، ۸، و ۹ همگی به وسیله نگهبان ناحیه ۱ دیده می‌شوند. هم‌چنین هر کدام از ناحیه‌های ۲، ۳، ۴ و ۵ نیز به وسیله یک نگهبان قابل رویت است و به آن‌ها چهار نگهبان اختصاص می‌یابد. با این شرایط حداقل به ۵ نگهبان برای دیدن کل چندضلعی نیازمندیم. شکل ۲ ب به ۳ ناحیه تقسیم شده است که با توجه به شکل با قرار گرفتن یک نگهبان در r-Star اول، قادر هستیم تا دو ناحیه دیگر را نیز به طور کامل ببینیم.



(ب) فقط به یک نگهبان ابتدایی نیازمندیم

(آ) با توجه به جهت هاشور زدگی که در نواحی

مختلف تغییر جهت می‌دهد پنج نگهبان لازم است

شکل ۲. دو شکل ساده و شانه‌ای که مستطیل‌بندی شده اند که تعداد نگهبان‌های شان با مستطیل‌های خط چین

مشخص شده است

خط‌چین‌ها، یال‌های فرعی هستند که در مرحله مستطیل‌بندی ایجاد شده‌اند و هاشورها r-Star های افزاز شده هستند.

۲-۴. محدوده نگهبان برای چندضلعی‌های کوچک یا همان مستطیل نگهبان

چنان‌که گفته شد هر چندضلعی بدست آمده کوچک‌تر با روش ذکر شده یک r-Star است. برای به دست آوردن محدوده نگهبان چندضلعی‌های r-Star، از یال شمالی و یال(های) شرقی، یال(های) غربی و یال(های) جنوبی بهره می‌بریم، به این ترتیب که یک مستطیل را طوری درون این چندضلعی در نظر می‌گیریم که اگر نگهبان در هر قسمت از داخل آن قرار گیرد تمام چندضلعی اولیه P را می‌بیند. چندضلعی‌ها ممکن است با جذب قسمت‌هایی از چندضلعی‌های دیگر بزرگ‌تر شده و محدوده نگهبانشان نیز تغییر کند و در برخی مواقع قسمت‌های جذب شده به صورت مجزا نسبت

به چندضلعی مذکور قرار می‌گیرند، زیرا ممکن است پیوسته نباشند و فقط روی مستطیل نگهبان تأثیرگذار باشند. برای تعیین محدوده نگهبان این موارد را برای درخواست‌های نگهبان، از هر r -Star پیدا می‌کنیم:

۱. N یال

۲. بالاترین یال S

۳. چپ‌ترین یال E

۴. راست‌ترین یال W

به طوری که از امتداد دادن این یال‌ها و انتخاب محل تقاطع پاره خط‌ها به‌عنوان رئوس مستطیل، می‌توان چهار رأس مستطیل نگهبان را به‌دست آورد.

۳-۴. پیمایش r -Star ها و تعیین حداقل تعداد نگهبان

پس از مراحل مستطیل‌بندی و عملیات افراز، تمامی r -Star ها به‌همراه مختصات یال‌های مستطیل‌های شان ذخیره می‌شوند که ممکن است برخی از این r -Star ها به‌صورت مستطیلی ایجاد شوند و یا مستطیل‌هایی که به‌طور قطع به یک نگهبان نیازمندند، که شرایط ویژه‌ای را برای تعیین نگهبان به‌وجود می‌آورند. در ادامه به مفهوم بن‌بست می‌پردازیم و نحوه تعیین نگهبان را برای بن‌بست‌ها شرح می‌دهیم.

بن‌بست: به مستطیلی که یال شمالی و سه یال مجاور از نوع اصلی (همه یا قسمتی از یال) و یک یال به‌طور کامل از نوع فرعی داشته باشد بن‌بست گفته می‌شود.

محدوده نگهبان افزایشی: این نوع محدوده بیش‌ترین کاربرد را روی بن‌بست‌ها دارد اما روی r -Star هایی نیز اعمال می‌شود که حداقل یک سمت یال آن‌ها به‌طور کامل از نوع فرعی باشد و در خواست نگهبان داده باشند. برای این منظور ابتدا تغییراتی در بن‌بست یا r -Star مذکور به‌منظور گسترش آن انجام می‌دهیم و سپس مستطیل نگهبان را برای محدوده جدید فرضی تعیین می‌کنیم که با توجه به انواع بن‌بست‌های پیش‌بینی شده داریم:

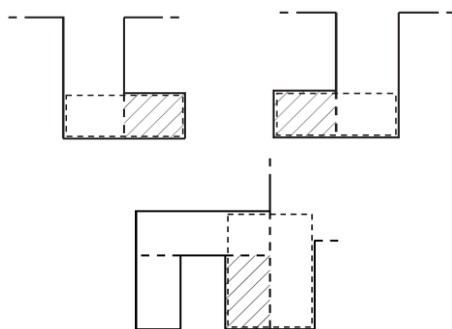
۱. برای بن‌بست‌هایی که دارای یال شرقی فرعی هستند و یا r -Star هایی که دارای یک یال شرقی و از نوع فرعی هستند، دو رأس یال فرعی را حذف کرده و از دو رأس آن یال فرعی، در امتداد یال‌های شمالی و جنوبی، به سمت خارج از آن (به‌سمت راست) پاره خط‌هایی رسم می‌کنیم تا چندضلعی متعامد P را در یالی (و نه رأسی مگر این‌که به انتهای P برسد) قطع کند. از بین نقاط تقاطع ایجاد شده، نقطه تقاطعی که دارای مؤلفه x کم‌تر است را به‌عنوان یک رأس جای‌گزین برای یال شرقی مستطیل یا r -Star مذکور تعیین کرده و رأس دیگر را به‌کمک مختصات رئوس دیگر به‌دست می‌آوریم.

۲. برای بن‌بست‌هایی که دارای یال غربی فرعی هستند و یا r -Star هایی که دارای یک یال غربی و از نوع فرعی هستند، دو رأس یال فرعی را حذف کرده و از دو رأس آن یال فرعی، در امتداد یال‌های شمالی و جنوبی، به سمت خارج از آن (به‌سمت چپ) پاره خط‌هایی رسم می‌کنیم تا چندضلعی متعامد P را در یالی (و نه رأسی مگر اینکه به انتهای P برسد) قطع کند. از بین نقاط تقاطع ایجاد شده، نقطه تقاطعی که دارای مؤلفه x بیش‌تر است را به‌عنوان یک رأس جای‌گزین برای یال غربی مستطیل یا r -Star مذکور تعیین می‌کنیم و رأس دیگر را به‌کمک مختصات رئوس دیگر می‌یابیم.

