دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

رساله دکتری

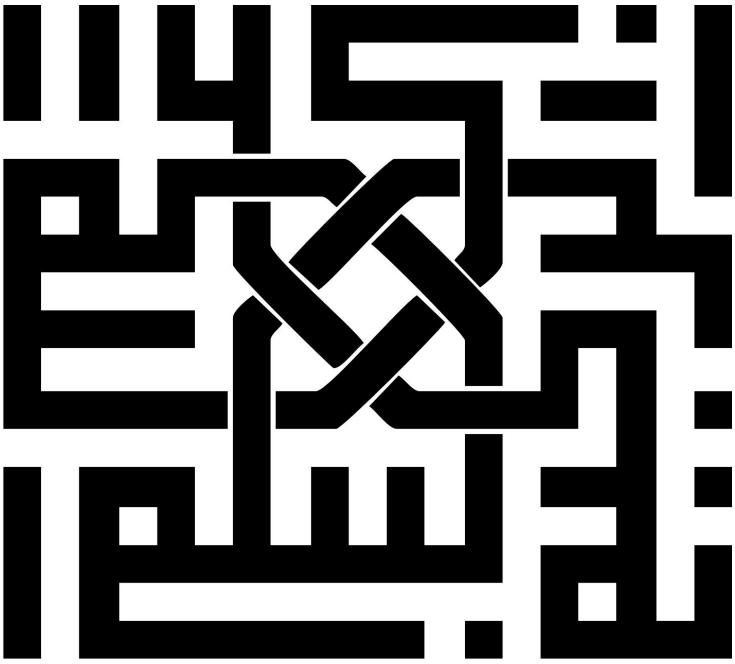
عنوان

مسیریابی ربات با حسگرهای محدود در محیط ناشناخته

نگارش
آزاده طباطبائی

استاد راهنما
دکتر محمد قدسی

مهر ۱۳۹۵



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

رساله دکتری

مسیریابی ربات با حسگرهای محدود در محیط ناشناخته

نگارش: آزاده طباطبائی

استاد راهنما: دکتر محمد قدسی امضاء:

استاد ممتحن داخلی اول: دکتر محمد علی آبام امضاء:

استاد ممتحن داخلی دوم: دکتر امیرحسین جهانگیر امضاء:

استاد ممتحن خارجی اول: دکتر علیرضا باقری امضاء:

استاد ممتحن خارجی دوم: دکتر علیرضا زارعی امضاء:

استاد ممتحن خارجی سوم: دکتر علی محدث امضاء:

تقدیم

به همسرم و فرزندان عزیزم ماندانا و پارسا

تقدیر و تشکر

وظیفه‌ی خود می‌دانم از زحمات استاد ارجمندم، جناب آقای دکتر محمد قدسی، صمیمانه تشکر و قدردانی نمایم. در تمامی مراحل انجام این رساله از راهنمایی‌های ایشان بهره بردم و شاگرد ایشان بودن از بزرگترین افتخارات زندگی‌ام است. از اساتید گرامی آقای دکتر محمد علی آبام، آقای دکتر امیرحسین جهانگیر و آقای دکتر منصور جم‌زاد که همواره در این راه به من کمک کردند و با پیشنهادهای خود باعث بهبود این اثر شدند، کمال امتنان را دارم. از اعضای آزمایشگاه الگوریتم دانشکده، خصوصا خانم سپیده آقامولائی به دلیل کمک‌های صمیمانه‌شان سپاسگزارم. همچنین از تمامی دوستان و خانواده‌ی عزیزم که در طول انجام رساله همراه من بودند، تشکر می‌کنم. از همسر و فرزندان عزیزم نیز که صبورانه در به سرانجام رساندن این راه سخت من را همراهی کردند قدردانی می‌کنم.

چکیده

در این پژوهش هدف طراحی و حل مسئله‌های هندسی، به ویژه مسئله‌های مسیریابی در محیط‌های ناشناخته است. بر خلاف اکثر مسئله‌های هندسی که یک نقشه از محدوده کاری در دسترس است، در این‌جا نقشه‌ای از محیط کاری در دست نیست. یک (یا چند) ربات در محیط شروع به حرکت می‌کند، این ربات در حالی که محیط را شناسایی می‌کند به گونه‌ای هدایت می‌شود که به هدف مسئله برسد طوری که از مسیر بهینه‌ی رسیدن به هدف، زیاد فاصله نگیرد. میزان اطلاعاتی که ربات از محیط به دست می‌آورد متناسب با قابلیت حسگرهایی است که به آن‌ها تجهیز شده است. فرض بر این است که ربات خواص متریک محیط مانند فاصله‌ها و زوایا را تشخیص نمی‌دهد بلکه ویژگی‌های ترکیبیاتی محیط مانند شکاف‌ها (ناپیوستگی عمقی)، ترتیب راس‌های محدوده کاری و غیره را شناسایی می‌کند. این ربات‌ها به نام ربات‌های ساده با حسگرهای محدود مطرح شده‌اند.

در این رساله الگوریتم‌های برخطی برای هدایت یک ربات ساده مجهز به حسگر شکاف که در محدوده‌های ناشناخته از یک نقطه شروع به حرکت می‌کند و با توجه به اطلاعات محدودی از محیط که از طریق حسگرهایش به‌دست می‌آورد، به نقطه هدف می‌رسد، ارایه می‌دهیم. نشان می‌دهیم مسیری که ربات طی می‌کند، از نظر طول و تعداد چرخش‌ها قابل رقابت با مسیر بهینه است. سپس ربات مجهز به حسگر شکاف را با اضافه کردن یک قطب‌نما تقویت می‌کنیم. نشان می‌دهیم ربات تقویت شده قادر است محدوده‌های وسیع‌تری را در امتداد یک مسیر مناسب پیمایش کند.

کلمات کلیدی: هندسه محاسباتی، الگوریتم‌های برخط، ضریب رقابتی، برنامه ریزی مسیر حرکت ربات، محیط ناشناخته، حسگرهای محدود

فهرست شکل‌ها

۱.۲ ارتباط بین فضای فیزیکی و ماشین(ربات) از طریق حسگرها و تاثیر ربات بر فضای فیزیکی. . ۹
۲.۲ (آ) یک چندضلعی ساده که زنجیره سازنده مرز آن، با رنگ تیره نشان داده شده است. ناحیه رنگی چندضلعی رویت پذیر نقطه q است. (ب) یک چندضلعی حفره‌دار. ۱۲

۱.۳ نمایشی از حسگرها در خودروی هوشمند [۷۸]. ۱۹
۲.۳ خودروی بدون سرنشین گوگل. ۲۱
۳.۳ حسگرها و برد و دقت آن‌ها. ۲۲
۴.۳ دوربین مناسب برای خودروی هوشمند با برد ۶۰ متر و دقت ۷۴۰ در ۴۸۰ نقطه [۷۳]. . . . ۲۴
۵.۳ مسیری که از روی پرتوها عبور کرده است [۶۱]. ۲۶
۶.۳ نواحی تیره رنگ سایه‌های تشخیص داده شده توسط حسگر هستند [۳۹]. ۲۷
۷.۳ (آ) ناحیه رویت‌پذیری نقطه q و شکاف‌هایی که حسگر شکاف گزارش می‌دهد. (ب) موقعیت شکاف‌ها. ۲۸
۸.۳ رخدادهای بحرانی و روش ساخت درخت پیمایش شکاف. (آ) شکاف‌های اولیه. (ب) رخداد تقسیم. (ج) رخداد ناپدید شدن. (د) یک رخداد تقسیم دیگر. ۳۰
۹.۳ نمایش بردار ترکیبیاتی $cvv(p) = (۱, ۰, ۱, ۰, ۱, ۰, ۱, ۱)$ بین دنباله راس‌های رویت‌پذیر از نقطه p. ۳۱
۱۰.۳ آنچه ربات از محیط درک می‌کند در سمت چپ نشان داده شده است. در شکل سمت راست یک چندضلعی حفره‌دار و گراف رویت‌پذیری آن که ربات تولید می‌کند، نشان داده شده است [۱۴]. ۳۲
۱۱.۳ (آ) قابلیت حرکتی ربات بیت‌باتس. (ب) ربات قادر است تشخیص دهد که روی راس انعکاسی یا محدب است و همچنین تشخیص می‌دهد که روی ضلع یا داخل چندضلعی قرار دارد [۳۲]. ۳۳

۱.۴ یک خیابان که روی نقاط s و t ساخته شده است. ۳۸
۲.۴ g_rسمت چپ‌ترین (پیشرفته‌ترین) شکاف راست و g_l سمت راست‌ترین (پیشرفته‌ترین) شکاف چپ است. (آ) ربات فقط یک راه دارد و بدون ابهامی آن را طی می‌کند. (ب) ربات دو راه دارد که یکی از آن‌ها را انتخاب می‌کند پس احتمالا از کوتاه‌ترین مسیر منحرف می‌شود. . . ۴۰
۳.۴ (آ) در نقاط ۱ تا ۶ رخدادهایی اتفاق می‌افتد که باعث تغییر پیشرفته‌ترین شکاف‌ها می‌شود. (ب) خطوط نقطه‌چین زنجیر محدب راست و چپ را نشان می‌دهد. نمایش چگونگی ساخته شدن و به‌روز شدن S-GNT وقتی ربات روی زنجیر راست حرکت می‌کند. ۴۱
۴.۴ (آ) یک خیابان متعامد. مسیر نقطه‌چین، مسیر تولید شده توسط استراتژی ما است. (ب) نمایش ساختار داده در نقطه شروع. ۴۶
۵.۴ مکان‌های ممکن برای شکاف‌ها در نقطه شروع حرکت. فلش‌ها جهتی را که ربات می‌تواند برای حرکت انتخاب کند، نشان می‌دهند. ۴۷
۶.۴ ربات سمت راست در امتداد مسیر نقطه‌چین حرکت می‌کند و ربات سمت چپ در امتداد مسیر ممتد حرکت می‌کند. همکاری دو ربات مانع از حرکت ربات چپ در مسیری اشتباه می‌شود. . ۵۱

۱.۵ خیابان و آنچه ربات از این محیط درک می‌کند. (آ) ربات فقط یک راه دارد و بدون ابهامی آن را طی می‌کند. (ب) ربات دو راه دارد که شکاف پیشرفته چپ را انتخاب می‌کند. . . . ۵۶
۲.۵ نمایش تغییرات ساختار داده‌ای که مکان g_l و g_r را نگه می‌دارد. در نقطه u زنجیر ساعتگرد از s به u به طور کامل از روی مسیر نقطه چین که s را به u وصل می‌کند، قابل دید است و شکاف پیشرفته چپ تا قبل از رسیدن به نقطه u در نقطه ۲ غیر فعال شده است. ربات در این نقطه به سمت شکاف پیشرفته راست می‌چرخد. ۵۸
۳.۵ مسیر نقطه‌چین از s به t مسیر حرکت ربات را نشان می‌دهد، در هر یک از نقاط تیره رنگ شکاف‌های پیشرفته تغییر می‌کنند که این تغییر باعث تغییر در وضعیت فعال بودن آن‌ها نیز می‌شود. ۵۹
۴.۵ در نقطه ۱ مکان شکاف پیشرفته چپ عوض می‌شود و غیر فعال می‌شود. در نقطه ۲ شکاف پیشرفته راست (تنها شکاف پیشرفته موجود) ناپدید می‌شود. مسیر نقطه چین $SP(s,t)$ است و u_iu_j یک گوش روی این مسیر است. ۶۱
۵.۵ دو شکل ساده از خیابان. (آ) $SP(s,t)$ فقط چپ‌گردی دارد. (ب) $SP(s,t)$ فقط راست‌گردی دارد. ۶۲
۶.۵ حالت کلی که کوتاه‌ترین مسیر از s به t هم راست‌گردی و هم چپ‌گردی دارد. مسیر پر رنگ مسیر حرکت ربات را نشان می‌دهد. ۶۴
۷.۵ یک خیابان متعامد. (آ) مسیر نقطه‌چین، مسیری که ربات برای رسیدن به هدف طی می‌کند. (ب) نمایش مکان شکاف‌هایی که ربات در نقطه شروع درک می‌کند و جهت‌هایی که ربات قادر به حرکت در راستای آن‌ها است. ۶۵
۸.۵ در نقطه شروع حرکت یک شکاف متعامد وجود دارد. ربات راستای این شکاف را برای حرکت انتخاب می‌کند. ۶۶
۹.۵ (آ) یک راستای متعامد در بین دو شکاف قرار دارد و ربات این راستا را برای حرکت از نقطه شروع انتخاب می‌کند. (ب) ربات یکی از راستاهایی که بلافاصله قبل یا بعد از شکاف‌های پیشرفته قرار دارد را برای حرکت از نقطه شروع انتخاب می‌کند. ۶۷
۱۰.۵ در نقطه شروع فقط یکی از شکاف‌های پیشرفته چپ یا راست وجود دارد. فلش پر رنگ جهت حرکت ربات را نشان می‌دهد. ۶۷

۱.۶ (آ)یک چندضلعی بسیار ساده ستاره‌ای که فقط ربات در نقطه شروع، فقط یک شکاف با برچسب L در آن حس می‌کند. ناحیه رنگی هسته آن است. ربات مجهز به حسگر ساده شکاف در نقطه شروع، با حرکت به سمت تنها شکاف موجود (مسیر نقطه‌چین) به اندازه زیادی از کوتاه‌ترین مسیر به سمت هسته دور می‌شود. (ب) آنچه ربات مجهز به حسگر شکاف و قطب‌نما در نقطه شروع از محیط درک می‌کند. ۷۲
۲.۶ ناحیه رنگی هسته چندضلعی ستاره‌ای است. مسیر تیره رنگ مسیر حرکت ربات از نقطه شروع s تا نقطه‌ای از هسته است. ۷۳
۳.۶ (آ)ربات ساده یکی از جهت‌های افقی عمودی که بعد از شکاف قرار دارد را برای حرکت انتخاب می‌کند. (ب) ربات با دید کامل، اولین جهتی که بعد از شکاف قرار دارد و زاویه آن تا شکاف بیش از ۴۵ درجه است را انتخاب می‌کند. ۷۵
۴.۶ (آ)ربات ساده در جهت افقی عمودی حرکت می‌کند در حالی که که فلش جهت حرکت ربات با دید کامل را نشان می‌دهد. (ب) ربات ساده در جهت یک گام به سمت شکاف راست و یک گام به سمت شکاف چپ گام بر می‌دارد در حالی که فلش جهت حرکت ربات با دید کامل را نشان می‌دهد. ۷۶
۵.۶ (آ)ربات ساده در جهت افقی عمودی حرکت می‌کند در حالی که مسیر پیوسته از p به k مسیر حرکت ربات با دید کامل را نشان می‌دهد. (ب) مسیر افقی عمودی ربات برای ساده‌تر شدن مقایسه به صورت مورب نشان داده شده است. ۷۸

۱.۷ نمایش مسیر حرکت ربات در دو حالت زاویه باز کوچک و زاویه باز بزرگ. ۸۱

۱.۸ خیابان تعمیم یافته. (آ) برخی از وترها در خیابان تعمیم یافته. (ب) آنچه ربات در نقطه شروع از محیط درک می‌کند. ۸۷
۲.۸ یک خیابان تعمیم یافته متعامد و مکان شکاف‌ها در دو نقطه s و p. ۸۸
۳.۸ مسیر پیموده شده ربات بین دو وتر شکاف متوالی. ۹۰

فهرست مطالب

۱ مقدمه ۱
۱.۱ انگیزه پژوهش ۱
۲.۱ نتیجه‌های به دست آمده ۲
۱.۲.۱ راهبرد رقابتی برای هدایت ربات در یک خیابان ناشناخته ۳
۲.۲.۱ هدایت ربات ساده در یک خیابان ناشناخته در امتداد مسیری با کمترین تعداد چرخش ۴
۳.۲.۱ پیمایش یک چندضلعی ستاره‌ای توسط یک ربات ساده ۴
۴.۲.۱ راهبرد رقابتی بهینه برای هدایت ربات با استفاده از اطلاعات محلی ۵
۵.۲.۱ راهبرد رقابتی برای هدایت ربات در یک خیابان عمومی ۵
۳.۱ مقاله‌های حاصل شده از این پژوهش ۵
۴.۱ طرح رساله ۶

۲ مفاهیم و تعاریف ۸
۱.۲ مسیریابی در محیط‌های ناشناخته ۸
۲.۲ ربات مجهز به حسگر محدود (ربات ساده) ۱۰
۳.۲ محدوده کاری ۱۱
۱.۳.۲ رویت پذیری ۱۱
۲.۳.۲ چندضلعی خیابان ۱۲
۳.۳.۲ وتر ۱۲
۴.۳.۲ خیابان تعمیم یافته ۱۳
۵.۳.۲ چندضلعی ستاره‌ای ۱۳
۶.۳.۲ چندضلعی رویت‌پذیر ۱۳
۴.۲ حسگر شکاف ۱۳
۱.۴.۲ شکاف‌های پیشرفته و ساختار داده S-GNT ۱۴
۲.۴.۲ وضعیت قیف ۱۴
۳.۴.۲ شکاف فعال ۱۴
۵.۲ قوی کردن ربات ۱۵
۶.۲ تحلیل رقابتی ۱۵
۷.۲ جمع‌بندی ۱۶

۳ حسگرها ۱۷
۱.۳ وضعیت تکنولوژی کنونی و روندهای آن در ماشین‌های هوشمند و خودمختار ۱۸
۱.۱.۳ حسگرهای متداول مورد استفاده در ماشین‌های هوشمند ۲۲
۲.۳ چرا از حسگرهای محدود استفاده می‌کنیم؟ ۲۴

۳.۳ تاریخچه‌ای ازحسگرهای محدود هندسی ۲۵
۴.۳ حسگرهای محدود استفاده شده در این پژوهش ۳۴
۵.۳ اهمیت استفاده از حسگرهای محدود در این پژوهش ۳۵
۶.۳ جمع‌بندی ۳۶

۴ راهبرد رقابتی برای هدایت ربات در یک خیابان ناشناخته ۳۷
۱.۴ نتایج اولیه ۳۸
۲.۴ راهبرد اصلی ۳۹
۱.۲.۴ ساختار داده ۴۱
۲.۲.۴ الگوریتم ۴۴
۳.۲.۴ تحلیل الگوریتم ۴۴
۳.۴ مسیریابی در یک خیابان متعامد در امتداد یک مسیر متعامد ۴۵
۴.۴ راهبرد تصادفی برای هدایت ربات ۴۸
۱.۴.۴ تحلیل الگوریتم ۴۹
۵.۴ همکاری ربات‌ها ۵۰
۱.۵.۴ الگوریتم ۵۰
۲.۵.۴ تحلیل الگوریتم ۵۱
۶.۴ جمع‌بندی ۵۴

۵ هدایت یک ربات ساده در خیابان در امتداد مسیری با کمترین تعداد چرخش ۵۵
۱.۵ مفاهیم مورد نیاز ۵۵
۲.۵ الگوریتم ۵۹
۱.۲.۵ تحلیل الگوریتم ۶۱
۳.۵ هدایت ربات در یک خیابان متعامد در امتداد یک مسیر متعامد با کمترین تعداد چرخش ۶۵
۱.۳.۵ جمع‌بندی ۶۹

۶ پیمایش یک چندضلعی ستاره‌ای توسط یک ربات ساده ۷۰
۱.۶ ضرورت وجود قطب‌نما ۷۱
۲.۶ مدل حسگر و محدودیت‌های حرکتی ربات ۷۱
۳.۶ الگوریتم ۷۲
۴.۶ تحلیل درستی و محاسبه ضریب رقابتی الگوریتم ۷۵
۵.۶ جمع بندی ۷۸

۷ راهبرد رقابتی بهینه برای هدایت ربات با استفاده از اطلاعات محلی ۷۹
۱.۷ مفاهیم اولیه مورد نیاز ۸۰
۲.۷ اطلاعات اساسی ۸۰
۳.۷ الگوریتم ۸۱
۱.۳.۷ الگوریتم قطعی ۸۱
۲.۳.۷ الگوریتم تصادفی ۸۲
۳.۳.۷ تحلیل الگوریتم ۸۲
۴.۷ جمع‌بندی ۸۴

۸ راهبرد رقابتی برای هدایت ربات در یک خیابان تعمیم یافته ۸۵
۱.۸ مفاهیم مورد نیاز ۸۶
۱.۱.۸ محدوده کاری ۸۶
۲.۱.۸ مدل حسگر و محدودیت حرکتی ربات ۸۶
۲.۸ راهبرد اصلی برای هدایت ربات در یک خیابان تعمیم یافته متعامد ۸۷
۱.۲.۸ الگوریتم ۸۹
۲.۲.۸ تحلیل الگوریتم ۹۱
۳.۸ جمع‌بندی ۹۲

۹ نتیجه‌گیری و پیشنهادهایی برای ادامه کار ۹۳
۱.۹ خلاصه‌ای از نتایج بدست آمده ۹۳
۱.۱.۹ مزایا و معایب الگوریتم‌های ارائه شده ۹۵
۲.۹ پیشنهادها ۹۵
۱.۲.۹ توسعه محدوده کاری ۹۵
۲.۲.۹ تغییر مدل حسگر ۹۶
۳.۲.۹ محدودیت‌های حرکتی برای ربات و تغییر شکل ربات ۹۶
۳.۹ مسئله‌های باز مشابه ۹۶

فصل ۱

مقدمه

مسئله طرح مسیر یکی از مسئله‌های مطرح در زمینه هندسه محاسباتی [۶] و رباتیک [۴۵، ۵۱] و الگوریتم‌های برخط [۵] است که در دهه‌های اخیر بسیار مورد توجه قرار گرفته است. دو رویکرد اساسی در این زمینه وجود دارد:

رویکرد اول: حالتی است که محیط کاری از قبل شناخته شده است و یک نقشه از این محدوده کاری در اختیار ربات (عامل مسیریاب) قرار می‌گیرد و ربات سعی می‌کند با استفاده از این نقشه فعالیت مناسبی را برای رسیدن به هدفش انجام دهد. چنین محیطی فضای پیکربندی[۱] نامیده می‌شود و ربات در این فضا به مسیریابی می‌پردازد [۱۵، ۱۹، ۲۰].