۳. برای r -Star ها و مستطیل‌هایی که دو یال از یال‌های آن‌ها به‌طور کامل فرعی باشد نیز باید محدوده نگهبان افزایشی تعیین شود که شامل شرقی-غربی، شمالی-شرقی و شمالی-غربی هستند. برای نوع اول نیاز به چهار

رأس جدید فرضی داریم که مانند مراحل ۱ و ۲ که در بالا آمده است عمل کرده و در نهایت تمام محدوده را به عنوان مستطیل نگهبان افزایشی تعیین می‌کنیم. اما برای حالت‌های دوم و سوم به سه رأس جدید فرضی نیازمندیم که مانند مراحل بالا برای هر نوع از یال‌های شرقی و غربی عمل کرده با این تفاوت که باید پس از آن، یال شمالی بروز شده را در امتداد شرقی‌ترین و غربی‌ترین یال به سمت بالا تا جایی که یکی از رئوس آن یک یال شمالی از P را قطع کند، گسترش داد.

در شکل ۳ می‌توان حالت‌هایی از موارد بالا را مشاهده کرد.



شکل ۳. حالت‌هایی از نگهبان افزایشی با یال‌های فرعی مختلف

با توجه به تعاریف بالا پس از تشخیص بن‌بست‌ها که با استفاده از اطلاعات ذخیره شده در مرحله افزایش r-Star ها میسر است، برای هر کدام از بن‌بست‌ها یک نگهبان درخواست شده که در صورت نداشتن اشتراک با محدوده یا محدوده‌های دیگر نگهبانان در محدوده نگهبان مستطیلی افزایشی خود درج می‌شود و به همراه مختصات رئوس ذخیره می‌شود و به دلیل آن که این نواحی بن‌بست به‌طور یقین به یک نگهبان در آن محدوده خاص برای دیده شدن نیاز دارند، آنان را نگهبانان اولیه یا نگهبانان قطعی نام‌گذاری می‌کنیم.

تبصره ۲: ممکن است دو محدوده نگهبان اولیه با هم دارای اشتراک باشند که در این صورت باید اشتراک آن‌ها را به‌عنوان مستطیل نگهبان اولیه در نظر گرفته و محدوده‌های نگهبان عامل اشتراک حذف گردند.

نکته: اگر یک محدوده نگهبان اولیه با دو یا چند محدوده نگهبان اولیه دیگر دارای اشتراک بود، دو محدوده‌ای که مؤلفه y یال شمالی‌شان بیش‌تر باشد را انتخاب می‌کنیم.

در ادامه به بررسی نواحی مختلف ایجاد شده در چندضلعی متعامد P پرداخته و پس از بررسی درخواست‌های آن‌ها نگهبان(های) جدید را در صورت لزوم درج می‌کنیم.

پیمایش را از بیش‌ترین اولویت که بالاترین r-Star است آغاز می‌کنیم، به این‌صورت که مستطیل‌های r-Star جاری را نسبت به اولین نگهبان اولیه به‌ترتیب اولویت (از بالا به پایین) برای دیده شدن بررسی می‌کنیم که مراحل بررسی و تعیین نگهبان در صورت لزوم بدین‌صورت است:

۱. ابتدا بررسی بر روی بالاترین و پایین‌ترین مستطیل از r-Star جاری صورت می‌گیرد که در صورت دیده شدن به معنای آن است که تمام r-Star قابل رویت است و نیازی به درخواست نگهبان جدید نیست و باید به‌سراغ r-Star بعدی برویم.

۲. اکنون با توجه به مرحله قبل که آیا بالاترین مستطیل دیده شده و یا پایین‌ترین مستطیل، یکی از مراحل زیر انجام می‌پذیرد:

• اگر پایین‌ترین مستطیل دیده شده باشد، بررسی از مستطیل ماقبل آخر به سمت بالا آغاز می‌شود تا جایی که دیگر نتواند مستطیلی را رویت کند و در نتیجه اجتماع مستطیل‌های دیده شده را به‌عنوان r -Star فرعی ذخیره کرده و r -Star جاری را با توجه به مستطیل‌های باقی‌مانده بروز می‌کنیم و ادامه بررسی با نگهبانان عادی به ترتیب اولویت صورت می‌گیرد.

• اگر بالاترین مستطیل دیده شده باشد بررسی از مستطیل دوم به سمت پایین آغاز می‌شود تا جایی که دیگر نتواند مستطیلی را رویت کند و در نتیجه اجتماع مستطیل‌های دیده نشده را به‌عنوان r -Star فرعی ذخیره کرده و نگهبان جاری را نسبت به مستطیل‌های دیده شده بروز می‌کنیم.

۳. اگر هیچ‌یک از مستطیل‌های بالا یا پایین از r -Star جاری به‌وسیله هیچ‌کدام از نگهبانان اولیه قابل مشاهده نباشد، r -Star جاری باید با نگهبانان عادی که در طول پیمایش چندضلعی P ایجاد می‌شوند، بررسی شود. اما اگر r -Star جاری اولین r -Star باشد ناگزیر نگهبان جدیدی درخواست شده و در محدوده مستطیلی خود قرار می‌گیرد و اولویتش بعد از نگهبانان اولیه و نسبت به یال شمالی r -Star مربوطه‌اش به‌عنوان نگهبان عادی تعیین می‌شود.

در حالت دوم، در صورتی که بخش پایین یک r -Star جدا شود باعث به‌وجود آمدن یک r -Star با یال شمالی فرعی شده که به آن‌ها r -Star های فرعی می‌گوییم و باید اولویت جدیدی برای آن در نظر گرفته شود ولی اجازه ندارد تا پایان کار درخواست نگهبان بدهد. در ادامه انواع r -Star هایی که باید تا پایان کار بررسی r -Star های چندضلعی متعامد P ، منتظر بمانند بدین‌قرار است:

۱. چندضلعی متعامد r -Star والد

۲. چندضلعی متعامد r -Star فرزند

۳. چندضلعی متعامد r -Star فرعی

هر کدام از سه بخش ذکر شده در بالا نباید تا اتمام کار بررسی دیگر r -Star ها بررسی شوند. پس از اتمام بررسی‌ها و درخواست‌ها و درج نگهبانان مورد نیاز، موارد مذکور درخواست‌های خود را بر طبق دستورالعمل بخش ۴-۳-۱ اعلام می‌کنند و در صورت دیده نشدن به‌وسیله نگهبانان موجود، نگهبان جدیدی درج می‌شود. هر چندضلعی باید به نگهبانان با توجه به اولویت‌شان درخواست دهد. اولویت نگهبانان درج شده بدین‌ترتیب آمده است:

۱. نگهبانان اولیه به ترتیب از بالا به پایین

۲. نگهبانان عادی به ترتیب از بالا به پایین

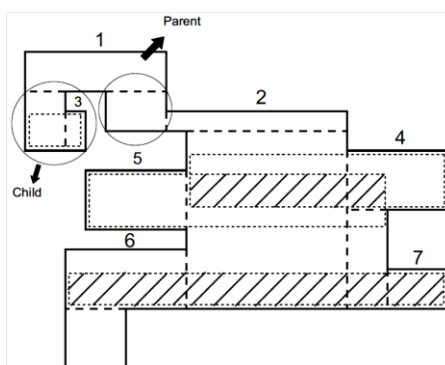
در مرحله بعد نوبت به r -Star بعدی در یک سطح پایین‌تر می‌رسد که با توجه به نگهبانان درج شده و اولویت‌شان درخواست‌های خود را ارسال می‌کند. اگر هیچ‌یک از نگهبانان اولیه قادر به دیدن بالاترین یا پایین‌ترین مستطیل آن نشدند، r -Star جاری درخواست خود را برای دیده شدن به نگهبانان عادی (در صورت وجود) می‌دهد که اگر باز هم دیده نشد باید برای خود نگهبانی بگیرد و آن را به لیست نگهبانان عادی اضافه کند.