رویکرد دوم: حالتی است که ربات نقشه‌ای از محدوده کاری در اختیار ندارد و همچنین مشخصات (مختصات) مکانی را که در آن است ، ندارد. در این حالت اطلاعات مورد نیاز از طریق حسگرها در حین اجرای فرایند مسیریابی به‌دست می‌آید. این وضعیت در بسیاری از کاربردهایی که در تعامل با محیط فیزیکی هستند پیش می‌آید [۲، ۱۲، ۱۳، ۲۷، ۲۸]. (این رویکرد محور اصلی این پژوهش است.)

۱.۱ انگیزه پژوهش

اکثر مطالعات قبلی در زمینه مسیریابی در محیط ناشناخته براساس محاسبات دقیق و زیاد برای پیدا کردن مسیرهای بهینه است. به عبارت دیگر از رباتی برای مسیریابی استفاده شده است که اطلاعات زیادی از محیط (نقشه محیط) را درک می‌کند. امروزه سادگی عامل مسیریاب (ربات) و هزینه ساخت سخت افزار

[۱] Configuration Space

مورد استفاده در ربات‌ها بسیار اهمیت پیدا کرده است. به عبارت دیگر استفاده از ربات‌هایی که اطلاعات بسیار اندکی از محیط را درک می‌کنند (مجهز به حسگرهای محدود [۲] هستند) و با پردازش بسیار کم به هدف می‌رسند مورد توجه قرار گرفته است. بنابراین معرفی الگوریتم‌هایی که برای چنین ربات‌هایی مسیرهای نزدیک به بهینه را پیدا می‌کند، نسبت به الگوریتم‌هایی که مسیرهای دقیق را برای ربات‌های پیچیده پیدا می‌کند، برتری دارند. در این پژوهش فرض ما نیز بر این است که ربات، خواص متریک محیط مانند فاصله‌ها و زوایا را تشخیص نمی‌دهد بلکه ویژگیهای ترکیبیاتی محیط مانند شکاف‌ها (ناپیوستگی عمقی)، ترتیب راس‌های محدوده کاری و غیره را شناسایی می‌کند. این ربات‌ها به نام ربات‌های ساده با حسگرهای محدود مطرح شده‌اند [۳۲، ۵۳]. منظور از هدف مسئله در این‌جا، رسیدن به نقطه خاصی از محیط، شناسایی نواحی خاصی از محدوه کاری یا شمارش اشیا خاصی در محیط کاری می‌باشد. ما به دنبال ارائه یک راهبرد[۳] برخط[۴] برای ربات هستیم طوری‌که مسیری برای ربات بسازد که دارای ضریب رقابتی[۵] با مسیر بهینه باشد. منظور از ضریب رقابتی یعنی اینکه ربات مسیری را طی کند که طول آن ضریب ثابتی از طول کوتاهترین مسیر باشد، یا مسیری را طی کند که تعداد چرخش‌های آن ضریب ثابتی از مسیر با کمترین تعداد چرخش باشد.

به عنوان نمونه می‌توان به ربات‌هایی که برای کاربردهای نظامی استفاده می‌شوند، اشاره کرد. این ربات‌ها اگر ساختار ساده‌ای داشته باشند، می‌توانند با انرژی بسیار محدودی در مدت طولانی به فعالیت در محیط بپردازند. به عنوان نمونه‌ای دیگر می‌توان به ربات‌هایی که در سوانح و حوادثی مانند زلزله و آتش‌سوزی به کار می‌روند اشاره کرد. رباتی که ساختار ساده‌ای دارد می‌تواند در شرایط سخت با انعطاف‌پذیری بیشتر در محیط به جمع‌آوری اطلاعات و امداد بپردازد.

۲.۱ نتیجه‌های به دست آمده

در این پژوهش هدف طرح و حل مسئله‌های هندسی، به ویژه مسئله‌های مسیریابی در محیط‌های ناشناخته است. در اغلب مسئله‌های مطرح شده در زمینه هندسه محاسباتی، نقشه‌ای از محدوده کاری در دسترس است اما در این‌جا نقشه‌ای از محیط کاری در دست نیست. یک (یا چند) ربات ساده در محیط شروع

[۲] Minimal Sensing

[۳] Strategy

[۴] Online

[۵] Competitive Ratio

به حرکت می‌کند. این ربات‌ها اطلاعات محدودی راجع به محیط از طریق حسگرهای محدودشان به دست می‌آورند. هدف این است که علی‌رغم داشتن اطلاعات کم از محیط به گونه‌ای هدایت شوند که به هدف مسئله برسند طوری که از مسیر بهینه برای رسیدن به آن، زیاد فاصله نگیرند. در ادامه نتایج به دست آمده همراه با توضیح مختصری بیان شده است.

۱.۲.۱ راهبرد رقابتی برای هدایت ربات در یک خیابان ناشناخته

اولین مسئله‌ای که مطرح کرده‌ایم، یکی از مسئله‌های معروف هندسه محاسباتی، به نام مسئله هدایت ربات در خیابان[۶] است. خیابان‌ها دسته‌ای از چندضلعی‌های ساده هستند که اگر یک ربات از نقطه شروع s در خیابان شروع به حرکت کند با رسیدن به انتهای خیابان در نقطه t، کل محدوده خیابان را دیده است. در این جا یک ربات ساده که اطلاعات محدودی از محیط ناشناخته اطرافش را درک می‌کند، از نقطه s در خیابان شروع به حرکت می‌کند. هدف این است که این ربات فقط با استفاده از اطلاعاتی که از طریق حسگرش به دست می‌آورد به نقطه t در انتهای خیابان برسد. این ربات فقط قادر است، ناپیوستگی‌های عمقی یا شکاف‌ها در محدوده دیدش را به ترتیب گزارش کند. نکته مهم در این جا این است که مسیری که ربات برای رسیدن به هدف طی می‌کند با کوتاه‌ترین مسیر خیلی تفاوت نداشته باشد. به عبارت دیگر طول مسیری که ربات طی می‌کند ضریب ثابتی از طول کوتاه‌ترین مسیر باشد. برای این ربات یک الگوریتم برخط جستجو ارائه داده‌ایم که این ربات را با استفاده از یک نشانه‌گذار با طی کردن مسیری به طول حداکثر ۱۱ برابر طول کوتاه‌ترین مسیر به هدف می‌رساند. علاوه بر این ثابت کرده‌ایم،برای این ربات هیچ راهبردی وجود ندارد که آن را در طول مسیری کمتر از ۹ برابر کوتاه‌ترین مسیر به هدف برساند [۵۴].

سپس فرض کرده‌ایم که همان ربات ساده مجهز به حسگر شکاف بخواهد یک خیابان متعامد را در طی یک مسیر متعامد جستجو کند. الگوریتمی برای هدایت این ربات ارائه داده‌ایم که این ربات را طی مسیری که طول آن $\sqrt{۲}$ برابر کوتاه‌ترین مسیر است، در خیابان با شروع از s، به نقطه هدف t می‌رساند. نشان داده‌ایم این ربات علی‌رغم سادگی در درک محیط، قادر است همان مسیری را طی کند که یک ربات با دید کامل از محیط طی می‌کند [۵۵].

بر اساس الگوریتم قطعی جستجو در خیابان، یک راهبرد تصادفی ارائه می‌دهیم که در آن، مسافت مورد انتظاری که ربات برای رسیدن به هدف طی می‌کند، ۶/۵۹ برابر کوتاه‌ترین مسیر برای رسیدن به

[۶] Walking in Streets

هدف است. نتیجه این الگوریتم تصادفی تقریبا دو برابر بهتر از الگوریتم قطعی ارائه شده، است [۵۷].

همچنین برای حالتی که دو ربات ساده بخواهند محدوده خیابان را برای رسیدن به یک هدف مشترک جستجو کنند، یک الگوریتم رقابتی ارائه داده‌ایم [۱].

۲.۲.۱ هدایت ربات ساده در یک خیابان ناشناخته در امتداد مسیری با کمترین تعداد چرخش

دوباره مسئله هدایت ربات ساده در یک خیابان را مطرح می‌کنیم. تفاوت این ربات با ربات مطرح شده قبلی در این است که چرخش برای آن هزینه دارد در حالی‌که حرکت در راستای مستقیم هزینه زیادی ندارد. به عبارت دیگر هدف این است که برای ربات ساده مجهز به حسگر شکاف یک راهبرد جستجو ارائه شود که ربات را از نقطه شروع s به نقطه هدف t برساند به طوری‌که مسیر تولید شده برای ربات کمترین تعداد پیوند[۷] را داشته باشد. یک الگوریتم برای هدایت ربات ارائه می‌کنیم که مسیری بهینه با کمترین تعداد پیوند را برای ربات تولید می‌کند.

مفهوم جدیدی به نام شکاف‌های فعال تعریف می‌کنیم. ایده ما در کم کردن تعداد پیوندهای مسیر، حداکثر استفاده از یک راستای انتخاب شده است که این هدف با استفاده از مفهوم شکاف فعال دست یافتنی است. همچنین ثابت می‌کنیم ضریب رقابتی برای این مسیر $۲ - ۱/m$ است. در این ضریب، m تعداد پیوندها در مسیر بهینه‌ای است که با در دست داشتن نقشه محیط به دست می‌آید [۵۶].

۳.۲.۱ پیمایش یک چندضلعی ستاره‌ای توسط یک ربات ساده

ربات ساده مجهز به حسگر شکاف شروع به جستجو در یک چندضلعی ستاره‌ای می‌کند. چندضلعی ستاره‌ای دسته خاصی از چندضلعی‌ها هستند که در آن نقطه‌ای وجود دارد که از آن نقطه کل نقاط چندضلعی قابل دید باشد. به مجموعه چنین نقاطی، هسته چندضلعی گفته می‌شود. هدف این است که ربات از یک نقطه دلخواه درون چندضلعی شروع به حرکت کند و به نقطه‌ای از هسته برسد. نشان داده‌ایم ربات ساده مجهز به حسگر شکاف با این قابلیت قادر نیست با طی کردن فاصله مناسبی به هسته برسد و ممکن است برای رسیدن به هسته، مقدار زیادی از کوتاه‌ترین مسیر تا هسته منحرف شود. بنابراین قابلیت ربات در درک محیط اطراف را افزایش داده‌ایم به طوری‌که این ربات یک قطب‌نما نیز همراه

[۷] Link

دارد. قطب‌نما جهت‌های جغرافیایی را به آن نشان داده و ربات می‌تواند در امتداد آن جهت‌ها نیز حرکت کند. برای هدایت این ربات نیز راهبردی رقابتی ارائه داده‌ایم. این راهبرد مسیر جستجویی برای ربات تولید می‌کند که طول آن حداکثر ۹/۲ برابر طول کوتاه‌ترین مسیر برای رسیدن به هسته چندضلعی است [۸۱]. مسئله را در حالتی که دو ربات ساده از یک نقطه در خیابان شروع به حرکت کرده و برای رسیدن به یک هدف مشترک تلاش می‌کنند، نیز بررسی می‌کنیم. الگوریتمی برای هدایت ربات‌ها ارائه می‌شود که طول مسیری که ربات‌ها طی می‌کنند از ۲ برابر طول کوتاه‌ترین مسیر کمتر است [۱].

۴.۲.۱ راهبرد رقابتی بهینه برای هدایت ربات با استفاده از اطلاعات محلی

هدف، هدایت ربات بدون نگهداری اطلاعات محیط است. ربات فقط بر اساس اطلاعات محلی که از طریق حسگرش به دست می‌آورد، یک مسیر بهینه را که طول آن ۹ برابر طول کوتاه‌ترین مسیر است، برای رسیدن به هدف طی می‌کند. بر اساس راهبرد ارائه شده فوق، یک الگوریتم تصادفی ارائه داده‌ایم که ضریب رقابتی مورد انتظار آن ۵/۳۳ است [۵۸].

۵.۲.۱ راهبرد رقابتی برای هدایت ربات در یک خیابان عمومی

ربات ساده مجهز به حسگر شکاف را با یک قطب‌نما تقویت کرده و از آن برای پیمایش چندضلعی خیابان عمومی استفاده می‌کنیم. در حالتی که چندضلعی خیابان عمومی متعامد است، یک الگوریتم با ضریب رقابتی ۹/۰۶ ارائه می‌دهیم [۵۹].

۳.۱ مقاله‌های حاصل شده از این پژوهش

[1] Tabatabaei, Azadeh, and Mohammad Ghodsi. “Walking in Streets with Minimal Sensing.” The 7th Annual International Conference on Combinatorial Optimization and Applications (COCOA’13). Springer International Publishing (2013): 361-372.

[2] Tabatabaei, Azadeh, and Mohammad Ghodsi. “Optimal Strategy for Walking in Streets with Minimum Number of Turns for a Simple Robot.” The 8th Annual

International Conference on Combinatorial Optimization and Applications (COCOA'14). Springer International Publishing (2014): 101-112.

[3] Tabatabaei, Azadeh, and Mohammad Ghodsi. "Walking in Streets with Minimal Sensing." J. Comb. Optim, 30.2 (2015): 387-402.

[4] Tabatabaei, Azadeh, and Mohammad Ghodsi. "Randomized Strategy for Walking in Streets for a Simple Robot." The 31th Annual European Workshop on Computational Geometry (EuroCG 2015), March 15-18, Ljubljana, Slovenia (2015).

[5] Abouei Mehrizi, Mohammad, Mohammad Ghodsi, and Azadeh Tabatabaei. "Robots' Cooperation for Finding a Target in Streets." In International Conference on Topics in Theoretical Computer Science (TTCS). Springer International Publishing (2015): 30-43.

[6] Tabatabaei, Azadeh, and Mohammad Ghodsi. "A Randomized Strategy for Walking in Streets for a Simple Robot Using Local Information ." Submitted to IPL (2016).

[7] Tabatabaei, Azadeh, Mohammad Ghodsi, and Fardin Shapouri. "A Competitive Strategy for Walking in Generalized Streets for a Simple Robot." The 28th Canadian Conference on Computational Geometry (CCCG 2016).

۴.۱ طرح رساله

مطالبی که در ادامه این رساله آمده است، در بخش‌های زیر طبقه‌بندی می‌شود:

در فصل ۲ مفاهیم و تعاریف اولیه ارائه شده است. از این مفاهیم در فصل‌های آتی استفاده می‌شوند.

در فصل ۳ در مورد حسگرها، حسگرهای استفاده شده در تکنولوژی روز، اهمیت حسگرهای محدود و حسگرهای محدود استفاده شده در این پژوهش توضیح می‌دهیم. همچنین تاریخچه‌ای از مسئله‌های هندسی مطرح شده قبلی در محیط‌های ناشناخته، با تاکید بر نوع حسگرها شرح داده می‌شود.

در فصل ۴ مسئله هدایت ربات مجهز به حسگر شکاف در چندضلعی خیابان مطرح شده و الگوریتمی

برای حل آن ارائه می‌شود. به طور خاص این مسئله را برای حالتی که خیابان متعامد است، را نیز حل می‌کنیم. همچنین برای حالتی که دو ربات با همکاری هم می‌خواهند به هدف برسند، نیز یک راهبرد رقابتی ارائه می‌دهیم.

در فصل ۵ مسئله هدایت ربات در خیابان دوباره مطرح می‌شود؛ چرخیدن برای این ربات بسیار هزینه‌بر است ولی حرکت در راستای مستقیم هزینه‌ای برای آن ندارد. تنها وسیله درک محیط برای این ربات نیز همان حسگر شکاف است. یک راهبرد رقابتی برای این ربات ارائه می‌شود، مسیر تولید شده توسط این راهبرد، ربات را با کمترین تعداد چرخش به هدف می‌رساند.

در فصل ۶ قدرت درک ربات مجهز به حسگر شکاف را، با یک قطب‌نما تقویت می‌کنیم و آن را برای رسیدن به هسته چندضلعی ستاره‌ای هدایت می‌کنیم. برای رسیدن این ربات به هسته نیز یک راهبرد با ضریب رقابتی ثابت ارائه می‌دهیم.

در فصل ۷ الگوریتمی برای هدایت در خیابان با استفاده از اطلاعات محلی ارائه می‌دهیم.

در فصل ۸ محدوده کاری را وسیع‌تر کرده و ربات را در خیابان عمومی هدایت می‌کنیم.

در فصل ۹ خلاصه‌ای از نتایج به دست آمده و مسئله‌های قابل طرح در این زمینه، به عنوان فرصت‌های پژوهش آتی بیان شده است.

فصل ۲

مفاهیم و تعاریف

۱.۲ مسیریابی در محیط‌های ناشناخته

اولین گام مسیریابی در محیط‌های ناشناخته جمع آوری اطلاعات در مورد محدوده کاری با استفاده از حسگرها می‌باشد، همان طور که در شکل ۱.۲ نشان داده شده است.

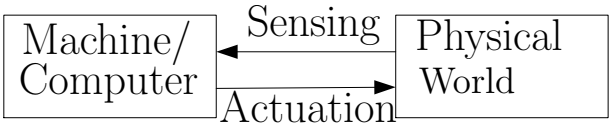
اطلاعات در چنین محیطی به یکی از سه طریق زیر به دست می‌آید.

۱. قبل از اینکه هر عملی انجام شود، اطلاعات زیادی را می‌توان از وضعیت آغازین (در صورت وجود) استخراج کرد. به عنوان مثال، اگر ربات مختصات مکانی را که از آن شروع به حرکت کرده، بداند، پس از یک واحد حرکت به سمت شمال، می‌تواند مختصات جدیدش را به‌دست آورد. در بعضی از مسئله‌ها وضعیت آغازین داده شده و در برخی از حالات هم اطلاعاتی از وضعیت آغازین وجود ندارد.

۲. در حین اجرا، مشاهدات حسگر، اطلاعاتی در ارتباط با وضعیتی که ربات در آن قرار دارد، فراهم می‌کند.

۳. تاریخچه فعالیت‌هایی که سابقا در طی مسیر انجام داده است، اطلاعات ارزشمندی در مورد وضعیت فعلی فراهم می‌کند. به عنوان مثال، اگر یک ربات باید به سمت غرب حرکت کند قابل انتظار است که وضعیت جاری غربی‌تر از وضعیت قبلی باشد. بنابراین تاریخچه فعالیت‌های انجام شده در استنباط وضعیت جاری ارزشمند است.