تبصره ۳: اگر یک r -Star فقط یک مستطیل داشته باشد و بن‌بست نباشد اجازه درخواست نگهبان را در ابتدا ندارد و باید به انتهای لیست r -Star های اصلی انتقال داده شوند و منتظر پایان درخواست‌های این دسته بمانند.

در شکل ۴، r -Star های ۳، ۴، ۵ و ۷ بن‌بست بوده است و باید برای آن‌ها نگهبانانی به‌صورت افزایشی درج شود. چنان‌که در شکل می‌بینید ابتدا r -Star شماره ۳ درخواست نگهبان افزایشی هد که محدوده‌اش مشخص می‌شود. در

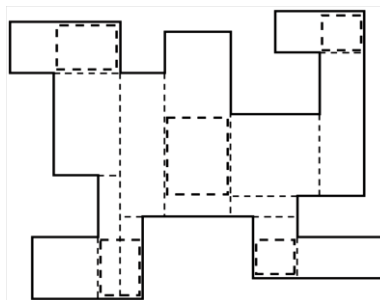
مرحله بعد r-Star شماره ۴ درخواست نگهبان افزایشی خود را صادر کرده که باید با نگهبان اولیه قبلی برای وجود اشتراک بررسی شود که در شکل ۴ پاسخ بررسی منفی است و یک نگهبان به آن اختصاص داده می شود. سپس r-Star شماره ۵ درخواست می دهد که نتیجه بررسی اشتراک با نگهبانان اولیه قبلی در این مرحله مثبت است و باید مستطیل اشتراکی بین آن دو را به عنوان نگهبان اولیه ذخیره کرده و عامل اشتراک حذف شود و برای هر دوی آنها فقط یک نگهبان درج می شود.

چنان که گفته شد، مستطیل(های) دیده شده از یک r-Star به وسیله یکی از نگهبانان ممکن است به r-Star صاحب آن نگهبان متصل نباشند ولی در محدوده مستطیل نگهبان تأثیرگذار است. مثلاً r-Star شماره ۶ در شکل ۴ پس از درخواست های متوالی به نگهبانان اولیه، نگهبان اولیه سوم که r-Star شماره ۶ است پاسخ مثبت داده و با برورسانی محدوده نگهبانش قادر است تمام r-Star شماره ۵ را ببیند. هر یک از نگهبانان اگر به طور کامل و یا مستطیل(های) بالای یک r-Star را ببینند اتمام پیمایش برای r-Star جاری است و به پیمایش r-Satr بعدی می پردازیم.



شکل ۴. یک چندضلعی متعامد ساده به همراه دو نگهبان اولیه

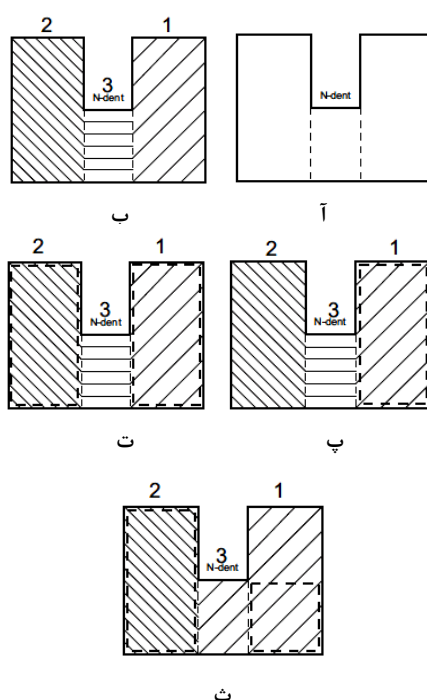
چنان که در شکل ۴ مشاهده می کنید r-Star شماره ۱ باعث به وجود آمدن والد و فرزند شده که r-Star والد فقط شامل یک مستطیل است که در شکل، با کلمه Parent مشخص شده و دارای دو فرزند است که در شکل با دایره و کلمه Child نشان داده شده اند. r-Star های فرعی نیز مانند r-Star های فرزند دارای یال شمالی فرعی هستند که در مرحله تخصیص نگهبان این دو دسته از r-Star ها با هم ادغام شده و نسبت به مؤلفه y یال شمالی از پایین به بالا مرتب می شوند و سپس درخواست نگهبان خود را طبق الگوریتم پیشنهادی اعلام می کنند. در شکل ۵ برای یک چندضلعی متعامد با هر چهار نوع فرورفتگی، پس از مستطیل-بندی و افزای، حداقل تعداد نگهبانان مورد نیاز که در شکل با مستطیل های خط چین مشخص شده اند، تعیین شده است.



شکل ۵. چندضلعی ساده متعامد با همه فرورفتگی ها که حداقل تعداد نگهبان را با مستطیل های خط چین، برای نواحی نیازمند نگهبان نشان می دهد

در شکل ۶ مراحل پیمایش بعد از مستطیل‌بندی در یک چندضلعی متعامد با فرورفتگی شمالی را مشاهده می‌کنیم که در نهایت به دو نگهبان نیاز دارد که با مستطیل‌های خط چین نشان داده شده است و هم‌چنین نواحی قابل رویت برای هر نگهبان با هاشورهای متفاوت مشخص شده است.

چنان‌که مشاهده شود در شکل ۶ ب، ابتدا سه r -Star افزا شده است که اولویت‌هایشان نیز به ترتیب از ۱ تا ۳ مشخص شده است و در قسمت پ، محدودهٔ اولین نگهبان نسبت به اولین r -Star تعیین شده است. در قسمت ت r -Star با اولویت پایین‌تر نسبت به مستطیل نگهبان اول بررسی شده که به دلیل دیده نشدن به وسیلهٔ آن، یک نگهبان درخواست کرده و هم‌چنین در قسمت ث هم به دلیل رویت کامل r -Star شمارهٔ ۳ به وسیلهٔ نگهبان اول نیازی به درج نگهبان نیست و فقط در مستطیل نگهبان مربوط به r -Star شمارهٔ ۱ به روز رسانی انجام شده و در نتیجه تمام چندضلعی برای دیده شدن حداقل به دو نگهبان نیاز دارد.



شکل ۶. مراحل پیمایش در یک نمونه شکل با فرورفتگی نوع N و تعیین حداقل تعداد نگهبان.