دو دیدگاه برای حل مسئله‌های مسیریابی در محیط‌های ناشناخته وجود دارد.



شکل ۱.۲: ارتباط بین فضای فیزیکی و ماشین(ربات) از طریق حسگرها و تاثیر ربات بر فضای فیزیکی.

- یک دیدگاه سنتی، به دست آوردن تمام اطلاعات قابل دسترس و سعی در تخمین وضعیت است. در رباتیک منظور از وضعیت (حالت[۱])، نقشه محوطه کاری و پیکربندی ربات است. اگر تخمین به اندازه کافی قابل اطمینان باشد (به واقعیت نزدیک باشد) می‌توان ادعا کرد اطلاعات مربوط به هر وضعیت از دقت خوبی برخوردار است. این باعث می‌شود که بسیاری از الگوریتم‌های طرح مسیر در محیط‌های شناخته شده بدون تغییر یا با اندکی تغییر قابل استفاده باشند. معمولا یک طرح مسیر به صورت $\prod : X \longrightarrow U$ بیان می‌شود، (X محدوده کاری و U فضای فعالیت‌های ربات است) [۱۹، ۲۰، ۳۷].

- وضعیت جالب‌تر (دیدگاه دوم) وقتی اتفاق می‌افتد که تخمین وضعیت و مکان ربات کاملا نادیده گرفته شود. این دیدگاه به این دلیل مورد توجه قرار گرفته است که مسئله‌های بسیاری در زمینه رباتیک می‌تواند بر پایه این دیدگاه تعریف و حل شود. در این حالت یک فضا به نام فضای اطلاعات[۲] برای مسئله‌های مسیریابی تعریف می‌شود. فضای اطلاعات برای مسئله‌هایی که همراه با حسگرها هستند همان نقشی را ایفا می‌کند که فضای پیکربندی برای مسئله‌های حاوی نقشه‌های هندسی ایفا می‌کند. هر نقطه از فضای اطلاعات نمایش دهنده اطلاعی است که از طریق حسگرها به دست آمده است. موفقیت در حل مسئله‌های مسیریابی در محیط ناشناخته بستگی به توانایی ما در نگهداری، ساده‌سازی و کنترل فضای اطلاعات دارد. در بسیاری از موارد راه حل‌های بسیار زیبایی وجود دارد. در حال حاضر در بسیاری از موارد راه حل‌های مناسبی وجود ندارد. اطلاعات مفید (غربال شده[۳]) فراهم شده توسط حسگرها در ساختارهای داده‌ای ذخیره می‌شوند که می‌توانند در مسیریابی مورد استفاده قرار گیرند. در این دیدگاه معمولا یک طرح مسیر به صورت $\prod : I \longrightarrow U$ بیان می‌شود، (I محدوده کاری و U فضای فعالیت‌های ربات است) [۱۰، ۱۷، ۲۱، ۶۵].

---

[۱] State

[۲] Information Space

[۳] Filtered Data

به طور سنتی دیدگاه اول در نظر گرفته می‌شود که به آن مسیریابی بازخوردی[۴] گفته می‌شود. این دیدگاه در [۳۸، ۴۲] به طور کامل شرح داده شده است. تمرکز این پژوهش روی دیدگاه دوم است که روش قدرتمندی برای حل مسئله‌های مسیریابی است.

به طور کلی در مسئله‌های مربوط به طراحی مسیر حرکت یک ربات با حضور حسگرها، سه گام اصلی در طراحی و حل مسئله وجود دارد.

۱. انتخاب نوع حسگری که ربات به آن مجهز می‌شود.

۲. انتخاب ساختار داده‌ای (غربال‌گر اطلاعاتی) که اطلاعات به دست آمده از حسگرها در آن ذخیره می‌شود.

۳. طراحی مسیر حرکت ربات

۲.۲ ربات مجهز به حسگر محدود (ربات ساده)

در این پژوهش فرض بر این است که عاملی که برای مسیریابی در محیط ناشناخته از آن استفاده می‌شود، یک ربات دارای قابلیت درک محدود[۵] از محیط است و با پردازش بسیار کم به هدف می‌رسد. این ربات، خواص متریک[۶] محیط مانند فاصله‌ها و زوایا را تشخیص نمی‌دهد بلکه ویژگی‌های ترکیبیاتی[۷] محیط مانند شکاف‌ها (ناپیوستگی عمقی)، ترتیب راس‌های محدوده کاری و غیره را شناسایی می‌کند. در مواردی هم ممکن است قادر به تشخیص جهت‌های جغرافیایی باشد. در ادامه‌ی این فصل چگونگی عملکرد ربات مجهز به حسگر شکاف و آنچه از محیط کاری درک می‌کند با جزئیات بیشتری توضیح داده شده است.

---

[۴] Feedback Motion Planning

[۵] Minimal Sensing

[۶] Metric

[۷] Combinatorial

۳.۲ محدوده کاری

محدوده‌ای که ربات ساده شروع به شناسایی آن می‌کند و نقشه‌ای از آن در اختیار ندارد یک چندضلعی است. یک چندضلعی P ناحیه بسته‌ای در صفحه است که با تعدادی پاره‌خط که به آن‌ها یال‌های چندضلعی گفته می‌شود، محدود شده است. بین هر دو نقطه در این ناحیه مسیر وجود دارد که هیچ یک از نقاط P را قطع نمی‌کند. نقاط انتهایی یال‌ها راس‌های P نامیده می‌شوند. چون P یک ناحیه بسته است، مرز آن از زنجیرهایی تشکیل شده که توسط یال‌های P به وجود آمده‌اند. در هر زنجیر، هر دو یال متوالی یک راس مشترک دارند. اگر P از یک زنجیر بسته (زنجیری که ابتدا و انتهای آن در یک راس قرار دارند) تشکیل شده باشد، یک چندضلعی ساده[۸] است (شکل ۲.۲آ) و در غیر اینصورت یک چندضلعی حفره‌دار[۹] نامیده می‌شود (شکل ۲.۲ب). یک راس از P، محدب[۱۰] خوانده می‌شود اگر زاویه‌ای که دو یال مجاور به آن راس در داخل چندضلعی می‌سازند، حداکثر ۱۸۰ درجه باشد. اگر این زاویه بیشتر از ۱۸۰ درجه باشد، یک راس انعکاسی[۱۱] نامیده می‌شود.

در بین چندضلعی‌های ساده، ربات ساده ما قادر به شناسایی دسته‌های خاصی از آن‌ها است که در ادامه آن‌ها را شرح می‌دهیم. این چندضلعی‌ها شامل چندضلعی خیابان، خیابان متعامد و چندضلعی ستاره‌ای هستند. برای تعریف آن‌ها به مفاهیم مربوط به رویت‌پذیری نیاز داریم که در ادامه شرح می‌دهیم.

۱.۳.۲ رویت پذیری

دو نقطه p و q (در داخل چندضلعی P) از یکدیگر رویت‌پذیر[۱۲] (قابل دید) هستند اگر پاره‌خطی که این دو نقطه را به هم وصل می‌کند، به طور کامل در داخل P قرار بگیرد و شامل هیچ نقطه‌ای خارج P نباشد. پاره‌خط واصل p و q می‌تواند از یک راس انعکاسی عبور کند یا بر یکی از اضلاع چندضلعی مماس باشد.

---

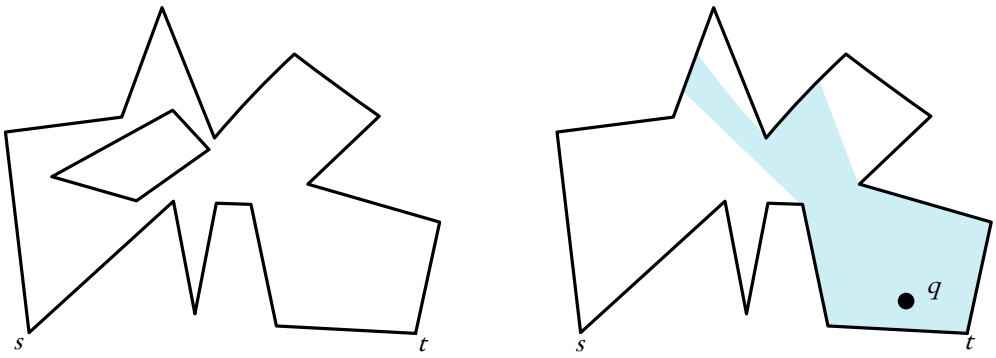
[۸] Simple Polygon

[۹] Polygon with Hole

[۱۰] Convex vertex

[۱۱] Reflex vertex

[۱۲] Visibile



(ب) (آ)

شکل ۲.۲: (آ) یک چندضلعی ساده که زنجیره سازنده مرز آن، با رنگ تیره نشان داده شده است. ناحیه رنگی چندضلعی رویت پذیر نقطه q است. (ب) یک چندضلعی حفره‌دار.

۲.۳.۲ چندضلعی خیابان

یک چندضلعی ساده P با دو راس جداگانه s و t یک خیابان[۱۳] نامیده می‌شود در صورتی که هر نقطه روی یکی از دو زنجیره‌ای که توسط s و t (یکی ساعت‌گرد از s به t و دیگری پادساعت‌گرد از s به t) ساخته می‌شود حداقل توسط یک نقطه از زنجیر دیگر رویت‌پذیر باشد. به عبارت دیگر کلیه نقاط روی چندضلعی توسط هر مسیر از s به t رویت‌پذیر باشند [۳۴] (چندضلعی نشان داده شده در شکل ۲.۲آ یک خیابان است.) مفهوم خیابان در بسیاری از مقاله‌ها به نام چندضلعی‌های $L-R$رویت‌پذیر[۱۴] معروف است [۹، ۶۴]. چنانچه اضلاع خیابان متعامد باشند، این چندضلعی به عنوان یک خیابان متعامد معروف است.

۳.۳.۲ وتر

هر خط افقی در درون چندضلعی که هر دو راس آن روی اضلاع چندضلعی باشد، یک وتر[۱۵] نامیده می‌شود. چنانچه یک راس از وتر روی زنجیر راست و یک راس آن روی زنجیر چپ باشد یک LR–وتر است.

---

[۱۳] Street Polygon

[۱۴] $L-R$ Visible

[۱۵] Chord

۴.۳.۲ خیابان تعمیم یافته

یک چندضلعی ساده یک خیابان تعمیم یافته[۱۶] است در صورتی که هر نقطه روی مرز چندضلعی از روی حداقل یک LR−وتر دیده شود.

در شکل ۱.۸ چندضلعی خیابان و تعدادی از وترهای آن نمایش داده شده است.

۵.۳.۲ چندضلعی ستاره‌ای

یک چندضلعی ستاره‌ای[۱۷] است اگر نقطه‌ای در آن وجود داشته باشد که از آن نقطه، کل مرزهای چندضلعی قابل دیدن باشد. به مجموعه چنین نقاطی هسته چندضلعی گفته می‌شود. نمونه‌هایی از چندضلعی‌های ستاره‌ای در شکل‌های ۱.۶آ و ۲.۶ آمده است.

۶.۳.۲ چندضلعی رویت‌پذیر

مجموعه همه نقاطی از چندضلعی P که از یک نقطه دلخواه q درون P قابل دید هستند، چندضلعی رویت‌پذیر[۱۸] نقطه q نامیده می‌شود. این چندضلعی با $V(q)$ نمایش داده می‌شود (شکل ۲.۲آ). واضح است که با حرکت نقطه در محدوده کاری، چندضلعی رویت‌پذیر تغییر می‌کند.

۴.۲ حسگر شکاف

ربات مجهز به حسگر شکاف قادر به شناسایی ناپیوستگی‌های عمقی (شکاف[۱۹]) است (شکل ۷.۳). این ربات یک اشعه لیزری همراه دارد که به دیوارهایی که در محدوده ۳۶۰ درجه در اطراف آن قرار دارند، برخورد می‌کند. پس از اینکه این اشعه از یک دیوار نزدیک به یک دیوار دور برخورد کند، یک شکاف راست را گزارش می‌کند. وقتی از یک دیوار دور به یک دیوار نزدیک برخورد می‌کند، یک شکاف چپ را گزارش می‌دهد [۴۴، ۶۲، ۶۳]. به عبارت دیگر آنچه این ربات در هر لحظه از محیط

[۱۶] Generalized Street

[۱۷] Star Polygon

[۱۸] Visibility Polygon

[۱۹] Gap

اطراف گزارش می‌دهد، مجموعه‌ای از شکاف‌ها با برچسب چپ یا راست است. با حرکت ربات در محدوده کاری و تغییر چندضلعی رویت‌پذیرش، مکان و تعداد شکاف‌های چپ و راست در هر لحظه می‌تواند تغییر کند (شکل ۲.۴).

۱.۴.۲ شکاف‌های پیشرفته و ساختار داده S-GNT

در حین مسیریابی، از بین شکاف‌هایی که در هر لحظه توسط حسگر شکاف گزارش می‌شوند، دو شکاف نقش کلیدی دارند. این شکاف‌ها شکاف‌های پیشرفته راست و چپ هستند. از بین شکاف‌های راست موجود، شکافی را که در سمت چپ بقیه قرار دارد پیشرفته‌ترین شکاف راست می‌نامیم. از بین شکاف‌های چپ موجود شکافی که در سمت راست بقیه قرار دارد پیشرفته‌ترین شکاف چپ می‌نامیم. چگونگی ساخت و به‌روز رسانی این ساختار داده در بخش ۱.۲.۴ به طور مفصل شرح داده شده است.

۲.۴.۲ وضعیت قیف

با حرکت ربات و تغییر ساختار ناحیه رویت‌پذیر ربات، مکان کل شکاف‌ها از جمله شکاف‌های پیشرفته چپ و راست تغییر می‌کنند. در هر نقطه از مسیر حرکت که هر دو شکاف پیشرفته چپ و راست وجود دارند، ربات با یک وضعیت قیف[۲۰] روبرو است (شکل ۲.۴ب را ببینید.)

۳.۴.۲ شکاف فعال

در حین مسیریابی برای ربات ساده، برای اینکه تعداد چرخش‌های ربات برای رسیدن به هدف را کمینه کنیم، مفهومی به نام شکاف فعال را تعریف کرده‌ایم. اگر حرکت در یک راستای انتخاب شده باعث شود ناحیه مخفی پشت یک شکاف کاهش پیدا کند، این شکاف فعال[۲۱] است در غیر اینصورت شکاف غیر فعال است. برخی از رخدادهایی که باعث تغییر ساختار شکاف‌های موجود می‌شوند، می‌توانند وضعیت فعال بودن شکاف‌ها را نیز عوض کنند. جزییات بیشتر در مورد این تغییرات در فصل ۵ آمده است.

[۲۰] Funnel

[۲۱] Active Gap

۵.۲ قوی کردن ربات

موارد متعددی پیش می‌آید که ربات مجهز به حسگر شکاف قادر نیست به نحو مطلوبی به هدف برسد. به عنوان مثال ممکن است از مسیر بهینه‌ی رسیدن به هدف به اندازه بسیار زیادی دور شود. در این شرایط قدرت درک از محیط با استفاده از تجهیزات دیگری افزایش پیدا می‌کند[۲۲]. از این قبیل تجهیزات می‌توان به نشانه‌گذار[۲۳] و قطب‌نما[۲۴] اشاره کرد. نشانه‌گذار وسیله‌ای مانند یک سنگ یا هر چیز دیگری است که برای ربات قابل شناسایی باشد. ربات می‌تواند هر جا از محیط کاری را با آن علامت‌گذاری کند. هر وقت که آن نشانه‌گذار در محدوده دیدش قرار بگیرد می‌تواند آنجا را به یاد آورد. ضمن اینکه می‌تواند نشانه‌گذار را برداشته و با خودش ببرد و دوباره استفاده کند. در مورد قطب‌نما هم می‌توان گفت که وقتی که ربات همراه خودش یک قطب‌نما دارد قادر به شناسایی جهت‌های جغرافیایی بوده و می‌تواند در امتداد این جهت‌ها گام بردارد.

۶.۲ تحلیل رقابتی

جهت ارزیابی کارایی[۲۵] الگوریتم‌های ارائه شده در این پژوهش از تحلیل رقابتی[۲۶] استفاده می‌شود. در حالتی که هدف کوتاه کردن طول مسیر حرکت ربات است، منظور از تحلیل رقابتی، به دست آوردن نسبت مسافتی که ربات طی می‌کند به کوتاه‌ترین مسیر[۲۷] برای رسیدن به هدف است. در حالتی که هدف کم کردن تعداد چرخش‌ها در طول مسیر حرکت ربات است، منظور از تحلیل رقابتی، به دست آوردن نسبت تعداد چرخش‌هایی که ربات انجام می‌دهد به تعداد چرخش‌ها در مسیر با کمترین تعداد چرخش[۲۸] است. در هر دو حالت فوق به عدد به دست آمده، ضریب رقابتی گفته می‌شود. به طور معمول در طراحی مسیر برای ربات، به دنبال پیدا کردن راهبردی هستیم که مسیری با ضریب رقابتی ثابت برای حرکت ربات تولید می‌کند.

---

[۲۲] Empower

[۲۳] Pebble

[۲۴] Compass

[۲۵] Performance Evaluation

[۲۶] Competitive Analysis

[۲۷] Shortest Path

[۲۸] Minimum Link Path

۷.۲ جمع‌بندی

در این فصل مفاهیم اولیه و تعاریف استفاده شده در ادامه رساله به طور مختصر شرح داده شده است. برای مطالعه جزییات بیشتر در مورد هر یک از آن‌ها می‌توان به بخشی که در آن استفاده شده‌اند، مراجعه کرد.

فصل ۳

حسگرها

با توسعه مرزهای علوم کامپیوتر، اینترنت و فناوری اطلاعات تعامل ماشین (ربات) با دنیای خارج اهمیت پیدا کرده است. در طراحی سیستم‌های مهندسی برای حل مسائلی مثل دیده‌بانی محیط، برقراری امنیت و حمل و نقل خودکار و حتی تعمیر لوازم خانگی با چالش‌های جدیدی روبرو هستیم. در این سیستم‌ها، اطلاعات مستقیما از طریق حسگرها به دست می‌آید. وسایل فیزیکی دیگری ممکن است جهت ارتقای توانایی به دست آوردن اطلاعات یا انجام فعالیت‌های فیزیکی توصیه شود. طراحی موفق و توسعه چنین سیستم‌هایی بستگی به درک اثر متقابل چگونگی حس کردن محیط بر نحوه‌ی به کار گرفتن اطلاعات حس شده و میزان محاسبات دارد. با توجه به این موارد، یک موضوع اصلی مطرح می‌شود: میزان اطلاعات مورد نیاز برای انجام یک کار خاص به طوری که پیچیدگی کل سیستم کاهش یابد، چقدر است. این می‌تواند باعث ارتقای بهره‌وری (هوشمندی ربات) و کاهش هزینه‌ها شود. تغییرات کوچک در مدل ربات‌ها می‌تواند منتهی به تغییرات مهم در قابلیت‌های ربات شود. بنابراین یک ارزیابی شفاف از قدرت ربات قطعا مورد نیاز است.

۱.۳ وضعیت تکنولوژی کنونی و روندهای آن در ماشین‌های هوشمند و خودمختار

یک ماشین هوشمند یک وسیله نقلیه است که قابلیت درک محیط و حرکت در محیط را دارد. ماشین‌های خودمختار اطرافشان را با تکنولوژی‌هایی مثل رادار[۱]، لیدار[۲]، جی‌پی‌اس[۳]، فاصله سنج[۴] و بینایی کامپیوتر[۵] درک می‌کنند. سیستم‌های کنترلی پیشرفته، اطلاعات حس شده را برای مشخص کردن یک مسیر حرکت مناسب، ترجمه می‌کنند درحالی که علائم و موانع موجود را تشخیص می‌دهند. ماشین‌های خودمختار دارای سیستم های کنترلی هستند که قادرند، بین ماشین‌های مختلف در جاده تمایز قائل شوند و این اهمیت ویژه‌ای در طراحی مسیر دارد.

اولین ماشین‌های هوشمند بدون سر نشین[۶] (خودمختار) در دهه ۱۹۸۰ توسط مراکز تحقیقاتی در دانشگاه‌ها و شرکت‌های خودروسازی ساخته شدند. از جمله این مراکز می‌توان به دانشگاه کارنگی ملون و شرکت مرسدس بنز اشاره کرد [۷۴]. از آن زمان به بعد شرکت‌ها و مراکز پژوهشی زیادی شروع به کار در این حوزه کردند، ازجمله دانشگاه آکسفود و شرکت گوگل و کارخانه‌هایی مثل تویوتا، جنرال موتور و نیسان و غیره. در سال ۲۰۱۵ ایالتهای نوادا، فلوریدا، کالیفرنیا، میشیگان همراه با واشنگتن دی سی برای آزمایش خودروی بدون سرنشین تمام خودکار گوگل به ویرجینیا پیوستند [۵۰]. در اروپا، کشورهایی مثل فرانسه، ایتالیا، انگلیس در حال طراحی سیستم‌های حمل و نقل با خودروهای بدون راننده هستند و در کشورهایی مثل آلمان و لهستان خودروهای بدون سرنشین در ترافیک هم آزمایش شده‌اند [۱۶]. در ادامه دو نمونه از پیشرفته‌ترین خودروهایی را که آخرین تکنولوژی روز در آن‌ها استفاده شده است، معرفی می‌کنیم.

---

[۱] Radar

[۲] Lidar

[۳] GPS

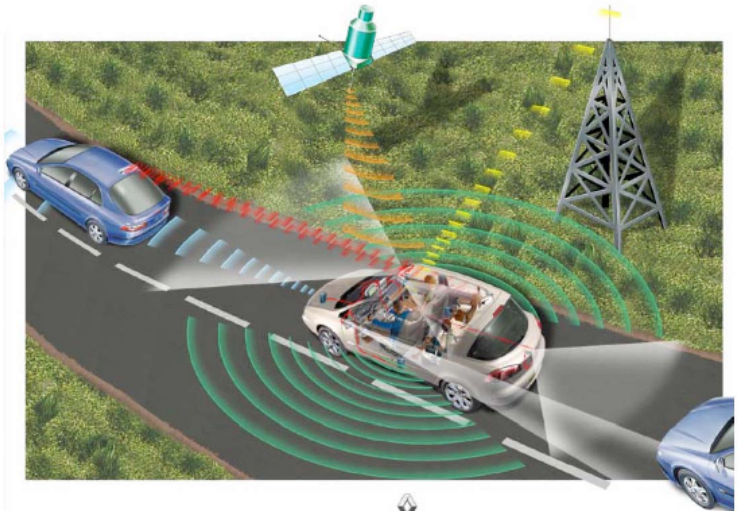
[۴] Odometry

[۵] Computer Vision

[۶] Self-Driving Car

خودروی هوشمند تسلا

خودروی اتوپایلت نیمه خودکار تسلا [۷] دارای ویژگی‌های شگفت‌انگیزی است که به منظور کمک به کاهش تصادفات رانندگی، توسط یک شرکت آمریکایی طراحی شده است. سازنده خودرو مدعی است این خودرو تا پنجاه درصد حوادث رانندگی را کاهش می‌دهد. این خودرو در حین حرکت به توجه مداوم یک انسان نیاز دارد و نمی‌توان این سیستم را به حال خود رها کرد [۳۳، ۷۹]. سیستم اتوپایلت از حسگرهای مختلفی تشکیل شده که دور تا دور آن نصب شده‌اند. این حسگرها به خودرو کمک می‌کنند که محیط پیرامونش را درک کند و با امنیت کامل در بزرگراه‌ها هدایت شود. در سیستم اتوپایلت تسلا یک رادار جلویی، دوربین جلو، جی‌پی‌اس، سیستم الکتریکی کمک ترمز کنترل شونده دیجیتالی و دوازده حسگر دوربرد فراصوت [۸] تعبیه شده است که دور تا دور بدنه کار گذاشته شده‌اند و قادرند تا ۴/۸ متر اطراف خودرو را در هر جهتی حس کنند (ببینند). این حسگرها به ماشین امکان می‌دهند که وقتی چیزی بیش از اندازه به خودرو نزدیک شد، فاصله مناسب را با آن حفظ نموده و مثلا با عوض کردن خط حرکت، از بروز سانحه جلوگیری نمایند، در شکل ۱.۳ مجموعه حسگرهایی که در یک خودروی هوشمند پیشرفته[۹] مثل تسلا به کار می‌رود، نمایش داده شده است. مواردی نظیر گرد و غبار می‌تواند روی این حسگرها را بپوشاند و مانع از دید محیط پیرامون توسط آن‌ها شود.



شکل ۱.۳: نمایشی از حسگرها در خودروی هوشمند [۷۸].

[۷] Semi-Autonomous Autopilot

[۸] Ultrasonic

[۹] Sensors in ADAS (Advanced Driver Assistance Systems)

رادار، شناسایی ماشین‌ها و دیگر اشیاء در حال حرکت را میسر می‌کند. دوربین جلویی که در بالای شیشه جلوی ماشین تعبیه شده است، امکان تشخیص موانع پیش روی را، برای خودرو فراهم می‌کند. این دوربین در اصل به منزله‌ی چشم ماشین است که به ماشین امکان می‌دهد ترافیک، عابرین پیاده، علائم جاده‌ای، خطوط جدا کننده جاده و هر چیز دیگری در مسیرش را تشخیص دهد.

برای فعال نمودن اتوپایلت باید دکمه کنترل مربوطه را دو بار به داخل فشار داد تا ماشین هدایت خود را به دست بگیرد. در زمانی که سیستم اتوپایلت فعال است، ماشین می‌تواند در یک خط به صورت خودکار حرکت نماید، خط عوض کند، سرعت خود را مدیریت نماید و کنترل ترمز را ضمن رانندگی در اختیار بگیرد.

ماشین‌های تسلا همچنین قابلیت پارک دوبله و پارک خودکار در موقعیت عمودی را دارند. این شرکت سیستم خود را به روز رسانی کرد به نحوی که حالا به لطف آن، خودرو می‌تواند بدون حضور راننده در ماشین نیز وارد پارکینگ شده یا از آن خارج گردد. راننده‌ها همچنین می‌توانند ماشین خود را فراخوانی کنند تا به صورت خودکار به موقعیت رفته و سوارشان کند. البته سیستم اتوپایلت تسلا هنوز با محدودیت‌های زیادی رو به رو است و مثلا نباید در مناطق مسکونی حاوی چراغ‌های راهنمایی یا تابلوی توقف از آن استفاده نمود. در واقع این سیستم صرفا برای استفاده در بزرگراه‌ها طراحی شده است. سیستم اتوپایلت به گونه‌ای طراحی شده که تشخیص می‌دهد آیا دستان راننده روی فرمان قرار دارند یا خیر. اگر دست روی فرمان نباشد، سیستم به صورت تصویری به وی هشدار خواهد داد و با انتشار یک فرمان صوتی درخواست می‌کند که کنترل فرمان را در اختیار بگیرید. سیستم اتوپایلوت تسلا، همانند دیگر خودروهای تولیدی توسط این شرکت، مرتبا بهبود داده می‌شود.

خودروی بدون سرنشین گوگل

این خودروها، مدل تغییر شکل یافته‌ی لکسوس[۱۰] یا تویوتا[۱۱] هستند که به انواع گیرنده‌ها برای تشخیص جزئیات پیرامونشان (ثابت یا متحرک)، ماشین‌های دیگر و یا عابرین پیاده، مجهز شده‌اند. داخل این اتومبیل نیز تعدادی دوربین و یک دکمه‌ی بزرگ وجود دارد که با فشردن آن راننده می‌تواند کنترل ماشین را به دست بگیرید [۲۶]. گوگل سال‌هاست در سانفرانسیسکو بر روی ماشین‌های بدون نیاز به راننده، کار می‌کند و تاکنون توانسته بیش از ۲ میلیون و ۴۰۰ هزار کیلومتر بدون تصادف حاد را ثبت کند؛

[۱۰] Lexus RX450h hybrid

[۱۱] Toyota Prius

تصادف‌های جزیی هم که پیش آمده، به ادعای گوگل، به دلیل خطای کامپیوتری نبوده است [۱۸].

خودروهای بدون راننده‌ی گوگل برای تشخیص محیط از تکنولوژی لیدار[۱۲] (استفاده از امواج با طول موج کوتاه برای تصویربرداری ۳ بعدی) استفاده می‌کند. این دستگاه در وسط سقف ماشین نصب شده است. گوگل می‌گوید هر یک از این دستگاه‌های نصب شده روی ماشین‌ها حدود ۷۰ هزار دلار برای شرکت هزینه داشته است. البته گوگل امیدوار است با تولید انبوه خودروهای بدون راننده‌ی خود، بتواند این هزینه را تعدیل کند. بدون در نظر گرفتن حسگر بزرگ لیدار در بالای ماشین، این ماشین‌ها ظاهری کاملا شبیه ماشین‌های عادی دارند. به جای لوگوی بزرگ لکسوس در وسط جلوبندی این خودروها، راداری قرار گرفته که می‌تواند از آن چه جلوی ماشین می‌گذرد اطلاعات جمع‌آوری کند و همچنین می‌تواند سرعت حرکت آن‌ها را تشخیص دهد. این رادار به کامپیوتر مرکزی متصل است و به آن قدرت تصمیم‌گیری برای کاهش یا افزایش سرعت حرکت خودرو را می‌دهد. اگر با دقت نگاه نکنید، شاید اصلا تفاوتی بین لکسوس عادی و مدل بدون راننده‌ی گوگلی آن متوجه نشوید. اما اگر بیشتر نگاه کنید، متوجه دوربین‌ها و حسگرهایی خواهید شد که در بالای شیشه‌ی جلوی ماشین نصب شده‌اند، شکل ۲.۳ را ببینید.

شکل ۲.۳: خودروی بدون سرنشین گوگل.

دور تا دور این خودروها عبارت اخطار دهنده‌ی اتومبیل خودکار[۱۳] را نوشته که دیگر خودروهایی که کنار و یا پشت آن‌ها رانندگی می‌کنند، فاصله مناسب را رعایت کنند. البته مطمئناً این اقدام جنبه‌ی تبلیغاتی هم دارد.

[۱۲] LiDAR (Light Detection and Ranging)

[۱۳] Self-Driving Car

۱.۱.۳ حسگرهای متداول مورد استفاده در ماشین‌های هوشمند

اکنون، به طور خلاصه حسگرهایی را که در فناوری روز رباتیک و به خصوص در خودروهای هوشمند مورد استفاده قرار می‌گیرند در جدول ۱.۳ توصیف می‌کنیم. تصویر و برخی از خصوصیات این حسگرها در شکل‌های ۳.۳ و ۴.۳ آمده است.

(آ) حسگر رادار با برد ۲۵۰ متر و دقت ۰/۰۵ متر [۷۸].

(ب) حسگر لیدار که فاصله ۰/۳ تا ۸۰ متر را تشخیص می‌دهد و دارای دقت ۰/۰۰۵ متر است [۷۸].

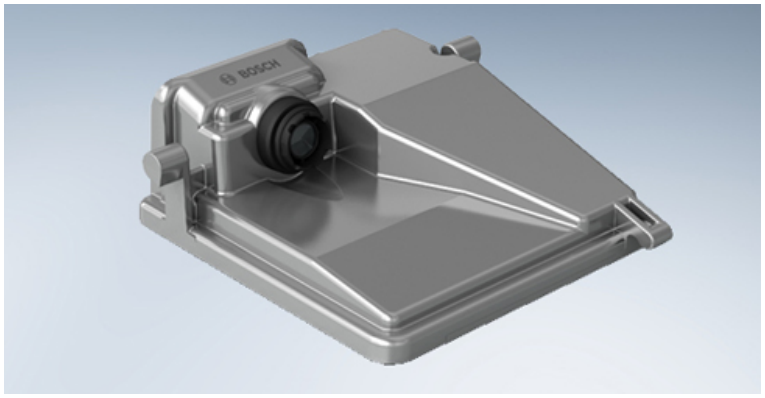
(ج) حسگر جی‌پی‌اس، قابل استفاده برای مکان‌یابی کل نقاط کره زمین با دقت ۰/۰۱ متر [۷۶].

(د) حسگر فاصله‌سنج با دقت ۰/۰۰۱ متر [۷۰].

شکل ۳.۳: حسگرها و برد و دقت آن‌ها.

جدول ۱.۳: حسگرهای متداول مورد استفاده در تکنولوژی حال حاضر رباتیک.

نام حسگر	تکنولوژی به کار رفته	ویژگی (محاسن)	معایب	موارد استفاده	مراجع
رادار	امواج رادیویی	تعیین محدوده، زاویه، یا سرعت اشیاء	هزینه‌ی بالای ردیاب سطوح رادار، نداشتن قابلیت تشخیص سطوح میانی، عدم کارکرد صحیح در هوای خیلی مرطوب	ردیابی هواپیما و کشتی، خودروی هوشمند گوگل و تسلا، ربات بیتل و ربات گی‌رور	[۳۶، ۶۸، ۷۵، ۷۷]
لیدار	اسکن سه بعدی و اسکن لیزری	شناسایی پیرامون از طریق درک مشخصات (مختصات سه بعدی نقاط)، به دست آوردن نقشه سه بعدی محیط	هزینه‌ی بالا و ضعف در اسکن سطوح مرطوب و خیس	ربات‌های ساده تا ماشین‌های خودکار مثل ماشین بدون سر نشین گوگل	[۳، ۷۲، ۷۶]
جی‌پی‌اس	استفاده از ماهواره‌های خاص جی‌پی‌اس	تعیین موقعیت مکانی و زمانی نقاط تحت دید ماهواره‌های جی‌پی‌اس در هر شرایط آب وهوایی	تاخیر در ارسال تغییرات سرعت و جهت	سیستم‌های حمل و نقل هوشمند، خودروهای بدون سر نشین و ربات کاشف	[۴۶، ۶۶]
فاصله‌سنج	محاسبه مسافت طی شده‌ی ربات چرخ‌دار یا پادار	استفاده مطلوب از اطلاعات موجود در برآورد مکان ربات	حساس به خطا	خودروهای بدون سر نشین مثل گوگل و ربات سفری	[۴۹، ۶۹]
بینایی ماشین	تصاویر وفیلم‌های دیجیتالی	اخذ جزئیات محیط	حجم زیادی از اطلاعات که پیوسته تغییر می‌کنند.	ربات‌های انسان‌نما، رباتیک پزشکی، ربات کارگر وخودروهای بدون سر نشین	[۳۸، ۶۷، ۷۴]



شکل ۴.۳: دوربین مناسب برای خودروی هوشمند با برد ۶۰ متر و دقت ۷۴۰ در ۴۸۰ نقطه [۷۳].

۲.۳ چرا از حسگرهای محدود استفاده می‌کنیم؟

در این روزها نوآوری در فناوری انگیزه زیادی را برای ساخت میکرو ربات‌ها به وجود آورده است. به عنوان مثال، حسگرهای دوربین به طور پیوسته کوچک‌تر و ارزان‌تر می شوند و تصاویر با کیفیت بالاتری تولید می‌کنند. این طبیعی است که مهندسان رباتیک هم تمایل به ساخت ربات‌های سطح بالائی پیدا کنند که از کل اطلاعات محدود در دسترس استفاده می‌کنند.

اغلب واضح نیست که بیشتر اطلاعات به دست آمده از طریق حسگرهای پیچیده آیا همان چیزی است که ربات برای حل یک مسئله نیاز دارد. دانش در دست امروز به طور قطعی پاسخگوی این نیست که کدام حسگر برای انجام عمل خاصی مناسب‌تر است. دادن تمام قابلیت‌های حسگری ممکن به یک ربات مانع می‌شود که بتوان به سوال فوق پاسخ داد. اما دادن تمام قابلیت‌ها به ربات رایگان نیست بلکه هزینه‌های اضافی ساخت، حافظه مصرفی و انرژی مصرفی و غیره باید در نظر گرفته شود. ست‌تلر[۱۴] سازنده یک خودروی بدون سرنشین که در بین ماشین‌های رباتی و سایر ماشین‌ها کار می‌کند، در یک سخنرانی، مهم‌ترین جنبه‌ای که به آن اشاره می‌کند، مسئله اضافه بار [۱۵] حسگرها است [۶۰]. او بیان می‌کند که با وجود ۴۰ پردازشگر بسته‌بندی شده در بدنه ماشین، آن‌ها نمی‌توانند تمام اطلاعات به دست آمده از طریق حسگرها را پردازش کنند. واضح است که بسیاری از اطلاعات حس شده مورد استفاده قرار نمی‌گیرد.

با اینکه در این پژوهش، ما یک دیدگاه کاملا تئوری داریم، این مثال نشان می‌دهد که یک پژوهش پایه‌ای که میزان اطلاعات حس شده لازم و کافی برای انجام یک کار خاص (مثلا رانندگی) را مشخص

[۱۴] Seth Teller

[۱۵] Overload

می‌کند، با ارزش است. بنابراین درک محدودیت و قدرت حسگرها یک زمینه پژوهشی با اهمیت است.

در این پژوهش، ما تعدادی از مسائل هندسی در محدوده ناشناخته را مطرح می‌کنیم که قبلا برای ربات مجهز به حسگر قدرتمند نقشه‌ی دید حل شده‌اند. در پژوهش‌های قبلی، رباتی که مجهز به حسگر دید ۳۶۰ درجه محدوده اطرافش بود، محیط را برای رسیدن به هدف جستجو می‌کرد. ربات در هر نقطه‌ای از مسیر که قرار داشته، با در دسترس داشتن نقشه‌ای از محدوده‌ای که تاکنون دیده است، برای رسیدن به هدف، به جستجو ادامه می‌دهد.

۳.۳ تاریخچه‌ای ازحسگرهای محدود هندسی

اولین گام مسیریابی در محیط‌های ناشناخته، جمع‌آوری اطلاعات در مورد محدوده کاری با استفاده از حسگرها است. بنابراین میزان اطلاعاتی که ربات از محیط درک می کند اهمیت بالایی در چگونگی رسیدن به هدف دارد. در این پژوهش فرض بر این است که ربات خواص متریک محیط مانند فاصله‌ها و زوایا را تشخیص نمی‌دهد بلکه ویژگی‌های ترکیبیاتی محیط مانند شکاف‌ها (ناپیوستگی عمقی)، ترتیب قرار گرفتن راس‌های محدوده کاری و غیره را شناسایی می‌کند. این ربات‌ها به نام ربات‌های ساده با حسگرهای محدود مطرح شده‌اند [۳۲، ۴۲، ۵۳]. بعلاوه فرض بر این است که عوامل محیطی مثل دما، رطوبت، نور و غیره بر نحوه‌ی عملکرد ربات تاثیری نداشته و ربات با سرعت ثابت در محیط حرکت می‌کند.

در ادامه تاریخچه‌ای از این مسئله‌ها با تاکید بر نوع حسگر مورد استفاده بیان شده است.