۴-۳-۱. نحوه درخواست‌های r -Star های والد و فرزند و فرعی برای نگهبان

مطابق آنچه بیان شد، بالاترین اولویت پس از r -Star های عادی برای درخواست نگهبان، r -Star های والد هستند و موضوع بسیار مهم برای تخصیص نگهبان به والدها و فرزندها، اولویت پایین به بالای آنهاست. r -Star های فرعی نیز شباهت بسیاری به r -Star های فرزند دارند و دلیل این شباهت یال شمالی فرعی آنهاست که هم می‌توانند از بالا و هم از پایین دیده شوند. در نتیجه در الگوریتم پیشنهادی این دو دسته از r -Star ها را با یکدیگر به صورت صعودی (نسبت به مؤلفه y یال شمالی) ادغام کرده و در یک دسته قرار می‌دهیم و می‌توان از r -Star های فرعی به عنوان "فرزند خوانده" نام برد.

پس در ابتدا r -Star های والد درخواست نگهبان خود را ابتدا برای نگهبانان اولیه و سپس برای نگهبانان عادی ارسال کرده و طبق روشی که توضیح داده شد بررسی می‌شوند و در صورت دیده نشدن توسط هیچ‌کدام از آنها،

نگهبان جدیدی دریافت می کنند. در مرحله بررسی والد ها برای تخصیص نگهبان، درخواستی به والد دیگری ارسال نمی شود که ممکن است محدوده های نگهبان مشترکی نیز به وجود آید، اما باید تا انتهای درخواست های *r-Star* های فرزند و فرعی صبر کنند و در انتهای الگوریتم اگر اشتراکی باقی مانده باشد به عنوان مستطیل نگهبان محاسبه شده و عوامل اشتراک حذف می شوند.

r-Star های والد اگر نگهبان جدید دریافت کنند آن را در لیست جدید به نام نگهبانان والد قرار می دهند ولی اگر *r-Star* ها فرزند و یا فرعی که در یک لیست "فرزند-فرعی" به صورت ادغام شده قرار دارند، نگهبان جدیدی بگیرند، آن را به انتهای لیست نگهبانان عادی اضافه می کنند. سپس دسته *r-Star* های ادغام شده فرزند-فرعی به ترتیب از پایین ترین به بالاترین درخواست های خود را ابتدا به نگهبانان اولیه، سپس به نگهبانان عادی و در آخر به نگهبانان والد اعلام می کنند و در صورت دیده نشدن توسط این سه دسته، نگهبان جدید دریافت کرده و در به انتهای لیست نگهبانان عادی اضافه می کنند. تفاوت عمده این دسته از *r-Star* ها با بقیه، این است که آن ها به *r-Star* های فرعی تقسیم نمی شوند و همه قسمت ها باید بررسی شوند و در صورت دیده شدن نگهبان را بروز می کنند. یعنی اگر از بالا دیده شوند و قسمت پایینی آن ها جدا شود، در همان مرحله آن را به وسیله نگهبانان دیگر نیز بررسی کرده و در صورت دیده نشدن به آن نگهبان اختصاص می دهند. در انتهای درخواست های والد ها و فرزندان و فرعی ها، هر نگهبان والد با نگهبانان والد قبلی از لحاظ اشتراک بررسی می شود و در صورت داشتن اشتراک، مستطیل اشتراکی به عنوان نگهبان نهایی ثبت شده و عوامل اشتراک حذف می شوند. در نتیجه تعداد نگهبانان والد در انتها به تعداد قبلی (نگهبانان اولیه و عادی) اضافه می شود.

۴-۴. بیان و توصیف شبه کد الگوریتم پیشنهادی Min_rStar_Cover

این الگوریتم، یک چندضلعی متعامد ساده را از ورودی می گیرد پس از مستطیل بندی چندضلعی اولیه و افراز آن به *r-Star* ها و تعیین نگهبان برای *r-Star* های درخواست دهنده، حداقل تعداد نگهبانان را به همراه مکان قرارگیری شان تعیین می کند.

هنگام گرفتن ورودی، مختصات رئوس و توالی یال ها به همراه نوع شان به صورت پادساعت گرد در یک ساختمان داده ذخیره می شود. ابتدا یال های فرعی عمودی و افقی و هم چنین رئوس فرعی ذخیره می شوند. بعد عملیات جداسازی *r-Star* ها به همراه ذخیره سازی مستطیل های هر *r-Star* به صورت جداگانه ادامه یافته، بن بست ها نیز تعیین می شوند. سپس در ابتدا به درخواست های بن بست ها و سپس به درخواست های *r-Star* ها برای نگهبان ها به ترتیب اولویت شان (از بالاترین به پایین ترین) پاسخ داده می شود. با توجه به مستطیل بندی مطرح شده در این مقاله، هر *r-Star* افراز شده عادی دارای یک یال شمالی خواهد بود.

الگوریتم Min_rStar_Cover : پوشش چندضلعی متعامد ساده با حداقل تعداد نگهبان

ورودی: یک چندضلعی متعامد ساده (P)

خروجی: حداقل تعداد نگهبانان و تعیین محدوده مستطیلی برای آن ها

بخش پیش پردازش الگوریتم:

ذخیره سازی مختصات رئوس و نوع یال های شان

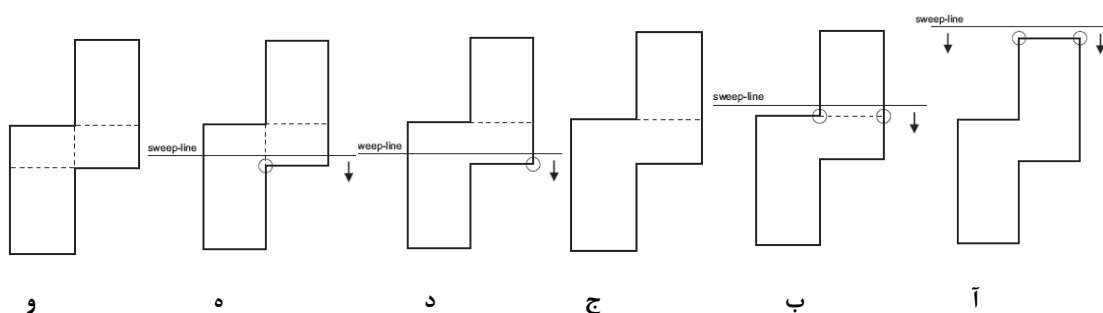
ذخیره سازی یال های شمالی به صورت نزولی (نسبت به مؤلفه y)

ذخیره سازی یال های فرعی عمودی برای همه یال های غربی و شرقی

ذخیره‌سازی یال‌های فرعی افقی برای همه یال‌های جنوبی
ذخیره‌سازی r-Star های اصلی، فرزند، والد، فرعی
ذخیره‌سازی مستطیل‌های r-Star های اصلی، فرزند، والد، فرعی
ذخیره‌سازی مستطیل‌های بن‌بست
آرایه‌هایی برای نگهداری نگهبانان اولیه، والد، عادی

مراحل الگوریتم در حالت کلی:

۱. مستطیل‌بندی چندضلعی اولیه P
 ۲. افزایش چندضلعی متعامد اولیه مستطیل‌بندی شده به چندضلعی‌های متعامد r-Star
 ۳. تعیین محدوده نگهبان برای چندضلعی‌های کوچک (مستطیل نگهبان)
 ۴. پیمایش r-Star ها و تعیین حداقل تعداد نگهبان
 ۵. تخصیص نگهبان برای r-Star های والد، فرزند و فرعی
- مستطیل‌بندی با روش آشنای خط جاروب است که باعث می‌شود در مرحله بعد بتوانیم چندضلعی اولیه را به تعدادی چندضلعی کوچک‌تر که هر کدام یک r-Star است افزایش کنیم. با شروع حرکت خط جاروب از بالا به پایین و در نظر گرفتن اولویت از راست به چپ محل تلاقی خط جاروب با رئوس P را به دست می‌آوریم:



شبه‌کد الگوریتم پیشنهادی:

Algorithm **Min_rStar_Cover** (P)

Input: A simple orthogonal polygon P

Output: A minimum number of r-visibility guards

1. **Sort** the N-edges of P by decreasing y-coordinate
2. **Put** the N-edges in an array
3. **For each** N-edge e **do**
4. **Sweep** from e to bottom
5. **Create** an orthogonal polygon
6. **while** (not reached to S-edge of last rectangle) **do**
- 7 **Create** orthogonal polygon
8. **Put** R as the new orthogonal polygon
9. **RG** ← **FIND_RECTANGLE_GUARD** (R)
10. **Initialize** an empty stack
11. **Push** R in the stack
12. **Check** whether R can see polygons inside the stack or not
13. **if** it can be seen **then**
14. **Remove** RG
15. **Update** the rectangle guard for R

در خط ۱ یال‌های شمالی چندضلعی P به صورت نزولی روی مؤلفه Y مرتب می‌شوند و در خط ۲ در یک آرایه قرار می‌گیرند. در خطوط ۳ تا ۱۵ با یک خط جاروب از بالا به پایین نگهبانان مورد نیاز را پیدا کنیم بدین ترتیب که یک نگهبان برای بالاترین چندضلعی افزاز شده پیدا کرده و به سراغ چندضلعی بعدی در یک سطح پایین‌تر می‌رویم (اگر بیش‌تر از یک چندضلعی در یک سطح قرار داشته باشند، راست‌ترین بین آن‌ها ابتدا انتخاب می‌شود) و محدوده‌اش را با نگهبان یا نگهبانان قبل بررسی می‌کنیم، خطوط ۶ تا ۷، که آیا تمام یا قسمت و یا قسمت‌هایی (منظور از قسمت، همان مستطیل‌های درون چندضلعی‌هاست) از آن به وسیله نگهبانان قبلی دیده می‌شود یا خیر. اگر تماماً دیده شد نگهبان جدید درخواست نمی‌شود و با چندضلعی افزاز شده بعدی ادامه پیدا می‌کند، و اگر قسمتی از آن دیده شد، از چندضلعی جاری جدا شده و به مالکیت نگهبان مذکور در می‌آیند و قسمت پایینی دیده نشده به صورت یک چندضلعی جدید افزاز شده در می‌آید و منتظر می‌ماند تا نوبتش فرارسد که دوباره مورد پردازش قرار بگیرد ولی اگر تماماً دیده نشود درخواست نگهبان جدید می‌شود. این مرحله تا زمانی تکرار می‌شود که تمام چندضلعی‌های افزاز شده بررسی شوند (خط ۸). در خط ۹ محل قرارگیری نگهبانان (تابع `FIND_RECTANGLE_GUARD`) با توجه به بالاترین یال جنوبی، راست‌ترین یال غربی، چپ‌ترین یال شرقی و تک‌یال شمالی به صورت یک مستطیل تعیین شده و مستطیل نگهبان نامیده می‌شود. در خطوط ۱۰ تا ۱۵ پیمایش یک چندضلعی افزاز شده برای دیده شدن به وسیله نگهبانان قبلی، نسبت به مستطیل نگهبان آن صورت می‌گیرد به این صورت که ابتدا پایین‌ترین یال جنوبی `r-Star` افزاز شده نسبت به مستطیل نگهبان مذکور بررسی شده که آیا دیده می‌شود یا خیر. اگر دیده شد به معنای آن است که تمام آن دیده می‌شود و گرنه از بالاترین مستطیل آن بررسی آغاز می‌شود تا به پایین، و هر تعداد مستطیل که دیده شد، به صورت فرضی از آن چندضلعی جدا شده و به چندضلعی مورد بحث می‌پیوندد و در نتیجه مستطیل نگهبان آن بروزرسانی می‌شود و این عمل تا انتها ادامه پیدا می‌کند. فرایند به روز رسانی که در خط ۱۵ مطرح شده بدین صورت است که این `r-Star` ها که یکی پس از دیگری فراخوانی می‌شوند شامل `r-Star` های اصلی، `r-Star` های والد و `r-Star` های فرزند و فرعی است. در ابتدا هر `r-Star` اصلی را بررسی می‌کند که آیا تک مستطیلی است یا خیر؟ اگر بود درخواستی برای نگهبان داده نمی‌شود و آن را به انتهای لیست `r-Star` های اصلی انتقال داده تا در انتها به درخواستش برای نگهبان پاسخ داده شود. مستطیل‌های `r-Star` ورودی با نگهبانان اولیه به ترتیب اولویت، بررسی می‌شوند که در صورت دیده شدن بالاترین و پایین‌ترین مستطیل فقط نیاز به بروزرسانی نگهبان اولیه جاری داریم و باید به بررسی `r-Star` بعدی بپردازیم ولی اگر فقط بالاترین مستطیل دیده شود، مستطیل‌های `r-Star` جاری از بالا به پایین بررسی می‌شوند و نگهبان، نسبت به مستطیل‌های دیده شده بروز شده و مابقی `r-Star` به جمع `r-Star` های فرعی می‌پیوندد و باز هم به سراغ `r-Star` بعدی می‌رویم. اگر پایین‌ترین مستطیل دیده شده باشد، مستطیل‌های `r-Star` مورد نظر از پایین به بالا تا جایی که دیده شوند را به صورت یک `r-Star` به لیست `r-Star` های فرعی اضافه کرده و `r-Star` جاری را با توجه به مستطیل‌های باقی‌مانده بروز می‌کند.