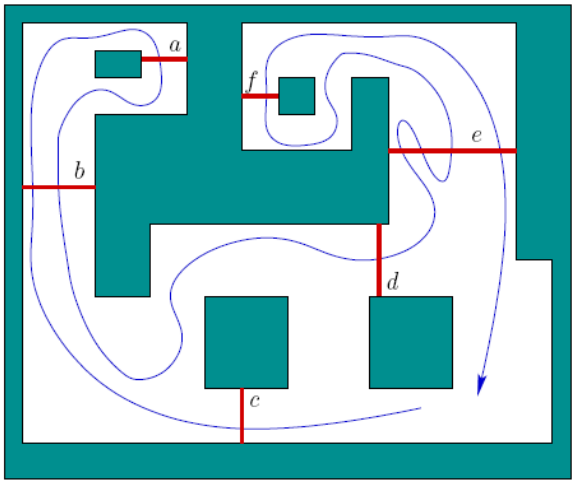
نقشه دید

در بسیاری از پژوهش‌های قبلی در زمینه هندسه محاسباتی در محیط ناشناخته که از یک ربات برای مسیریابی استفاده می‌شود فرض بر این است که ربات در هر نقطه‌ای از محیط که قرار می‌گیرد، کل ویژگی‌های هندسی محیط را، مثل فاصله تا دیوارهای اطرافش یا زاویه‌ها (زاویه‌های چندضلعی رویت‌پذیرش و زاویه‌ای که با نقاط مختلف این چندضلعی دارد) در اختیار دارد. به عبارت دیگر کل مشخصات (نقشه) چندضلعی قابل دیدش را دارد [۱۰، ۱۱، ۲۹، ۳۴، ۴۳]. در این حیطه به مسئله هدایت ربات در خیابان که توسط کلین مطرح شد می‌توان اشاره کرد [۳۴]. مسئله مشابه دیگری که طرح شده است، هدایت چنین

رباتی در یک چندضلعی ستاره‌ای برای رسیدن به نزدیک‌ترین نقطه از هسته‌ی[۱۶] محیط است. برای این مسئله نیز راهبردی با ضریب رقابتی ثابت ارائه شده است. ربات از یک نقطه ثابت s شروع به حرکت می‌کند و هدفش این است که به نقطه‌ای برسد که کل محدوده چندضلعی از آن نقطه قابل دید باشد [۳۰].

موانع و پرتوها

همان طور که در شکل ۵.۳ نشان داده شده است حسگرهای پرتوی[۱۷] در محیط نصب می‌شود [۶۱]. این حسگرهای پرتوی از یکدیگر متمایزند و قادرند حرکت ربات یا هر عامل دیگری از روی خودشان را هشدار دهند. هدف این است که با استفاده از هشدارهایی که در اثر عبور از روی این پرتوها به دست آمده است، اطلاعاتی در مورد مسیر حرکت ربات استنتاج شود. به هشدارهایی که پشت سرهم در اثر حرکت ربات از روی این پرتوها صادر می‌شود یک کلمه حسگری[۱۸] گفته می‌شود. به عنوان مثال، کلمه حسگری مربوط به مسیر نشان داده شده در شکل ۵.۳، برابر $cbabdeeefe$ است.



شکل ۵.۳: مسیری که از روی پرتوها عبور کرده است [۶۱].

این حسگرها که حسگرهای محدودی هستند، محدوده کاری را به چند ناحیه تقسیم می‌کنند. ربات با حرکت روی یک مسیر خاصی به تعدادی از این نواحی برخورد می‌کند. از هر کلمه حسگری صادر

[۱۶] Kernel

[۱۷] Beam Sensors

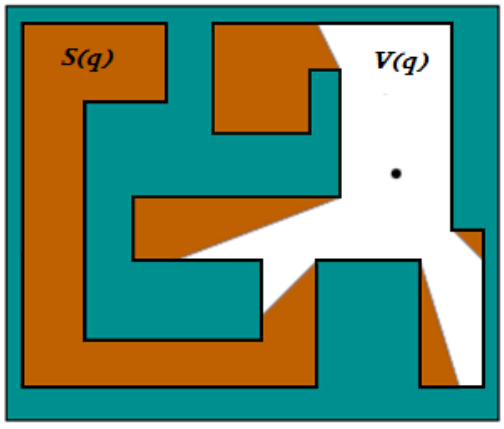
[۱۸] Sensor Word

شده توسط حسگرهای پرتوی می‌توان اطلاعات زیر را استخراج کرد.

- به دست آوردن دنباله‌ای از نواحی محدوده کاری که در اثر حرکت ربات ملاقات شده‌اند.
- مسیرهای معادلی[۱۹] را که ربات می‌توانست طی کند در حالی که همگی دارای کلمه حسگری یکسانی هستند. (دو مسیر معادل‌اند اگر بتوان بدون گذشتن از روی مانعی درون محیط، یکی را به دیگری تبدیل کرد.)
- به دست آوردن تعداد چرخش‌های ربات حول موانع موجود در محیط.

اطلاعات سایه

حسگر سایه‌ها[۲۰] قادر است مکان کل نواحی ناپیدای اطرافش را تشخیص دهد. این اطلاعات در داخل یک گراف دوبخشی ذخیره می‌شود و از این گراف برای استنتاج دانش در مورد کل محدوده کاری استفاده می‌شود (شکل ۶.۳).



شکل ۶.۳: نواحی تیره رنگ سایه‌های تشخیص داده شده توسط حسگر هستند [۳۹].

در ادامه یک مسئله که برای چنین رباتی طرح شده است، بیان می‌شود [۶۵].
یک ربات که قادر به تشخیص سایه‌ها در محیط است، روی یک مسیر از قبل مشخص حرکت می‌کند. در این محدوده کاری، تعدادی نقطه‌ی هدف از قبل در مکان‌های مختلف مخفی شده‌اند که می‌توانند حرکت کنند. یک توزیع از قبل مشخص روی تعداد نقاط هدف مخفی شده در سایه‌ها در اختیار ربات

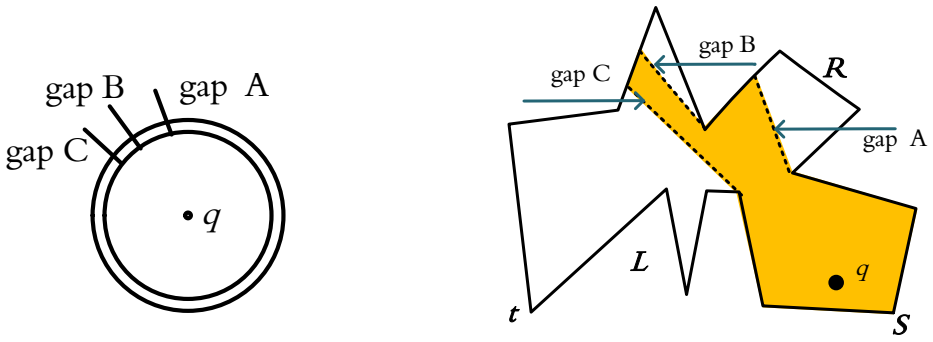
[۱۹] Homotopic

[۲۰] Shadow Information Space

است. هدف این است که در انتهای مسیر حداقل و حداکثر تعداد نقاط هدفی که دیده نشده‌اند مشخص شود.

پیمایش شکاف

این حسگر قادر به شناسایی ناپیوستگی‌های عمقی (شکاف[۲۱]) و همچنین نقاط هدف است (شکل ۷.۳). اطلاعات به دست آمده از حسگر در ساختار داده پویای درخت پیمایش شکاف [۲۲] ذخیره می‌شود که در [۴۴، ۶۲، ۶۳] به عنوان ابزاری برای پیمایش محیط ناشناخته معرفی شده است. اکنون توضیح مختصری در مورد نحوه ساخت این درخت ارائه می‌دهیم.



(ب) (آ)

شکل ۷.۳: (آ) ناحیه رویت‌پذیری نقطه q و شکاف‌هایی که حسگر شکاف گزارش می‌دهد. (ب) موقعیت شکاف‌ها.

در ابتدای حرکت ربات، درخت حاوی یک ریشه به نام $r(T)$ و یک مجموعه از فرزندان به ازای هر یک از شکاف‌های موجود در آن لحظه است. فرزندان به صورت دایره‌ای به همان ترتیبی که شکاف‌ها ظاهر می‌شوند، در درخت ذخیره می‌شوند. با عبور ربات از روی خطوط انعکاس[۲۳]، رویدادهای پدیدار[۲۴] و ناپدید شدن [۲۵] شکاف‌ها رخ می‌دهد. با عبور ربات از روی خطوط مماس مشترک[۲۶]، رویدادهای

[۲۱] Gap

[۲۲] Gap Navigation Tree(GNT)

[۲۳] Inflection

[۲۴] Appearance

[۲۵] Disappearance

[۲۶] Bitangent

تقسیم[۲۷] و ادغام[۲۸] اتفاق می‌افتد. هر یک از گره‌ها در این درخت دارای یک برچسب L یا R است. این برچسب نشان می‌دهد که قسمتی از چندضلعی که در پشت شکاف مربوطه است، در سمت راست شکاف یا چپ شکاف قرار دارد. با حرکت ربات و روی دادن رخدادهای بحرانی، درخت مطابق آن‌چه در ادامه آمده، به‌روز می‌شود. اگر رخداد ناپدید شدن فاصله اتفاق بیفتد، در درخت گره مربوط به آن حذف می‌شود. اگر رخداد پدیدار شدن شکاف اتفاق بیفتد، یک گره به درخت اضافه می‌شود به طوری که ترتیب دایره‌ای شکاف‌ها حفظ شود. گره‌هایی که به این ترتیب اضافه می‌شوند، گره بدوی[۲۹] نامیده می‌شوند و نشان دهنده محدوده‌هایی هستند که قبلا رویت‌پذیر بوده‌اند ولی الان مانعی جلوی دید آن‌ها را گرفته است. اگر دو گره با هم ادغام شوند، دو گره ادغام شده فرزندان یک گره جدید d می‌شوند و d فرزندی از ریشه می‌شود. رخداد ادغام فقط بین شکاف‌هایی که در مجاورت هم قرار دارند اتفاق می‌افتد. وقتی که یک شکاف تقسیم می‌شود، گره مربوط به آن در درخت حذف شده و با دو گره حاصل از تقسیم جایگزین می‌شود (شکل ۸.۳).

از مسئله‌های رویت‌پذیری که با استفاده از این ساختار داده حل شده است، مسئله معروف دزد و پلیس[۳۰] می‌باشد که لاواله و همکارانش حل کردند [۲۱]. حل کردن سایر مسئله‌های رویت‌پذیری توسط این ساختار داده به عنوان مسئله‌های باز در این حیطه معرفی شده است[۴۸].

بردار مشخصه نقطه و بردار رویت‌پذیری ترکیبیاتی

این غربال‌گر، اطلاعات به دست آمده از ربات مجهز به حسگری که قادر است ویژگی‌های ترکیبیاتی محیط اطرافش را تشخیص دهد، نگه می‌دارد. به طور خاص ربات می‌تواند راس‌های محیط اطرافش و همچنین اگر اشیا خاصی مثل نقاط هدف، یا رباتی دیگر یا علامت‌گذاری مثل مهره [۳۱]روی این راس‌ها باشد را تشخیص دهد. ربات می‌تواند تفاوت بین راس‌های معمولی و راس‌های حاوی نقاط خاص را تشخیص دهد. فرض بر این است که ربات با حرکت به صورت چرخشی در جهت خلاف عقربه‌های ساعت، نقاط اطرافش را می‌بیند و آن‌ها را در یک برداری از صفر و یک‌ها به نام بردار مشخصه نقطه

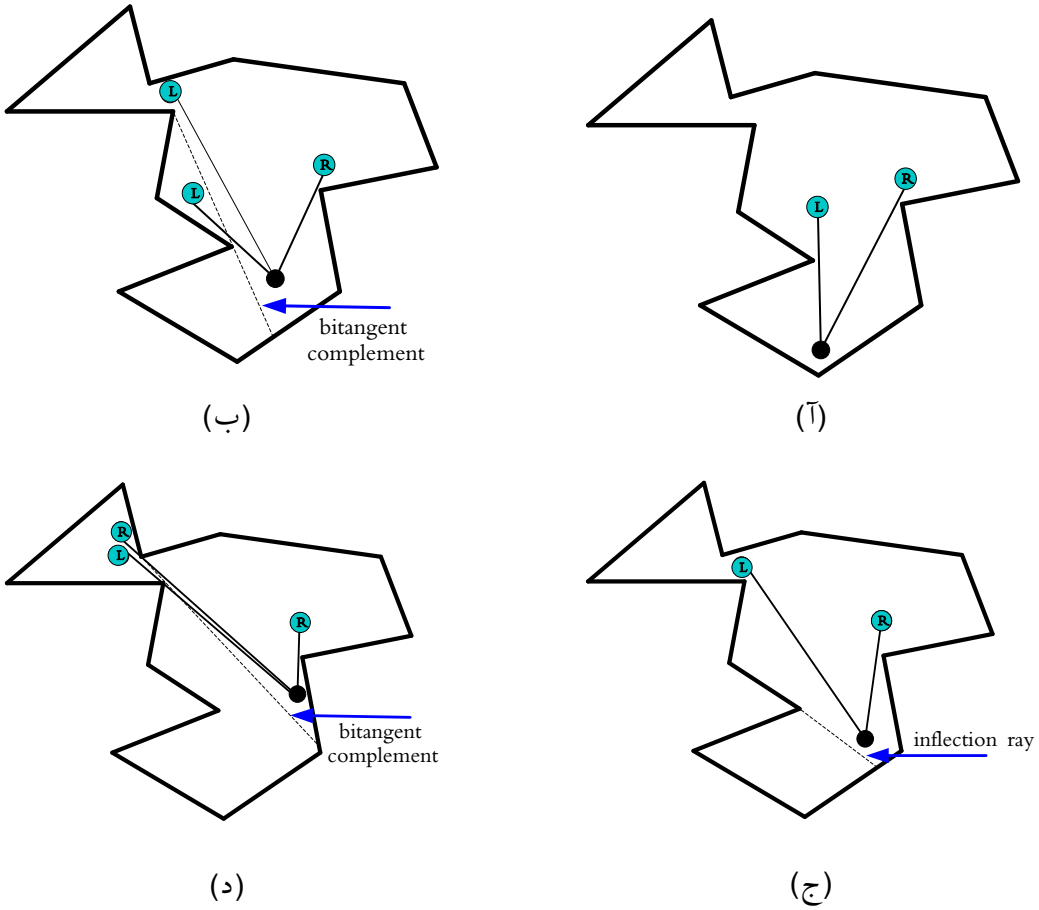
[۲۷] Split

[۲۸] Merge

[۲۹] Primitive Node

[۳۰] Pursuit-Evasion

[۳۱] Pebble



شکل ۸.۳: رخدادهای بحرانی و روش ساخت درخت پیمایش شکاف. (آ) شکاف‌های اولیه. (ب) رخداد تقسیم. (ج) رخداد ناپدید شدن. (د) یک رخداد تقسیم دیگر.

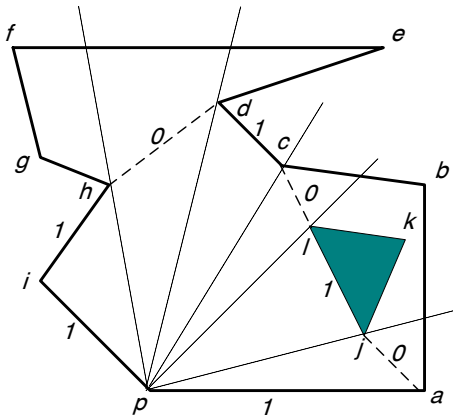
[۳۲] ذخیره می‌کند. عدد ۱ نشان دهنده نقطه خاص و ۰ نشان دهنده راس است. وقتی که ربات روی یک راس از چندضلعی قرار دارد، اولین عنصر این بردار همان راس از چندضلعی است. برای رباتی که درون چندضلعی قرار دارد، هیچ فرضی روی مکان اولین راس بردار در نظر گرفته نمی‌شود. فرض بر این است که بدترین حالت[۳۳] انتخاب می‌شود.

علاوه بر این ربات می‌تواند ضلع‌های چندضلعی را ببیند که این اطلاع از محیط نیز توسط یک بردار رویت‌پذیری ترکیبیاتی مدل می‌شود. این بردار یک بردار به طول k از تعداد راس‌های قابل دید ربات است که مولفه i ام آن در صورتی ۱ بوده که یک یال بین راس i و راس $i-1$ وجود داشته باشد (شکل ۹.۳).

ربات در این محیط جهتی را برای حرکت در نظر می‌گیرد. این جهت می‌تواند به سمت یک نقطه‌ی

[۳۲] Point Identification Vector (piv)

[۳۳] Adversary



شکل ۹.۳: نمایش بردار ترکیبیاتی $cvv(p) = (۱, ۰, ۱, ۰, ۱, ۰, ۱, ۱)$ بین دنباله راس‌های رویت‌پذیر از نقطه p.

هدف یا یک راس باشد. بعد از رسیدن به آن نقطه متوقف می‌شود و دوباره از محیط اطلاعات دریافت می‌کند. دریافت اطلاعات در صورتی انجام می‌شود که ربات در حالت توقف باشد. این مدل حسگری، توسط سوری و همکارانش [۵۳] معرفی شد. ایشان با استفاده از این حسگر محدود که اطلاعات محلی از محیط کسب می‌کند بسیاری از ویژگی‌های هندسی محیط را استخراج کردند که از جمله آن‌ها می‌توان به موارد زیر اشاره کرد:

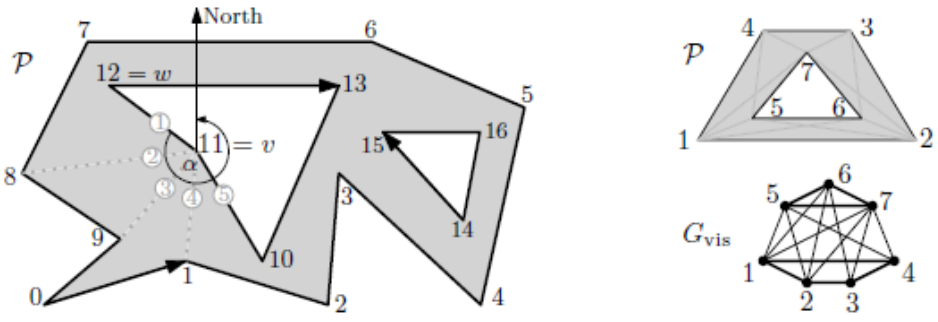
- تشخیص اینکه آیا چندضلعی محیط ساده است یا خیر.
- تشخیص تعداد حفره‌ها درون چندضلعی.
- شمارش تعداد حفره‌ها درون محیط.
- شمارش تعداد نقاط هدف درون محیط.
- مثلث‌بندی کردن محیط و سپس حل مسئله موزه هنر[۳۴].

سناریوهای مختلفی برای حرکت ربات مطرح می‌شود و در مورد برخی اثبات می‌شود که پیدا کردن جواب دقیق امکان‌پذیر نیست و الگوریتمی‌که به طور تقریبی تعداد نقاط هدف را می‌شمارد، ارائه شده است [۱۷، ۵۳].

[۳۴] Art Gallery Problem

تقویت کردن ربات با استفاده از قطب‌نما

دیسر [۳۵] و همکارانش قدرت ربات فوق (با حسگر بردار مشخصه نقطه و بردار رویت‌پذیری ترکیبیاتی) را با اضافه کردن یک قطب‌نما افزایش دادند [۱۴]. این ربات در هر نقطه علاوه بر دو بردار فوق راستای شمال را هم شناسایی می‌کند. به عبارت دیگر، در هر راس از چندضلعی زاویه تشکیل شده یال‌ها با راستای شمال قابل شناسایی است. قابلیت این ربات در شکل ۱۰.۳ (سمت چپ) نشان داده شده است. این ربات با وجود سادگی قادر است بسیاری از ویژگی‌های هندسی از جمله گراف رویت‌پذیری چندضلعی‌های حفره دار را بسازد. به عبارت دیگر، این ربات قادر به نگاشت چندضلعی حفره‌دار[۳۶] است. این در حالی است که همین ربات بدون قطب نما قادر نیست اطلاعات محیط حفره‌دار را استخراج کند.



شکل ۱۰.۳: آنچه ربات از محیط درک می‌کند در سمت چپ نشان داده شده است. در شکل سمت راست یک چندضلعی حفره‌دار و گراف رویت‌پذیری آن که ربات تولید می‌کند، نشان داده شده است [۱۴].