تابع `FIND_RECTANGLE_GUARD` برای تعیین محدوده مستطیل نگهبان یک `r-Star` است و چهار رأس برای محدوده مستطیل شکل نگهبان آن را تعیین می‌کند. اگر `r-Star` ورودی فقط یک یال شرقی داشته باشد و آن یال به طور کامل فرعی باشد سپس یال شرقی را به سمت راست تا یکی از رئوس آن یک یال شرقی از P را قطع کند گسترش می‌دهیم. اگر یال شمالی `r-Star` فرعی باشد سپس یال شمالی به روز شده را به سمت بالا تا یکی از رئوس آن یک یال شمالی از P را قطع کند گسترش می‌دهیم. و گرنه اگر `r-Star` ورودی فقط یک یال غربی داشته باشد و آن

یال، به‌طور کامل فرعی باشد سپس یال غربی را به‌سمت چپ تا یکی از رئوس آن یک یال غربی از P را قطع کند گسترش می‌دهیم. اگر یال شمالی r -Star فرعی باشد سپس یال شمالی به‌روز شده را به‌سمت بالا تا یکی از رئوس آن یک یال شمالی از P را قطع کند گسترش می‌دهیم. اکنون رئوس محدوده مستطیل شکل نگهبان را چنین می‌یابیم:

RW : راست‌ترین یال غربی r -Star ورودی،
 LE : چپ‌ترین یال شرقی r -Star ورودی،
 TS : بالاترین و چپ‌ترین یال جنوبی r -Star ورودی،
 N : یال شمالی r -Star ورودی،
 A : نقطه تقاطع امتداد یال LE و یال N ،
 Cx : برابر با مقدار x خط عمودی LE شود
 Cy : برابر با مقدار y خط افقی TS شود.
 Dx : برابر با مقدار x نقطه B شود.
 Dy : برابر با مقدار y نقطه D شود.

مستطیلی با رئوس A و B و C و D همان مستطیل نگهبان مد نظر است.

قضیه ۱: الگوریتم Min_rStar_Cover پوشش دید مستطیلی چندضلعی‌های متعامد ساده با حداقل r -Star ها را به درستی محاسبه می‌کند.

اثبات: در الگوریتم Min_rStar_Cover ابتدا عملیات مربوط به مستطیل‌بندی انجام می‌شود. ابتدا مختصات رئوس و نوع یال‌ها را توسط توالی آنها به‌صورت پادساعت‌گرد در یک ساختمان داده به‌نام P ذخیره می‌کنیم. سپس یال‌های شمالی در یک آرایه دوبعدی به‌نام N به‌صورت نزولی مرتب می‌شوند. پاره خط‌هایی عمودی از رأس پایینی یال‌های شرقی و غربی P در راستای آن‌ها امتداد داده تا P را در یک یال جنوبی قطع کنند. آن پاره خط را یال فرعی عمودی و آن نقطه تقاطع را رأس فرعی جنوبی می‌نامیم. در مرحله بعد به‌صورت مشابه، به‌وسیله یال‌های جنوبی که از درجه ۲ باقی‌مانده‌اند، یال‌های فرعی افقی پدید می‌آیند با این تفاوت که یال‌های جنوبی مذکور را در راستای افق تا یکی از یال‌های اصلی P و یا یال فرعی عمودی را قطع کند امتداد می‌دهیم. در ادامه برای یال‌های شمالی مرتب شده در آرایه N در دو حلقه تکرار (یک حلقه روی یال‌های شمالی و حلقه بعدی برای تعیین و ذخیره‌سازی مستطیل‌های r -Star) ابتدا چهار رأس مستطیل اول را پیدا کرده و به‌وسیله یال فرعی افقی که قسمتی از یال جنوبی آن را تشکیل می‌دهد به‌دنبال مستطیل بعدی رفته و این کار را تا زمانی ادامه می‌دهد که یکی از شرایط ذیل رخ دهد:

- به مستطیلی برسد که تمام یال جنوبی آن، همه یا قسمتی از یال جنوبی P باشد.
- به مستطیلی برسد که بیش از یک یال فرعی افقی در یال جنوبی خود داشته باشد.

اگر حالت اول رخ دهد اجتماع مستطیل‌های پیمایش شده را به‌عنوان یک r -Star عادی و اگر حالت دوم رخ دهد اجتماع آن‌ها را به‌عنوان r -Star والد ذخیره می‌کنیم و یال‌های فرعی افقی مذکور را به‌عنوان یال‌های شمالی فرزند در نظر گرفته تا مراحل فوق را پس از اتمام یال‌های شمالی برای آن‌ها تکرار کند و r -Star های فرزند پدید آیند.

در ادامه پس از مشخص کردن مستطیل نگهبان بن‌بست‌ها با تابع $FIND_RECTANGLE_GUARD$ بررسی می‌کنیم که آیا نگهبان اولیه جاری با نگهبانان اولیه قبلی اشتراک دارد یا خیر. اگر اشتراک داشته باشد، مستطیل نگهبان اشتراکی روی هر دو نگهبان عامل اشتراک بازنویسی می‌شود. سپس به بررسی r -Star های ورودی برای دیده شدن به‌وسیله نگهبانان اولیه می‌پردازیم. البته اگر r -Star ورودی (فرزند-فرعی) باشد و توسط نگهبانان عادی نیز دیده نشود برای بررسی به‌وسیله نگهبانان والد فرستاده می‌شود. در الگوریتم ابتدا بررسی می‌شود که آیا r -Star ورودی بن‌بست است یا خیر، که اگر بن‌بست باشد باید نگهبانش به‌صورت افزایشی در نظر گرفته شود. در مرحله اول از سمت یال فرعی عمودی آن، عملیات گسترش را انجام داده و در ادامه مستطیل نگهبان را چنان‌که در بخش ۲-۴ گفته

شد توسط یال شمالی، راست ترین یال غربی، چپ ترین یال شرقی و بالاترین یال جنوبی به دست می آوریم. در انتها اشتراک نگهبانان مستطیل های نگهبان r -Star های والد با یکدیگر بررسی شده و در صورت وجود اشتراک، عوامل اشتراک حذف و مستطیل نگهبان اشتراکی به عنوان نگهبان جدید ثبت می شود.

۴-۵. پیچیدگی زمانی الگوریتم

قضیه ۲: پیچیدگی زمانی بدترین حالت الگوریتم Min_rStar_Cover ، $O(n^5)$ است.

اثبات: با فرض این که n تعداد رئوس چندضلعی P باشد، ذخیره توالی یال های چندضلعی که تعدادشان n است در زمان $O(n)$ صورت می گیرد. در مرحله مستطیل بندی که در مجموع، امتداد یال های شرقی و غربی را در ابتدا و یال های جنوبی را پس از آن به دنبال دارد، برای یافتن نقطه تقاطع برای هر یک از یال ها به زمان اجرای $O(n)$ نیاز داریم که برای تعداد n یال برابر با $O(n^2)$ می شود. مرحله بعد، عملیات افزای r -Star ها را شامل می شود که برای هر یال شمالی اصلی و یا فرزند، یافتن مستطیل های متوالی در راستای عمود تا رسیدن به انتهای چندضلعی P و یا یال های شمالی فرزند نیاز به زمان $O(n)$ داریم. حال آن که چون تعداد یال ها را n فرض کرده ایم در نتیجه به زمان اجرای $O(n^2)$ نیازمندیم.

مرحله سوم الگوریتم، مرحله تخصیص نگهبان است که به دو قسمت مجزا تقسیم می شود. یک قسمت تخصیص نگهبانان اولیه است که با در نظر گرفتن تعداد n بن بست به دلیل این که هر نگهبان اولیه با نگهبانان اولیه قبل از خود از لحاظ داشتن اشتراک بررسی می شود، در زمان $O(n^2)$ انجام می شود و دیگری تخصیص نگهبان برای انواع گوناگون r -Star ها است. تعداد هر دسته از r -Star ها حداکثر n است. بررسی هر r -Star ابتدا به وسیله بالاترین و پایین ترین مستطیل آن صورت می گیرد. با توجه به این که بررسی دیده شدن هر مستطیل در زمان $O(1)$ انجام می شود، در بهترین حالت که تعداد n تا r -Star با n نگهبان اولیه به طور کامل دیده شوند، به زمان $O(n^2)$ خواهیم رسید. اما اگر در حالتی دیگر که می توان آن را حالت متوسط نامید به بررسی مستطیل های r -Star ها از بالا پردازیم، در این حالت با در نظر گرفتن تعداد n مستطیل به زمان اجرای $O(n^3)$ نیاز داریم. ولی اگر حالت سوم به عنوان بدترین حالت رخ دهد، مستطیل های r -Star از پایین قابل رویت باشند، که در آن حالت نیز اگر تعداد n مستطیل با n نگهبان عادی دیده شود زمان $O(n^2)$ صرف خواهد شد که در نتیجه در مجموع به زمان اجرای زیر دست می یابیم $O(n^2) * O(n^3) = O(n^5)$. از این رو، پیچیدگی در بهترین حالت $O(n^2)$ ، در حالت متوسط $O(n^3)$ و در بدترین حالت $O(n^5)$ اتفاق می افتد.

۴-۶. پیچیدگی فضایی الگوریتم پیشنهادی

قضیه ۳: پیچیدگی فضایی الگوریتم Min_rStar_Cover ، $O(n^2)$ است.

اثبات: در این بخش ساختار داده های استفاده شده در الگوریتم پیشنهادی را بررسی می کنیم. هر کدام از رئوس چندضلعی متعامد P را با دو مقدار به ازای مؤلفه های مختصات x و y مشخص می کنیم و هر کدام را می توانیم با یکی از انواع یال های شمال، جنوبی، شرقی و یا غربی نشان دهیم. پس برای هر رأس سه مقدار ذخیره می شود که عبارتند از مؤلفه x ، مؤلفه y و نوع یال. در واقع از بالاترین و راست ترین رأس به صورت پادساعت گرد توالی یال ها را دنبال کرده و یال ها را تعیین می کنیم. به این ترتیب که دومین رأس پیمایش شده را با برچسب یال شمالی در نظر می گیریم و به همین صورت ادامه می دهیم. یعنی هر دو مقدار متوالی در این آرایه مفهوم یک یال را دارد که نوع آن در رأس دوم

درج شده است. پس برای ذخیره یال‌های چندضلعی P به یک آرایهٔ دوبعدی با فضای $3n$ نیاز داریم. برای رئوس و یال‌های فرعی، بعد از مرحله مستطیل‌بندی نیز به فضایی بیش‌تر از فضای ذخیره‌سازی رئوس اصلی نیاز نداریم و به‌همین دلیل از محاسبه آن صرف‌نظر می‌کنیم.

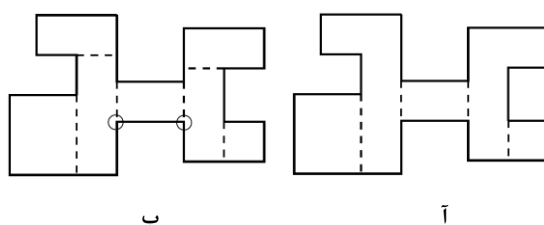
در مرحلهٔ بعد که مرحلهٔ افزایش‌پذیری r -Star هاست، برای هر r -Star مستطیل‌هایی را ذخیره کرده که برای هر مستطیل چهار رأس را نگهداری می‌کنیم و نیاز به ذخیرهٔ هشت مؤلفه (دو مؤلفه برای هر رأس) برای به‌آزای هر مستطیل داریم. از طرفی تعداد مستطیل‌های هر r -Star را به‌طور تقریبی حداکثر $n/2$ در نظر می‌گیریم و هر مستطیل را با هشت مؤلفه مختصات رئوس و چهار مشخصه برای بن‌بست، والد، فرزند و شماره اولویت ذخیره کنیم به‌مقدار حافظه $6n$ برای آن‌ها نیاز داریم. تعداد r -Star ها را نیز در بدترین حالت $n/2 - 1$ است که در نتیجه برای ذخیره r -Star ها به $n^2 - 2n$ حافظه برای ذخیره‌سازی r -Star ها نیاز داریم. اگر تعداد نگهبانان اولیه را حداکثر $n/2$ در نظر بگیریم و برای هر کدام هشت مؤلفه ذخیره کنیم به $4n$ فضا برای ذخیره آن‌ها نیاز داریم. تعداد نگهبانان عادی و والد را نیز می‌توان حداکثر $n/2$ در نظر گرفت که به‌صورت مشابه به حداکثر $4n$ فضای ذخیره‌سازی نیازمندیم. در ادامه r -Star های فرعی که به‌وجود می‌آیند در مراحل قبل در محاسبهٔ فضای ذخیره‌سازی r -Star ها شرکت کرده‌اند. پس از جمع‌بندی فضاهای مورد نیاز برای نگهداری داده‌های چندضلعی متعامد P خواهیم داشت: مجموع فضاهای ذخیره‌سازی الگوریتم پیشنهادی

$$n + n^2 - 2n + 4n + 4n = n^2 - 7n$$

در نتیجه پیچیدگی فضایی الگوریتم پیشنهادی را کران بالای مقدار بالا در نظر می‌گیریم که برابر $O(n^2)$ است.

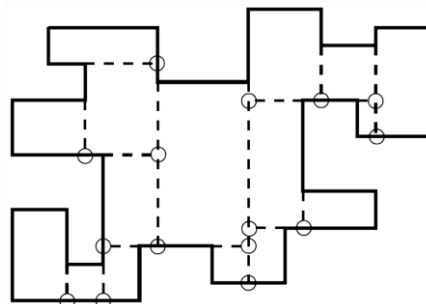
۴-۷. مستطیل‌بندی چندضلعی‌های ساده متعامد

در ابتدا لیستی از تمام یال‌های چندضلعی اولیه P به‌همراه رئوسشان ذخیره می‌شود. سپس در مرحله اول مستطیل‌بندی پاره‌خط‌هایی از رئوس پایینی یال‌های شرقی و غربی که با یال مجاورشان زاویهٔ 270° درجه دارند، در راستای عمود امتداد داده شده تا جایی که چندضلعی P را در یک نقطه قطع کند که تا این‌جا یال‌های عمودی فرعی به همراه رئوس فرعی خود ذخیره شده‌اند. در مرحلهٔ بعد، از آن رئوس یال‌های جنوبی که با یال مجاور زاویهٔ 270° درجه دارند و درجهٔ آن‌ها ۲ است پاره‌خط‌هایی به‌سمت داخل چندضلعی P در راستای افق امتداد می‌دهیم تا جایی که یکی از یال‌های فرعی و یا اصلی چندضلعی P را در یک نقطه قطع کند که در این مرحله یال‌های افقی فرعی نیز به‌همراه رئوس فرعی خود ذخیره می‌شوند (تمام رئوس فرعی ایجاد شده را به‌عنوان رئوس جنوبی می‌شناسیم) و کار مستطیل‌بندی چندضلعی P به اتمام می‌رسد. در شکل ۷ ترتیب مراحل مستطیل‌بندی یک چندضلعی متعامد ساده به‌ترتیب در قسمت‌های آ و ب مشاهده می‌شود.