ربات دنبال کننده راستای دیوارها

ربات ساده دیگری به نام بیت‌باتس[۳۷] توسط توار[۳۸] و همکارانش معرفی شد [۳۲]. این ربات فقط قادر است در امتداد دیوارها[۳۹] حرکت کرده و به هر راس که رسید، تشخیص دهد که این راس انعکاسی یا

[۳۵] Yann Disser

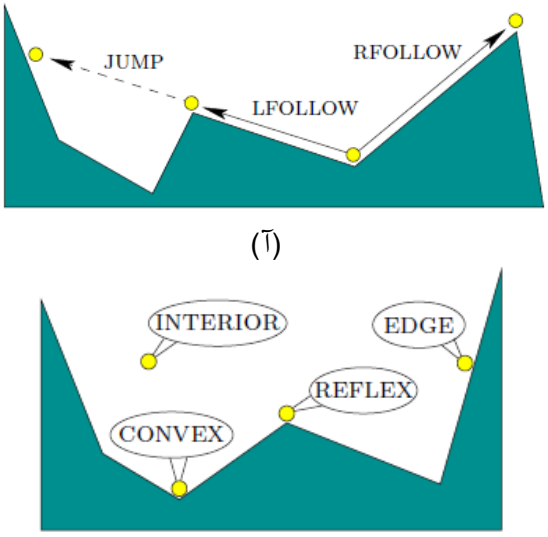
[۳۶] Mapping Polygon with Holes

[۳۷] Bitbots

[۳۸] Benjamin Tovar

[۳۹] Wall Following Robot

محدب است. این ربات در هر راس انعکاسی می‌تواند به حرکت در امتداد دیوار بعدی یا به حرکت در راستای دیوار فعلی ادامه داده و به ضلع مقابل برسد. قابلیت درک ربات از محیط در شکل ۱۱.۳ نمایش داده شده است.



شکل ۱۱.۳: (آ) قابلیت حرکتی ربات بیت‌باتس. (ب) ربات قادر است تشخیص دهد که روی راس انعکاسی یا محدب است و همچنین تشخیص می‌دهد که روی ضلع یا داخل چندضلعی قرار دارد [۳۲].

این ربات ساده هم قادر است بسیاری از ویژگی‌های هندسی محیط از جمله موارد زیر را استنباط کند.

- شمارش تعداد راس‌های چندضلعی
- نقشه ترکیبیاتی محیط[۴۰]
- حل مسئله دزد و پلیس در داخل چندضلعی

مسیریابی قابل اعتماد برای یک ربات غیر قابل اعتماد (تقریبا بدون حسگر)

اوکین و همکارانش برای یک ربات ساده‌ای که فقط قادر بود در راستای مشخصی (با مقداری خطا) حرکت کرده تا به یک مانع برخورد کند، یک الگوریتم جستجو ارائه دادند. به عبارت دیگر ربات

[۴۰] Combinatorial Map

مجهز به یک قطب‌نما و یک حسگر برخورد است. در الگوریتم آن‌ها یک گراف جهت‌دار برای پیمایش محیط ساخته می‌شود. ربات با حرکت در یک راستا و برخورد به مانعی، یک گوشه از محدوده کاری را پیدا می‌کند، از آن گوشه شروع به حرکت در راستای دیگری می‌کند. الگوریتم در گراف فوق نقشه کامل محدوده کاری را می‌سازد و با استفاده از آن نقشه، محدوده را برای رسیدن به هدف جستجو می‌کند [۴۱].

سلسله مراتب ربات‌ها

سوری و همکارانش تلاش می‌کنند، نشان دهند که برای انجام هر وظیفه‌ای چه رباتی و با چه قدرت حسگری مورد نیاز است. آن‌ها در یک محدوده چندضلعی، قدرت برخی از ربات‌ها با حسگرهای متفاوت را با یکدیگر مقایسه می‌کنند و نشان می‌دهند که چگونه ربات‌ها با قدرت‌های مختلف قادرند وظایف یکسانی را انجام دهند. آن‌ها بحث هم‌ارزی ربات‌ها را مطرح می‌کنند [۸].

۴.۳ حسگرهای محدود استفاده شده در این پژوهش

مهمترین حسگر محدودی که در این پژوهش مورد استفاده قرار می‌گیرد حسگر شکاف است، همان‌طور که ذکر شد ربات مجهز به این حسگر قادر است ناپیوستگی‌های عمقی موجود در پیرامونش را تشخیص دهد. در کاربردهای نظری معمولاً حسگرها را به صورت انتزاعی[۴۱] مطرح می‌کنند، بدین معنی که پیاده‌سازی‌های مختلفی از حسگر وجود دارد. یک حسگر شکاف را می‌توان با کنار هم گذاشتن دو حسگر فاصله‌سنج نادقیق در کنار هم ساخت. به این صورت که هر گاه فاصله‌ی اندازه‌گیری شده توسط حسگرها اختلاف زیادی داشت، یک شکاف حس می‌شود. به دلیل زمان زیاد لازم برای تحلیل خروجی حسگرهای فاصله‌سنج، مثلاً به دلیل تبدیل آنالوگ به دیجیتال، مقایسه‌ی خروجی دو حسگر که مثلاً به صورت ولتاژ است به مراتب ساده‌تر و سریع‌تر است.

ربات مجهز به حسگر شکاف به تعداد دلخواهی گام در محیط حرکت می‌کند، برای تولید گام برای ربات ساده، می‌توان از موتور پله‌ای[۴۲] استفاده کرد. موتورهای پله‌ای موتورهای جریان مستقیم[۴۳] هستند

[۴۱] Abstract

[۴۲] Stepper Motors

[۴۳] Direct Current

که در گام‌های گسسته کار می‌کنند. با هر تحریک، موتور یک گام در هر زمان می‌چرخد. با این کار می‌توان کنترل دقیقی روی سرعت یا مکان به دست آورد [۷۱]. پس به طور نظری، می‌توان گفت یک گام کمترین میزان حرکت قابل اندازه‌گیری برای ربات دارای موتور پله‌ای است. به همین دلیل تعداد گام‌ها برای چنین رباتی ملاک خوبی از میزان حرکت (فاصله) است. در نتیجه ربات، حتی اگر مانعی را حس کند، نمی‌تواند با دقت بیشتر از گام، مستقیماً به آن برسد. برای کنترل موتورهای جریان مستقیم از تغییر شار، جریان یا ولتاژ آن استفاده می‌کنند که دقت کافی برای به دست آوردن یک گام را ندارد. برای حل این مشکل، ملاک تعداد چرخش را در نظر می‌گیرند. اگر سرعت ربات زیاد و محیط کوچک باشد، تعداد چرخش‌ها زیاد می‌شود و در نتیجه اکثر زمان حرکت صرف چرخش می‌شود.

گاهی اوقات ربات مجهز به حسگر شکاف برای جستجوی بهتر برخی از محیط‌ها، قطب‌نما با خود همراه دارد و همچنین می‌تواند مکان‌هایی را توسط یک نشانه، علامت‌گذاری کند؛ برای اینکه بعدا بتواند آنجا را در مسیریابی شناسایی کند. ربات ساده ما قادر است نشانه‌ها و نقاط هدف را، به طور مثال توسط فرنکاسی که از آن‌ها صادر می‌شود و این فرکانس از دیوارها عبور نمی‌کند، تشخیص دهد.

چرا محیط را با چندضلعی‌ها خاص مدل می‌کنیم؟

در کاربردهایی که ربات نقشه‌ی محیط را ندارد و نمی‌سازد، پیدا کردن مسیر بهینه‌ی سراسری معمولاً امکان‌پذیر نیست. در چنین مواردی پیدا کردن مسیر بهینه‌ی محلی انتخاب دوم است. به همین دلیل از افراز محیط به صورت چندضلعی‌های ستاره‌ای شکل، چندضلعی‌های خیابان و ... استفاده می‌کنند که برای ربات قابل تشخیص هستند. کنار هم گذاشتن چنین چندضلعی‌هایی محیط را می‌پوشاند در نتیجه ربات در طول حرکت خود به صورت محلی، بهینه عمل می‌کند.

۵.۳ اهمیت استفاده از حسگرهای محدود در این پژوهش

در این بخش بیان می‌کنیم که استفاده از حسگرهای محدود در هر یک از مسائلی که در این پژوهش مطرح شده‌اند، چه مزیتی دارد. در مسئله هدایت ربات در یک چندضلعی خیابان و خیابان عمومی استفاده از حسگر محدود به دلایل زیر اهمیت دارد.

حسگر ربات ما فقط قادر به تشخیص مکان شکاف‌ها در چندضلعی رویت‌پذیری ربات است و خواص متریک محیط مثل اندازه زوایه‌ها و فاصله‌ها را تشخیص نمی‌دهد در حالی که در پژوهش‌های

قبلی، ربات قادر به تشخیص چندضلعی رویت‌پذیر با کلیه خواص (نقشه) آن است. این ویژگی منجر به استفاده از یک حسگر ارزان‌تر در ربات به کار گرفته شده در این پژوهش نسبت به مطالعات قبلی می‌شود.

در راهبردهای ارائه شده ما برای هدایت ربات ساده، فقط مکان تعدادی از شکاف‌ها در هر نقطه از مسیر حرکت، نگهداری و به روز می‌شود در حالی که در پژوهش‌های قبلی کل ناحیه‌ای که ربات از نقطه شروع تا مکان فعلی دیده است، نگهداری و به روز می‌شود. بنابراین ربات ما محاسبات کمتری را انجام می‌دهد و با سرعت بیشتری به هدف می‌رسد. این ویژگی منجر به استفاده از یک ربات با پردازنده ساده‌تر و با حافظه کمتر می‌شود. همچنین داشتن محاسبات و حافظه کمتر باعث می‌شود که ربات بتواند با خطای کمتر و انرژی کمتری کار کند. علاوه بر این در الگوریتمی که در فصل ۷ برای هدایت ربات ارائه شده است، فقط مکان دو شکاف خاص نگهداری و به روز می‌شود، بنابراین داشتن یک حافظه از اندازه ثابت برای ربات ساده ما کفایت می‌کند.

۶.۳ جمع‌بندی

در این فصل در مورد حسگرها که وسیله درک محیط ناشناخته توسط ربات هستند، مطالبی را بیان کردیم. نمونه‌هایی از فناوری روز رباتیک در مسیریابی و حسگرهای استفاده شده در آن‌ها را شرح دادیم. به طور کلی مزیت استفاده از حسگرهای محدود را توصیف کردیم. علاوه بر این، در این فصل جهت آشنایی با مسائل هندسی در محیط ناشناخته تعدادی از آن‌ها را مطرح کردیم که این مسائل را بر اساس نوع حسگر ربات مسیریاب در محیط دسته‌بندی کرذیم. به طور خاص ساختار حسگرهای محدود در این پژوهش و مزیت استفاده از آن‌ها را توضیح دادیم.

فصل ۴

راهبرد رقابتی برای هدایت ربات در یک خیابان ناشناخته

در این فصل یکی از مسئله‌های معروف هندسه محاسباتی، به نام مسئله هدایت ربات در خیابان[۱] را طرح و حل می‌کنیم، با این شرط که رباتی که از آن استفاده می‌کنیم، یک ربات مجهز به حسگر ضعیف شکاف[۲] است (شکل ۷.۳).

یک چندضلعی ساده P با دو راس جداگانه s و t یک خیابان نامیده می‌شود در صورتی که هر نقطه روی یکی از دو زنجیری که توسط s و t (یکی ساعت‌گرد از s به t و دیگری پادساعت‌گرد از s به t) ساخته می‌شود، حداقل توسط یک نقطه از زنجیر دیگر رویت‌پذیر باشد. به عبارت دیگر کلیه نقاط روی چندضلعی توسط هر مسیر از s به t رویت‌پذیر باشند [۳۴] (شکل ۱.۴). مفهوم خیابان در بسیاری از مقاله‌ها به نام چندضلعی‌های $L-R$رویت‌پذیر معروف است [۹، ۶۴].

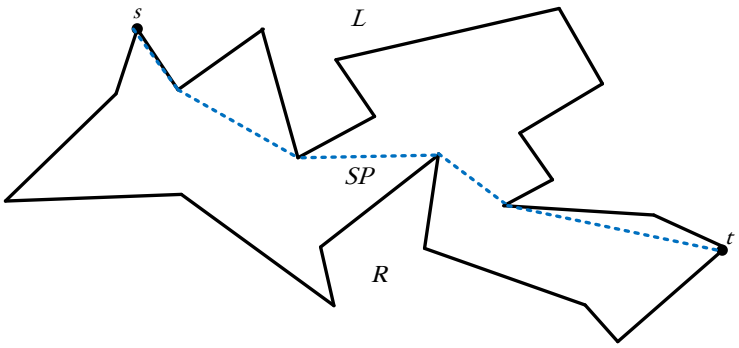
در این جا هدف ارائه یک راهبرد برخط در خیابان برای هدایت ربات نقطه‌ای است که از نقطه s در خیابان با سرعت ثابت شروع به حرکت کرده تا به نقطه هدف t برسد به طوری که طول مسیری که طی می‌کند تا حد امکان کوتاه باشد.

الگوریتمی برای هدایت این ربات در خیابان ارائه می‌دهیم که با کمک گرفتن از یک نشانه‌گذار، ربات مسیری به طول حداکثر ۱۱ برابر طول کوتاه‌ترین مسیر طی کرده و به هدف می‌رسد [۵۴].

---

[۱] Walking in Streets

[۲] Gap Sensor



شکل ۱.۴: یک خیابان که روی نقاط s و t ساخته شده است.

۱.۴ نتایج اولیه

در نقطه شروع حرکت، ربات با مجموعه‌ای از شکاف‌ها با برچسب‌های L و R روبرو می‌شود. با توجه به ساختار محدوده کاری خیابان، نیازی نیست که برای رسیدن به هدف، کل شکاف‌ها پیموده شود. بین شکاف‌هایی که دارای برچسب R هستند، هدف فقط می‌تواند پشت سمت چپ‌ترین (پیشرفته‌ترین) شکاف راست که با g_r نشان می‌دهیم، مخفی باشد و از بین شکاف‌هایی که دارای برچسب L هستند، فقط می‌تواند پشت سمت راست‌ترین (پیشرفته‌ترین) شکاف چپ که با g_l نشان می‌دهیم مخفی باشد [۴۵]، شکل ۲.۴ را ببینید. در هر نقطه از مسیر حرکت اگر فقط یکی از شکاف‌ها وجود داشته باشد، هدف پشت این شکاف بوده و ربات بدون هیچ ابهامی به سمت آن حرکت می‌کند (شکل ۲.۴آ). اگر هر دو شکاف پیشرفته وجود داشته باشند، هدف پشت یکی از آن‌ها قرار دارد. در این حالت که یک قیف نامیده می‌شود، ربات برای رسیدن به هدف مجبور است پشت هر دو شکاف را ببیند (شکل ۲.۴). از آن‌جایی که ربات هیچ اطلاعی در مورد محیط به جز مکان شکاف‌ها ندارد، انحراف از کوتاه‌ترین مسیر غیر قابل اجتناب است. در هر وضعیت قیف، دو زنجیر محدب در مقابل ربات قرار دارد (شکل ۲.۴ب)، این دو زنجیر دارای این خاصیت مهم هستند:

لم ۱.۴ کوتاهترین مسیر برای رسیدن به هدف فقط روی یکی از دو زنجیر محدب (زنجیر محدب راست[۳] یا زنجیر محدب چپ[۴]) قرار می‌گیرد.

اثبات. این لم نتیجه مستقیم قضیه زیر و ویژگی مهم شکاف‌های پیشرفته چپ و راست است که هدف پشت یکی از آن‌ها قرار دارد. ■

[۳] R_{chain}

[۴] L_{chain}

قضیه ۲.۴ [۲۲] برای هر راس $v_i \in L_{chain}(or, v_j \in R_{chain})$، کوتاه‌ترین مسیر از s به v_j چرخش به سمت راست دارد. به همین ترتیب کوتاه‌ترین مسیر به نقاط متعلق به زنجیر چپ چرخش به سمت چپ دارد.

از بین دو زنجیر فوق زنجیری که کوتاه‌ترین مسیر روی آن قرار می‌گیرد زنجیر اصلی این قیف نامیده می‌شود. به راحتی ثابت می‌شود روی هر دو زنجیر نقطه‌ای وجود دارد که وضعیت قیف در آن نقطه به پایان می‌رسد. این نقطه را نقطه بحرانی[۵] آن زنجیر می‌نامیم (شکل ۲.۴ب).

لم ۳.۴ فرض کنید که ربات در حال حرکت در امتداد یکی از زنجیرهای محدب است، ربات به محض عبور از روی نقطه بحرانی مربوط به آن زنجیر، قادر به تشخیص زنجیر اصلی آن قیف است.

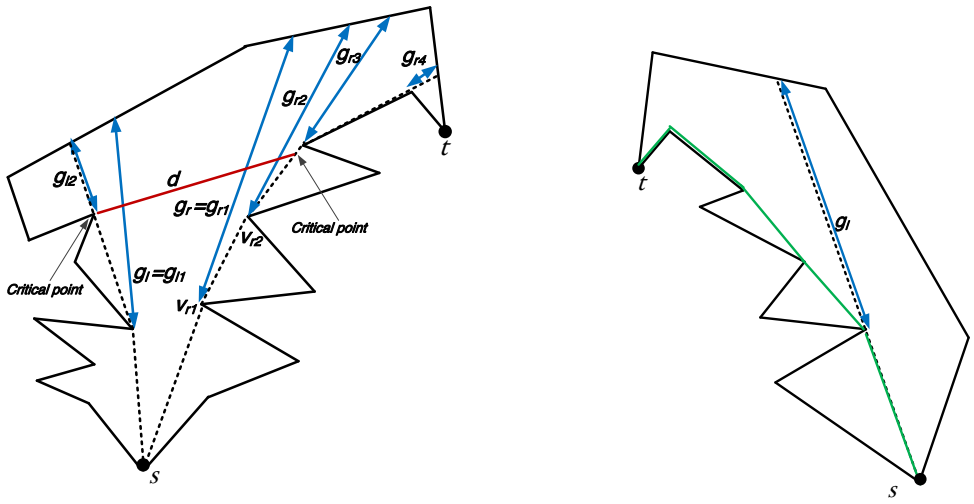
اثبات. ربات در دو حالت به نقطه بحرانی قیف برخورد می کند. ۱- ربات به نقطه ای برسد که یکی از شکاف‌های پیشرفته ناپدید می‌شود. در این حالت زنجیری که حاوی شکاف پیشرفته موجود است، زنجیر اصلی قیف است (شکل ۲.۴ب). ۲- ربات به نقطه ای برسد که هر دو شکاف پیشرفته در یک راستا قرار بگیرند. اگر در این نقطه شکاف پیشرفته‌ای که ربات به سمت آن حرکت می‌کرد ناپدید شود، زنجیر جاری، اصلی است (نقطه ۳ در شکل ۳.۴ب). در غیر اینصورت زنجیر مخالف زنجیر اصلی است (نقطه ۲ در شکل ۳.۴ب).

■

۲.۴ راهبرد اصلی

اکنون الگوریتم خود را برای هدایت ربات ارائه می‌دهیم. لازم به ذکر است که ربات در طول مسیر حرکتش موقعیت شکاف‌های چپ و راست را به‌روز می‌کند. در ادامه رخدادهایی را که باعث تغییر وضعیت شکاف‌ها در طول مسیر حرکت می‌شود، شرح می‌دهیم. زمانی که ربات در مکانی قرار دارد که فقط یکی از این شکاف‌های پیشرفته وجود دارند، بدیهی است که هر راهبرد منطقی ربات را به سمت این شکاف هدایت می‌کند. در حالتی که هر دو شکاف پیشرفته وجود دارند، از راهبرد جستجوی دوگانه

[۵] Critical Points



(ب) (آ)

شکل ۲.۴: g_r سمت چپ‌ترین (پیشرفته‌ترین) شکاف راست و g_l سمت راست‌ترین (پیشرفته‌ترین) شکاف چپ است. (آ) ربات فقط یک راه دارد و بدون ابهامی آن را طی می‌کند. (ب) ربات دو راه دارد که یکی از آن‌ها را انتخاب می‌کند پس احتمالا از کوتاه‌ترین مسیر منحرف می‌شود.

روی خط [۶] ایده گرفته‌ایم. در این راهبرد ربات برای رسیدن به نقطه خاصی از یک خط، در هر مرحله i به اندازه ۲^i به یک جهت حرکت می‌کند سپس به مبدا برمی‌گردد و به اندازه $۲^{i+۱}$ گام در جهت مخالف حرکت می‌کند و دوباره به مبدا حرکت برمی‌گردد. این روند آن‌قدر ادامه می‌یابد تا ربات به نقطه مورد جستجو برسد.

قضیه ۴.۴ [۴] راهبرد جستجوی دوگانه برای رسیدن به نقطه خاصی از خط، بهینه است و دارای ضریب رقابتی ۹ است.

اگر دو زنجیر محدب را به صورت یک خط در نظر بگیریم می‌توانیم با به کار بردن راهبرد جستجوی دوگانه روی این خط به نقطه بحرانی قیف که به مبدا نزدیک‌تر است برسیم. بنابراین هدایت ربات در این محیط به طوری که فقط روی دو زنجیر محدب حرکت کند، یک گام مهم در الگوریتم ما است.

لم ۵.۴ ربات زنجیر محدب راست/چپ را پیمایش می‌کند اگر و فقط اگر به سمت شکاف پیشرفته راست/چپ حرکت کند در حالی‌که به طور پویا موقعیت شکاف‌های پیشرفته چپ و راست را نگه می‌دارد.

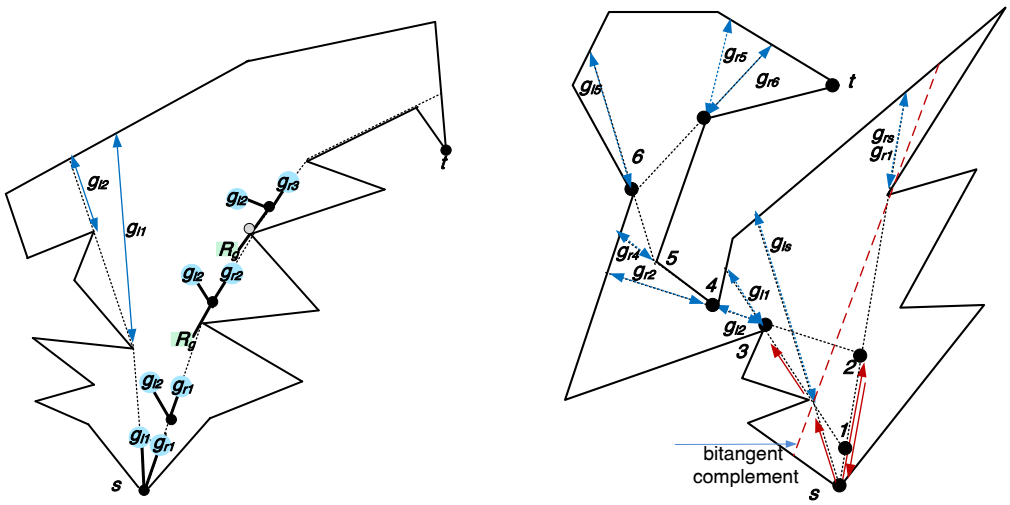
اثبات. از آنجایی که راس‌های متناظر با شکاف‌های پیشرفته همان راس‌های زنجیره‌های محدب هستند، حرکت به سمت این شکاف‌ها منجر به حرکت روی این دو زنجیر می‌شود. ■

[۶] Doubling Strategy

حرکت به سمت شکاف‌ها باعث تغییر موقعیت شکاف‌های پیشرفته می‌شود، در بخش ساختار داده چگونگی این تغییرات بیان شده است. علاوه بر این، یک ساختار داده که اطلاعات مورد نیاز برای بازگشت به مبدا در وضعیت قیف را ذخیره می‌کند، ارائه شده است.

۱.۲.۴ ساختار داده

در حین حرکت رو به جلوی ربات در امتداد هر یک از دو زنجیره چپ و راست، اطلاعات مربوط به مکان پیشرفته‌ترین شکاف‌ها (g_r و g_l) نگهداری می‌شود. با حرکت ربات در محدوده خیابان تغییرات همبندی[۷] در وضعیت شکاف‌ها ایجاد می‌شود که برخی از این تغییرات باعث تغییر در سمت راست‌ترین شکاف چپ و سمت چپ‌ترین شکاف راست می‌شود. همان‌طور که در بخش قبل توضیح داده شد، به طور کلی رخدادهایی که باعث تغییر همبندی شکاف‌ها می‌شوند، رخدادهای ظهور، ناپدیدی، ادغام و تقسیم هستند. این رخدادها همان‌طور که در ادامه شرح داده می‌شوند، در حالت‌های زیر می‌توانند باعث تغییر وضعیت دو شکاف g_r و g_l شوند. ■



(ب) (آ)

شکل ۳.۴: (آ) در نقاط ۱ تا ۶ رخدادهایی اتفاق می‌افتد که باعث تغییر پیشرفته‌ترین شکاف‌ها می‌شود. (ب) خطوط نقطه‌چین زنجیر محدب راست و چپ را نشان می‌دهد. نمایش چگونگی ساخته شدن و به‌روز شدن S-GNT وقتی ربات روی زنجیر راست حرکت می‌کند.

[۷] Topologic

- وقتی که ربات از روی مماس مشترک دو نقطه که هر دو بر روی یک زنجیر قرار دارند، عبور می‌کند یک رخداد تقسیم رخ می‌دهد. این رخداد باعث تغییر پیشرفته‌ترین شکافی که روی آن زنجیر قرار دارد، می‌شود (نقطه ۱ در شکل ۳.۴آ).
- وقتی که ربات از روی مماس مشترک دو راس عبور کند که یکی بر روی یک زنجیر قرار دارد و دیگری بر روی زنجیر دیگر، در این صورت پیشرفته‌ترین شکافی که ربات در حال حرکت به سمت آن است، عوض می‌شود (نقطه ۲ در شکل ۳.۴آ).
- وقتی که ربات از روی مماس مشترک شکاف پیشرفته‌ای که به سمت آن در حال حرکت است و یک شکاف دیگر از همین زنجیر عبور می‌کند (درلحظه ناپدید شدن این شکاف پیشرفته)، شکاف جدید جای شکاف پیشرفته را می‌گیرد (نقطه ۵ در شکل ۳.۴آ).
- زمانی که ربات از روی امتداد خطوط مجاور به راس‌های انعکاسی[۸] می‌گذرد، شکافی که در پشت راس انعکاسی است، ناپدید می‌شود. این رخداد را که ناپدیدی می‌نامیم باعث حذف پیشرفته‌ترین شکاف مربوطه می‌شود (نقطه ۳ در شکل ۳.۴آ).
- هنگامی‌که ربات از روی مماس مشترک دو راس که یکی بر روی زنجیر چپ و دیگری بر روی زنجیر راست قرار دارد، در لحظه ناپدید شدن شکافی که به سمت آن در حال حرکت است، عبور کند باز هم یک رخداد تقسیم اتفاق می‌افتد که باعث تغییر پیشرفته‌ترین شکاف زنجیر مقابل می‌شود (نقطه ۵ در شکل ۳.۴آ).

در لم بعدی نشان می‌دهیم از بین چهار رخداد اصلی که باعث تغییر موقعیت شکاف‌ها می‌شود فقط دو رخداد تقسیم و ناپدیدی، موقعیت شکاف‌های پیشرفته را عوض می‌کند که در ساختار داده در نظر گرفته شده است.

لم ۶.۴ فقط دو نوع از رخدادهای بحرانی باعث تغییر موقعیت پیشرفته‌ترین شکاف راست و پیشرفته‌ترین شکاف چپ می‌شوند: رخدادهای ناپدید شدن شکاف‌ها و رخداد تقسیم.

اثبات. در هنگام حرکت ربات هر رخداد ظهور شکاف باعث ایجاد شکاف‌هایی می‌شود که ربات قبلا محدوده پشت آن‌ها را دیده است، بنابراین این رخداد نمی‌تواند موقعیت شکاف‌های پیشرفته را که

[۸] Reflex Vertices

احتمال حضور هدف پشت آن‌ها است، تغییر دهد. هر رخداد ادغام شکاف‌های پیشرفته زمانی اتفاق می‌افتد که ربات از روی امتداد مماس مشترک[۹] شکاف پیشرفته و یک شکاف دیگر عبور کند. بنابراین ربات باید به ناحیه سمت راست پشرفته‌ترین شکاف راست یا ناحیه سمت چپ پیشرفته‌ترین شکاف چپ وارد شود. در الگوریتمی که برای حرکت برخط ربات ارائه کرده‌ایم چنین اتفاقی نمی‌افتد (شکل ۳.۴آ). بنابراین دو دسته رخداد ناپدید شدن شکاف‌ها و تقسیم تنها رخدادهایی هستند که اتفاق می‌افتند.

■

در وضعیت قیف، ربات از یک نشانه‌گذار برای علامت زدن این نقطه به عنوان مبدا، در راهبرد جستجوی دو گانه استفاده می‌کند. برای اجرای راهبرد جستجوی دوگانه، ربات وقتی روی یکی از زنجیرهای محدب حرکت می‌کند باید اطلاعات مورد نیاز برای بازگشت به مبدا را نیز ذخیره کند. این اطلاعات را در یک ساختار داده درختی به نام S-GNT[۱۰] ذخیره می‌کنیم. در ابتدا که ربات در وضعیت قیف قرار می‌گیرد، مکان جاری ربات، ریشه درخت و دو شکاف پیشرفته، فرزندان ریشه هستند. ریشه درخت در امتداد حرکت ربات جابجا می‌شود و شکاف‌های پیشرفته چپ و راست همانطور که قبلا توضیح داده شده به روز می‌شوند و مسیر بازگشت به مبدا مطابق آنچه در ادامه آمده است ساخته می‌شود. (فرض کنید ربات روی زنجیر راست حرکت می‌کند.)

- وقتی که ربات از روی یک شعاع انعکاسی عبور می‌کند یک شکاف ایجاد می‌شود. چنانچه این شکاف باعث مخفی شدن نشانه‌گذار شود یک فرزند به درخت اضافه می‌شود که موقعیت این شکاف را نگه می‌دارد. این شکاف را شکاف بازگشت به مبدا می‌نامیم (اولین راس از زنجیر محدب راست در شکل ۳.۴ب).
- وقتی که ربات از روی مماس مشترک مربوط به شکاف بازگشت و شکاف پیشرفته فعلی عبور می‌کند این دو شکاف ادغام می‌شوند. بنابراین شکاف بازگشت فرزندی از یک گره جدید شده که به درخت اضافه می‌شود (دومین راس از زنجیر محدب راست در شکل ۳.۴ب).

در شکل ۳.۴ب فرایند ساخت درخت وقتی که ربات به سمت شکاف پیشرفته راست حرکت می‌کند، نمایش داده شده است.

[۹] Bitangent Complement

[۱۰] Simple Gap Navigation Tree

۲.۲.۴ الگوریتم

ربات شروع به پیمایش خیابان بر اساس اطلاعات به‌دست آمده در مورد شکاف‌ها می‌کند. در هر نقطه از مسیر حرکت اگر فقط یکی از شکاف‌ها وجود داشته باشد، هدف پشت این شکاف است و ربات بدون هیچ ابهامی در حالی که موقعیت شکاف‌ها را به روز می‌کند، به سمت آن حرکت می‌کند (شکل ۲.۴آ). اگر هر دو شکاف پیشرفته وجود داشته باشند به عبارت دیگر ربات در وضعیت قیف قرار بگیرد، این نقطه را با نشانه‌گذار علامت‌گذاری می‌کند. سپس در امتداد هر یک از دو زنجیر محدب چپ و راست به سمت جلو و عقب حرکت می‌کند. در هر مرحله i ربات ۲^i گام در امتداد یکی از زنجیرها حرکت می‌کند در حالی‌که S-GNT را می‌سازد. سپس با استفاده از S-GNT به مبدا برمی‌گردد و $۲^{i+۱}$ در امتداد زنجیر مخالف حرکت می‌کند. این فرایند آن‌قدر ادامه می‌یابد تا این‌که به نقطه بحرانی آن قیف برسد. بعد از پیدا شدن نقطه بحرانی، زنجیر دقیق[۱۱] مشخص می‌شود. ربات در امتداد مسیر ضبط شده در آخرین S-GNT به مبدا برمی‌گردد و پس از برداشتن نشانه‌گذار در امتداد زنجیر دقیق حرکت می‌کند. ربات به حرکتش در حالی‌که موقعیت شکافهای پیشرفته را به روز می‌کند، ادامه می‌دهد تا اینکه یک نقطه قیف جدید به وجود بیاید که در این حالت فرایند مربوط به قیف تکرار می‌شود تا نقطه‌ی هدف را ببیند.

۳.۲.۴ تحلیل الگوریتم

در این قسمت نشان می‌دهیم که راهبرد ما برای هدایت ربات، آن را در زمان متناهی به هدف می‌رساند. علاوه بر این، طول مسیری که ربات برای رسیدن به هدف طی می‌کند با طول کوتاه‌ترین مسیر مقایسه می‌کنیم.

قضیه ۷.۴ راهبرد ما برای هدایت ربات پایان‌پذیر است و ربات را در انتها به نقطه هدفش می‌رساند.

اثبات. نکته کلیدی این است که هدف همواره پشت یکی از شکاف‌های پیشرفته پنهان است. با حرکت ربات به سمت این شکاف‌ها ممکن است یک شکاف پیشرفته جدید به وجود بیاورد یا آن‌ها را به‌روز کند ولی تعداد چنین رخدادهایی متناهی است. هر کدام از این رخدادها متناسب با عبور از روی یک مماس مشترک است. هر بار که با حرکت به سمت شکاف‌های پیشرفته یکی از آن‌ها ناپدید می‌شود،

[۱۱] Exact Chain

یک گام به نقطه هدف نزدیک‌تر می‌شویم. ■

برای مقایسه طول مسیر تولید شده توسط الگوریتم ما با طول کوتاه‌ترین مسیر، لم زیر را ثابت می‌کنیم.

لم ۸.۴ اگر از مسیر تولید شده توسط الگوریتم ما، مسیر رفت و برگشتی در هر وضعیت قیف برای پیدا کردن نقطه بحرانی را حذف کنیم، مسیر باقی‌مانده بر کوتاه‌ترین مسیر منطبق است.

اثبات. تنها در وضعیت قیف است که ربات نمی‌داند به کدام سمت حرکت کند. در چنین وضعیتی، ربات بعد از رسیدن به نقطه بحرانی به نقطه شروع قیف برمی‌گردد، با این تفاوت که الان می‌داند که هدف پشت کدام شکاف پیشرفته پنهان شده است. بنابراین حذف این مسیر رفت و برگشتی از مسیر تولید شده، باعث می‌شود مسیر تولید شده به کوتاه‌ترین مسیر تبدیل شود. ■

قضیه ۹.۴ راهبرد ما برای هدایت ربات برای جستجو در خیابان دارای ضریب رقابتی ۱۱ است و اگر اجازه دهیم تعداد دلخواهی نشانه‌گذار همراه داشته باشد به ضریب رقابتی ۹ می‌رسیم.

اثبات.

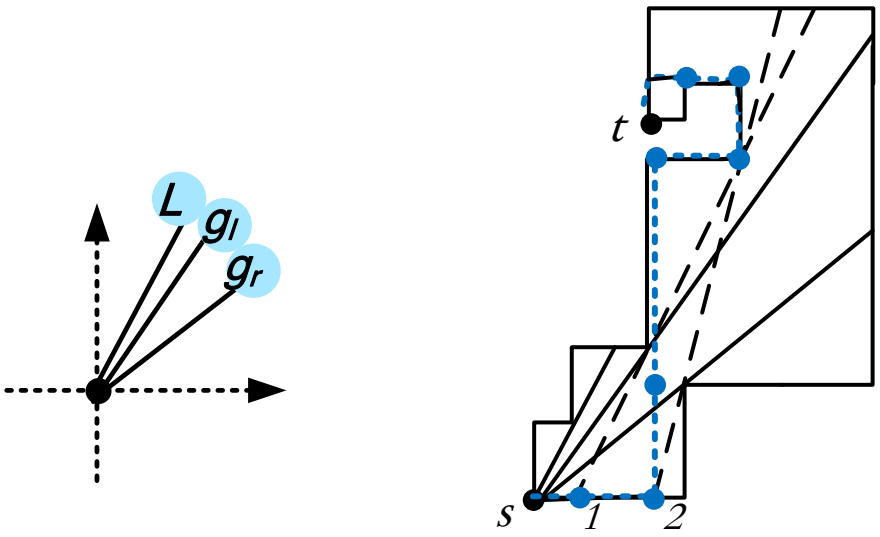
با توجه به اینکه فقط در وضعیت قیف کوتاه‌ترین مسیر و مسیر تولید شده مطابقت ندارند، می‌توانیم برای محاسبه ضریب رقابتی، این دو مسیر را فقط در یک وضعیت قیف با یکدیگر مقایسه کنیم. با توجه به اینکه ما راهبرد جستجوی دوگانه را در وضعیت قیف روی زنجیره‌های چپ و راست اعمال می‌کنیم، ربات پس از طی مسافتی با اندازه حداکثر ۹ برابر کوتاه‌ترین مسیر، به نقطه بحرانی می‌رسد. بعد از پیدا کردن نقطه بحرانی، ربات برای برداشتن نشانه‌گذار به مبدا بر می‌گردد و در امتداد مسیر صحیح حرکت می‌کند. پس مسیری که طی می‌کند حداکثر به اندازه ۱۱ برابر طول کوتاه‌ترین مسیر است. ■

۳.۴ مسیریابی در یک خیابان متعامد در امتداد یک مسیر متعامد

اکنون حالت خاصی از مسئله را طرح می‌کنیم. ربات مجهز به حسگر شکاف قادر است روی یک مسیر متعامد در یک خیابان که دارای اضلاع متعامد است، از نقطه s شروع به حرکت کرده و به نقطه t برسد. بدون از دست رفتن کلیت فرض می‌شود که خیابان دارای خطوط افقی-عمودی یا موازی با محورهای مختصات بوده و ربات قادر است روی یک مسیر افقی-عمودی حرکت کند (شکل ۴.۴آ). ربات در

حالی که وضعیت شکاف‌ها را درک می‌کند، این داده‌ها را در یک ساختار داده پویا ذخیره می‌کند (مانند S-GNT). ریشه این ساختار داده درختی، مکان جاری ربات بوده و فرزندان آن شکاف‌های موجود (به ترتیب دایره‌ای که در محیط قرار دارند) است (شکل ۴.۴).

■



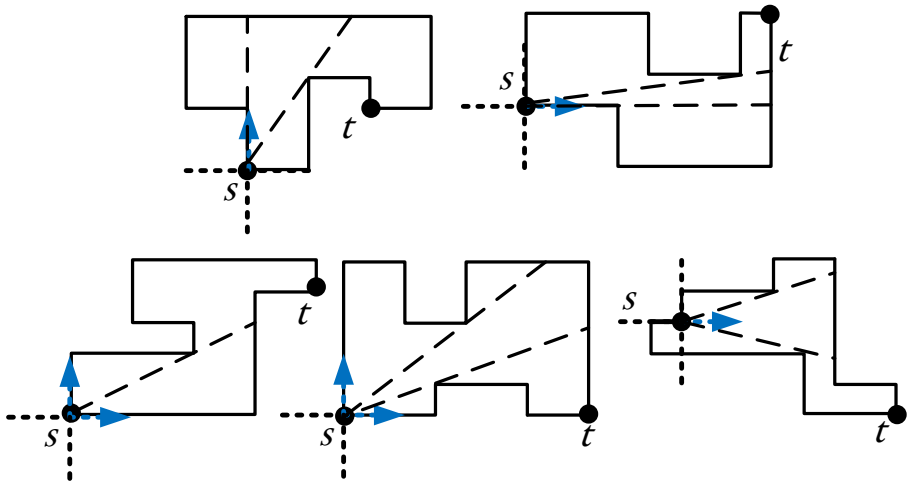
(ب) (آ)

شکل ۴.۴: (آ) یک خیابان متعامد. مسیر نقطه‌چین، مسیر تولید شده توسط استراتژی ما است. (ب) نمایش ساختار داده در نقطه شروع.