شکل ۷. مراحل مستطیل‌بندی یک چندضلعی متعامد ساده که یال‌های فرعی در آن با خط‌چین نشان داده شده‌اند.

با توجه به الگوریتم پیشنهادی در این مقاله، مستطیل‌های چندضلعی مستطیل‌بندی شده، در مرحلهٔ افزایش r -Star ها و نه در هنگام مستطیل‌بندی شناسایی شده و ذخیره می‌شوند. در شکل ۸ چندضلعی متعامد ساده مستطیل‌بندی شده که یال‌های فرعی و رئوس فرعی ایجاد شده در آن مشخص شده‌اند را با تمام فرورفتگی‌ها مشاهده می‌شود.



شکل ۸. چندضلعی متعامد ساده مستطیل‌بندی شده که یال‌های فرعی در آن با خط چین و رئوس فرعی با دایره‌های کوچک نمایش داده شده است.

۵. مقایسه الگوریتم پیشنهادی با کارهای انجام شده قبلی

الگوریتم پیشنهادی در این مقاله با روشی نوین، چندضلعی اولیه را به گونه‌ای مستطیل‌بندی می‌کند که بتوان آن را به تعدادی چندضلعی متعامد r -Star افزایش کرد. این الگوریتم روی هر نوعی از چندضلعی‌های متعامد ساده (بدون حفره) کاربرد دارد و قادر است با وجود هر نوع فرورفتگی حداقل تعداد نگهبانان را در محل خود مستقر کند. الگوریتم اخیر [12] که در سال ۲۰۱۵ منتشر شده فقط روی چندضلعی‌های متعامد بدون فرورفتگی نوع N به کار گرفته می‌شود یعنی حتی با در نظر گرفتن زمان اجرای مطلوب، روی چندضلعی‌هایی که از همه جهات دارای فرورفتگی هستند پاسخ نمی‌دهد. پس مستلزم الگوریتم‌هایی کامل، هر چند با مرتبهٔ اجرایی بالاتر هستیم که الگوریتم پیشنهادی قبل برای مسئله پوشش کامل چندضلعی‌های ساده متعامد با میدان دید مستطیلی در سال ۲۰۰۷ [10] با مرتبهٔ اجرایی $O(n^{17} \text{ poly-logn})$ منتشر شده که در الگوریتم پیشنهادی ما در این مقاله در بدترین حالت، مرتبه اجرایی آن به $O(n^5)$ می‌رسد.

۶. نتیجه‌گیری

در این مقاله مسئلهٔ پوشش چندضلعی‌های متعامد ساده بدون محدودیت در نوع و تعداد فرورفتگی‌ها را بررسی کردیم. این الگوریتم قادر به مستطیل‌بندی و افزایش یا جداسازی چندضلعی‌های کوچک‌تر است که هر کدامشان یک چندضلعی r -Star هستند. سپس با پیمایش r -Star ها نسبت به نگهبانان ثبت شده (به ترتیب اولویت) محدوده نگهبانان (مستطیل‌های نگهبان) را به روز کرده و در نهایت به حداقل تعداد نگهبانان با میدان دید r دست یافتیم. در این مقاله، ایدهٔ مطرح شده بر چندضلعی‌های ساده متعامدی بود که دارای حفره نیستند. این مقاله راه را برای کار روی نوع دیگر چندضلعی‌ها که دارای حفره هستند [9] نیز هموار کرده است. حفره‌ها می‌توانند ستون‌ها در گالری محسوب شوند که آن‌ها را نیز می‌توان به صورت مستطیل یا چندضلعی متعامد ساده در نظر گرفت. مسئله این جاست که این مستطیل‌های داخل چندضلعی با مستطیل‌های درون چندضلعی که در این مقاله مطرح شده متفاوتند و مانع دید نگهبانان می‌شوند.

تشکر و قدردانی

از داوران محترم که با پیشنهادات سازنده خود موجب ارتقاء کیفیت مقاله شده‌اند تقدیر و سپاسگزاری می‌کنیم.

منابع

1. Aggarwal A., "The Art Gallery Theorem: its Variations, Applications, and Algorithmic Aspects", PhD. Thesis, Department of Electrical Engineering and Computer Science, John Hopkins University (1984).
2. Culberson J., Reckhow R. A., "Orthogonally convex coverings of orthogonal polygons without holes", *J. Comput. Systems Science* 39 (2) (1989) 166-204.
3. Gewali L., Keil M., Ntafos S. C., "On covering orthogonal polygons with star-shaped polygons", *Information Sciences* 65 (1992) 45-63.
4. Keil J. M., "Minimally covering a horizontally convex orthogonal polygon", *Proc. 2nd Annual ACM Symp. Computational Geometry*, (1986) 43-51.
5. Lingas A., Wasylewicz A., ylin' ski P. Z, "Note on covering orthogonal polygons with star-shaped polygons", *Information Processing Letters* 104 (6) (2007) 220-227.
6. Motwani R., Raghunathan A., Saran H., "Covering orthogonal polygons with star polygons: the perfect graph approach", *J. Comput. Systems Science* 40, (1990) 19-48.
7. O'Rourke J., "Art Gallery Theorems and Algorithms", Oxford University Press (1987).
8. Urrutia J., "Art gallery and illumination problems", *Handbook of Computational Geometry*, Elsevier Science, Amsterdam, (2000) 973-1027.
9. Pawel Zylinski, "Institute of Mathematics", University of Gda_nsk, 80952 Gda_nsk, Poland, (2006).
10. Worman C., Keil J. M., "Polygon decomposition and the orthogonal art gallery problem", *International Journal of Computational Geometry & Applications* 17(2) (2007) 105-138.
11. Leonidas Palios, Petros Tzimas, "Covering Class-3 Orthogonal Polygons with the Minimum Number of r-Stars, European Social Fund-ESF", National Strategic Reference Framework (NSRF), MIS 375891 (2012).
12. Leonidas Palios, Petros Tzimas, "Minimum r-Star Cover of Class-3 Orthogonal Polygons, Springer International Publishing Switzerland", J. Kratochvíl et al. (Eds.): IWOCA 2014, LNCS 8986 (2015) 286-297, DOI: 10.1007/978-3-319-19315-125.