در طول مسیر جستجو دو حالت وجود دارد: در حالت اول هنوز نقطه هدف در ناحیه دید ربات قرار ندارد، پس نقطه هدف باید پشت یکی از شکاف‌های پیشرفته مخفی باشد. حالت دوم نقطه هدف در ناحیه دید ربات قرار دارد. وضعیت شکاف‌های پیشرفته در حالت اول، همواره به یکی از دو صورت زیر است:

- هر دو شکاف پیشرفته در یک ربع از دستگاه مختصات قرار دارند یا این‌که فقط یکی از شکاف‌های پیشرفته موجود است. اگر شکاف پیشرفته افقی یا عمودی موجود باشد، ربات به سمت این شکاف حرکت می‌کند. در غیر این صورت به سمت یکی از محورهای مختصات در برگیرنده این ربع حرکت می‌کند (شکل ۵.۴).
- این شکاف‌های پیشرفته در دو ربع متوالی قرار دارند. در این حالت ربات در امتدادی از دستگاه

مختصات که بین این دو شکاف قرار دارد، حرکت می‌کند (شکل ۵.۴).



شکل ۵.۴: مکان‌های ممکن برای شکاف‌ها در نقطه شروع حرکت. فلش‌ها جهتی را که ربات می‌تواند برای حرکت انتخاب کند، نشان می‌دهند.

راهبرد چرخش ربات وقتی که به سمت یک شکاف پیشرفته حرکت می‌کند به شرح زیر است.

در حین حرکت، رخدادهای بحرانی باعث تغییر وضعیت شکاف‌های پیشرفته می‌شود (همانطور که در بخش قبل توضیح داده شد). به محض اینکه این شکاف‌ها به‌روز می‌شوند، ربات بر اساس اینکه وضعیت جاری شکاف‌های پیشرفته کدام یک از دو حالت فوق را دارد، جهت حرکتش را انتخاب می‌کند. لازم به ذکر است که هر رخداد پدیدار شدن شکاف، مربوط به مخفی شدن قسمتی از خیابان است که از قبل رویت‌پذیر بوده است و یک گره که علامت بدوی می‌خورد به درخت اضافه می‌کند. برخلاف راهبرد هدایت ربات در خیابان معمولی، در این خیابان رخداد ادغام نیز باعث تغییر وضعیت شکاف‌های پیشرفته می‌شود. به محض این‌که چنین رخدادی اتفاق افتاد، ربات می‌چرخد و جهتی را که عمود بر جهت حرکت کنونی است، ادامه می‌دهد. علاوه بر این هر وقت ربات در حین حرکتش به یک دیوار عمودی برخورد کرد، بر اساس موقعیت شکاف‌های جاری جهت چرخشش را انتخاب می‌کند.

ربات با استفاده از الگوریتم فوق به جستجو ادامه می‌دهد تا به نقطه هدف برسد، یا به نقطه‌ای برسد که هدف از آن جا رویت‌پذیر باشد.

برهانی مشابه آنچه درستی قضیه ۷.۴ را ثابت کرد، اثبات می‌کند که ربات در زمان متناهی به نقطه هدف می‌رسد. در قضیه زیر طول مسیر تولید شده توسط این راهبرد و طول کوتاه‌ترین مسیر افقی و عمودی موجود بین دو نقطه s و t در خیابان با هم مقایسه می‌شوند.

قضیه ۱۰.۴ [۵۵] مسیر تولید شده توسط راهبرد ما یک L_1-کوتاه‌ترین مسیر از s به t است. همچنین

این راهبرد دارای ضریب رقابتی $\sqrt{۲}$ در فضای متریک اقلیدسی ($L_۲$-متریک) است.

اثبات. در حین جستجو، ربات بر اساس موقعیت شکاف‌های پیشرفته یک جهت افقی یا عمودی را برای حرکت انتخاب می‌کند. فرض کنید که (x_l, y_l) و (x_r, y_r) راس‌های انعکاسی مربوط به g_l و g_r هستند. اگر ربات یک جهت افقی را برای حرکت انتخاب کند، تا نقطه p که در آن $|x_s - x_p| = |x_s - x_l|$ یا $|x_s - x_p| = |x_s - x_r|$ به حرکت ادامه می‌دهد. به طور مشابه اگر ربات یک جهت عمودی را برای حرکت انتخاب کند، تا نقطه p که در آن $|y_s - y_p| = |y_s - y_l|$ یا $|y_s - y_p| = |y_s - y_r|$ به حرکت ادامه می‌دهد. به این دلیل که در آن نقطه یک رخداد بحرانی اتفاق می‌افتد (شکل ۴.۴آ). هر مسیر متعامد از s برای پوشاندن نقاط مخفی پشت شکاف‌ها در امتداد محورهای X و Y به اندازه مسیر ما ادامه می‌یابد. این نشان می‌دهد که نقطه p روی $L_۱$-کوتاه‌ترین مسیر از s به t است. از آنجایی که هر $L_۱$-کوتاه‌ترین مسیر حداکثر $\sqrt{۲}$ برابر $L_۲$-کوتاه‌ترین مسیر متناظر با آن است، ضریب رقابتی برای الگوریتم ما در فضای اقلیدسی $\sqrt{۲}$ است. ■

۴.۴ راهبرد تصادفی برای هدایت ربات

اکنون الگوریتم تصادفی خود را برای هدایت ربات ارائه می‌دهیم. تا زمانی که نقطه هدف در ناحیه دید ربات قرار نگرفته است ربات مجموعه‌ای از شکاف‌های چپ و راست را گزارش می‌دهد. موقعیت این شکاف‌ها در طول مسیر تغییر می‌کند. مکان شکاف‌های اساسی در ساختار داده S-GNT همان طور که چگونگی ساخت و به‌روز شدن آن شرح داده شد، ذخیره می‌شود. در هر نقطه از مسیر حرکت دوحالت اتفاق می‌افتد.

- زمانی که ربات در مکانی قرار دارد که فقط یکی از این شکاف‌های پیشرفته وجود دارند، بدیهی است که هر راهبرد منطقی ربات را به سمت این شکاف هدایت می‌کند.
- در حالتی که هر دو شکاف پیشرفته وجود دارند، ربات از یک نشانه‌گذار استفاده می‌کند برای اینکه بعدا بتواند این نقطه را به یاد آورد. از آنجائی که ربات نمی‌داند هدف پشت کدام یک از شکاف‌های g_r یا g_r قرار دارد، از یک متغیر تصادفی از مجموعه $\{۰, ۱\}$ استفاده می‌کند. چنانچه مقدار این متغیر تصادفی ۱ (۰) باشد، ربات در هر مرحله i به اندازه ۲^i گام در امتداد زنجیر راست

(چپ) حرکت می‌کند سپس به مبدا (مکانی که نشانه‌گذار را قرار داده بود) برمی‌گردد و به اندازه $۲^{i+۱}$ گام در امتداد زنجیر چپ (راست) حرکت می‌کند و دوباره به مبدا حرکت برمی‌گردد. این حرکت رفت و برگشت در امتداد دو زنجیر محدب چپ و راست ادامه پیدا می‌کند تا ربات به نقطه بحرانی قیف برسد. نقطه بحرانی بر اساس اطلاعات موجود در ساختار داده S-GNT قابل تشخیص است. بعد از به دست آمدن نقطه بحرانی، ربات برای برای برداشتن نشانه‌گذار به مبدا برمی‌گردد و در امتداد زنجیر درست حرکت می‌کند. ربات به حرکت در امتداد زنجیر درست ادامه می‌دهد تا به هدف برسد یا اینکه به یک نقطه قیف جدید برسد. در وضعیت قیف دوباره حرکت رفت و برگشت در امتداد زنجیر چپ و راست را با استفاده از متغیر تصادفی اجرا می‌کند. لازم به ذکر است که طول گام هم به صورت یک عدد تصادفی انتخاب می‌شود.

۱.۴.۴ تحلیل الگوریتم

در این قسمت نشان می‌دهیم که راهبرد ما ربات را در امتداد مسیری که طول آن ضریب ثابتی از طول کوتاه‌ترین مسیر است به هدف می‌رساند. به عبارت دیگر نسبت حداکثر مسافت مورد انتظار پیموده شده توسط ربات به طول کوتاه‌ترین مسیر برای رسیدن به هدف مقدار ثابتی است.

در حالت اول که ربات فقط یکی از زنجیرهای چپ و راست را پیش رو دارد، مسیری که ربات طی می‌کند همان کوتاه‌ترین مسیر است. برای پیدا کردن ضریب رقابتی باید مسیری که ربات طی می‌کند را با کوتاه‌ترین مسیر در وضعیت قیف مقایسه کنیم. چون بدترین حالت زمانی رخ می دهد که ربات همواره در وضعیت قیف قرار بگیرد. کافی است ضریب رقابتی را برای یک قیف به دست آوریم. این ضریب رقابتی، ضریب مورد انتظار برای کل مسیر حرکت ربات است. با استفاده از ساختار داده S-GNT می‌توانیم ربات را در امتداد زنجیر چپ و راست حرکت دهیم بدون اینکه به مکان دیگری منحرف شود. اگر این دو زنجیر چپ و راست را به عنوان یک خط در نظر بگیریم، با شروع از مبدا، نقطه‌ی هدف سمت چپ یا راست این خط قرار دارد. اثباتی شبیه به اثبات الگوریم گاو-سریع [۱۲] نشان می‌دهد که این ربات می‌تواند بعد از طی مسافتی حداکثر ۴/۵۹ برابر طول کوتاه‌ترین مسیر به نقطه بحرانی این قیف برسد. ربات پس از رسیدن به نقطه بحرانی به مبدا برمی‌گردد و در امتداد زنجیر صحیح حرکت می‌کند. بنابراین ربات در هر وضعیت قیف، برای رسیدن به نقطه‌ای از کوتاه‌ترین مسیر، حداکثر ۶/۵۹ برابر طول کوتاه‌ترین مسیر را طی می‌کند. این مطالب اثبات کننده قضیه زیر است.

[۱۲] Smartcow Algorithm

قضیه ۱۱.۴ راهبرد تصادفی ارائه مسیری برای رسیدن ربات به نقطه هدف t با شروع از نقطه s در خیابان تولید می‌کند. ضریب رقابتی مورد انتظار این مسیر **۶/۵۹** است.

۵.۴ همکاری ربات‌ها

اکنون می‌خواهیم الگوریتمی برای همکاری دو ربات با حسگر محدود، جهت رسیدن به هدف از یک نقطه شروع ارائه دهیم. ربات‌ها می‌توانند به هم کمک کرده و مسیر کوتاه‌تری برای رسیدن به هدف، از حالتی که یک ربات هدف را جستجو می‌کند، طی کنند. ربات‌ها را به یک وسیله ارتباطی مجهز کرده‌ایم تا بتوانند با یکدیگر در ارتباط بوده و اطلاعاتی را برای یکدیگر ارسال کنند. نوع وسیله ارتباطی بسته به میزان وسعت فضای کاری می‌تواند متفاوت باشد؛ به عبارتی دیگر، نوع وسیله ارتباطی این ربات‌ها با توجه به اینکه در چه گستره‌ای می‌خواهند اجرا شوند متفاوت خواهد بود. در اینجا، تمرکز ما بر روی نحوه حرکت ربات و راهبرد حرکتی ربات می‌باشد و به روش‌های ارتباط ربات‌ها و مسائل مربوط به آن (مانند خطا در ارسال اطلاعات، نحوه همگام[۱۳] شدن ربات‌ها، نحوه ارتباط با یکدیگر، پروتکل‌های مورد استفاده و...) نمی‌پردازیم؛ به عبارتی دیگر، فرض می‌کنیم که ربات‌ها یک درگاه ارتباطی پایدار[۱۴]، ایمن[۱۵] و مطمئن جهت ارتباط با یکدیگر دارند.

لازم به ذکر است که این مسئله، به صورت یک کار مشترک، در قالب یک پایان‌نامه کارشناسی ارشد نیز در دانشکده ارائه شده است [۱، ۸۰].

۱.۵.۴ الگوریتم

هر دو ربات از نقطه s شکاف‌های پیشرفته خود را شناسایی می‌کنند. در حالتی که فقط یک شکاف پیشرفته وجود داشته باشد، ربات‌ها بدون هر گونه ابهامی به سمت آن حرکت می‌کنند. در حالتی که هر دو شکاف فوق (وضعیت قیف) وجود دارد، ربات چپ، به سمت شکاف پیشرفته چپ و ربات راست به سمت شکاف پیشرفته راست حرکت می‌کند. هر ربات در هر لحظه فقط دو شکاف پیشرفته خود را ذخیره، نگهداری و به‌روز می‌کند و با توجه به رویدادهایی که برای این شکاف‌ها اتفاق می‌افتد واکنش

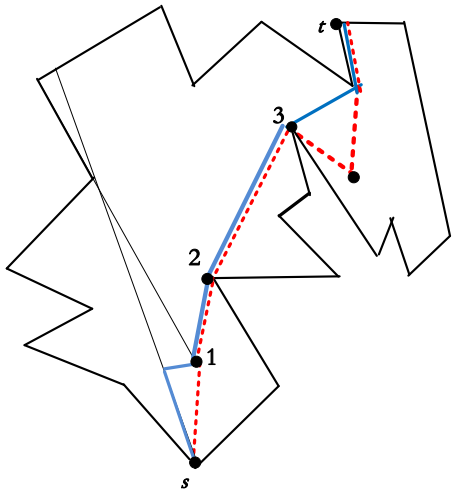
[۱۳] Synchronous

[۱۴] Stable

[۱۵] Safe

مناسب را انجام خواهد داد؛ به عبارتی دیگر، ربات‌ها نیاز به نگهداری هیچ گونه ساختار داده‌ای ندارند و صرفا به $O(1)$ حافظه جهت نگهداری دو شکاف پیشرفته نیاز دارند. لازم به ذکر است رخدادهایی که باعث می‌شود ربات‌ها تغییر مسیر حرکت بدهند همان رخدادهایی است که ربات تنها جهت تغییر مسیر از آن‌ها استفاده می‌کند، بعلاوه‌ی پیام‌هایی که از ربات همکار دریافت می‌کند.

ربات‌ها به حرکت به سمت شکاف پیشرفته مربوط به خود ادامه می‌دهند تا اینکه یکی از آن‌ها به نقطه‌ای برسد که در آن وضعیت قیف خاتمه یافته یا یک قیف جدید ایجاد شود (نقطه ۱ در شکل ۶.۴). رباتی که به نقطه فوق رسیده است، به ربات همکارش پیام می‌دهد و آن را به سوی خود فرا می‌خواند. در آن نقطه مانند شروع هر کدام مسیری را برای حرکت انتخاب می‌کنند، شکل ۶.۴ را ببینید.



شکل ۶.۴: ربات سمت راست در امتداد مسیر نقطه‌چین حرکت می‌کند و ربات سمت چپ در امتداد مسیر ممتد حرکت می‌کند. همکاری دو ربات مانع از حرکت ربات چپ در مسیری اشتباه می‌شود.

۲.۵.۴ تحلیل الگوریتم

در این بخش به تحلیل الگوریتم جهت اثبات اینکه این الگوریتم دارای ضریب رقابتی دو است، می‌پردازیم. همان طور که قبلا بیان شد هدف از طراحی این الگوریتم کمینه کردن طول مسیر طی شده توسط دو ربات است.

لم ۱۲.۴ راهبرد ارائه شده در بخش ۲.۵.۴ در نهایت منجر به رسیدن یکی از ربات‌ها به نقطه هدف t می‌شود.

اثبات. با توجه به راهبرد و رویدادهای بحرانی که منجر به تغییر جهت ربات می‌شوند، واضح است که ربات راست (چپ) فقط در صورتی مسیر خود را تغییر می‌دهد که ربات به نقطه‌ای برسد که وضعیت قیف به پایان می‌رسد یا یک قیف جدید ایجاد می‌شود، یا پیامی از ربات چپ (راست) دریافت کند. از آنجا که رویدادهایی که برای ربات‌ها مهم هستند، رویدادهای تقسیم و ناپدید شدن می‌باشند، لذا ربات هیچگاه حرکت رو به عقب نخواهد داشت؛ یعنی ربات هیچ گاه به ناحیه‌ای که قبلا مشاهده کرده است باز نخواهد گشت. از آنجا که محیط مورد نظر ما یک چندضلعی خیابان است و حرکت ربات همیشه رو به جلو است (به سمت شکاف‌هایی که هنوز پشت آن‌ها را ندیده است)، لذا با توجه به راهبرد حرکتی ربات‌ها، حداقل یکی از آن‌ها به نقطه هدف t خواهد رسید. ■

لم زیر نشان می‌دهد که در هر موقعیت قیف حتما، یکی از ربات‌ها در امتداد کوتاه‌ترین مسیر حرکت می‌کند.

لم ۱۳.۴ در راهبرد حرکت ربات که در بخش ۲.۵.۴ بیان شد، حرکت ربات‌ها به گونه‌ای می‌باشد که در هر بخشی، در بدترین حالت حداقل یکی از ربات‌ها در امتداد کوتاه‌ترین مسیر بین نقطه شروع s و نقطه هدف t است.

اثبات. می‌دانیم که نقطه هدف t پشت یکی از شکاف‌های پیشرفته است. با توجه به راهبرد حرکت ربات‌ها، در صورتی که ربات‌ها در موقعیتی قرار داشته باشند که فقط یکی از آن دو وجود دارد، هر دو ربات در امتداد کوتاه‌ترین مسیر بین نقطه آغاز s و هدف t حرکت می‌کنند. در صورتی که هر دو شکاف پیشرفته وجود داشته باشند، یکی از ربات‌ها به سمت شکاف پیشرفته چپ و دیگری به سمت شکاف پیشرفته راست حرکت خواهد کرد؛ که در این حالت نیز یکی از ربات‌ها حتما در امتداد کوتاه‌ترین مسیر بین نقطه آغاز s و نقطه هدف t خواهد بود؛ لذا در هر حالتی حداقل یکی از ربات‌ها در امتداد کوتاه‌ترین مسیر خواهد بود و لم فوق برقرار است. ■

در زیر لمی را ارائه می‌کنیم که نشان می‌دهد میزان حرکت اشتباه هر ربات محدود است؛ یعنی هر ربات در نهایت قادر به حرکت در محیط به اندازه محدودی خواهد بود و همکاری بین ربات‌ها و راهبرد حرکتی ربات‌ها، به آن‌ها اجازه نمی‌دهد که مسیر اضافه زیادی را طی کنند.

لم ۱۴.۴ راهبرد ارائه شده در بخش ۲.۵.۴ به هیچ یک از ربات‌ها اجازه نمی‌دهد مسیر انحرافی بیش از سه برابر کوتاه‌ترین مسیر بین نقطه آغاز s و نقطه هدف t را طی کنند.

اثبات. در صورتی که ربات‌ها در موقعیتی قرار داشته باشند که فقط یک شکاف پیشرفته وجود داشته باشد، هر دو ربات در امتداد کوتاه‌ترین مسیر بین نقطه آغاز s و نقطه هدف t قرار خواهند داشت. در

صورتی که ربات‌ها در موقعیت قیف قرار داشته باشند، یکی از ربات‌ها در امتداد کوتاه‌ترین مسیر قرار خواهد داشت، و ربات دیگر ممکن است مسیر انحرافی را طی کرده و پس از طی کردن مسافتی محدود پی به مسیر اشتباه خود برده و مسیر خود را اصلاح کند. بدون از دست دادن کلیت مسئله، فرض می‌کنیم ربات راست در امتداد کوتاه‌ترین مسیر حرکت می‌کند و ربات چپ در حال حرکت به سمت شکاف چپ می‌باشد که جهت نادرستی است. شکل ۶.۴ این موضوع را نشان می‌دهد. در این صورت ربات راست پس از طی کردن مسافتی به طول a از نقطه آغاز، متوجه می‌شود که ربات چپ در مسیر اشتباهی در حال حرکت است؛ لذا پیام «مسیر اشتباه» را به ربات چپ ارسال می‌کند. تا اینجا ربات چپ مسیری به طول a را در جهت اشتباه حرکت کرده است. ربات چپ با دریافت پیام «مسیر اشتباه» جهت حرکت خود را اصلاح کرده و به سمت شکاف g_r حرکت می‌کند؛ در بدترین حالت ربات چپ مجبور است به نقطه آغاز بازگشته و به سمت شکاف راست حرکت کند، یعنی مسیری به طول a را طی کرده و به نقطه آغاز باز گردد؛ سپس به سمت شکاف راست حرکت کند که این مسیر نیز دارای طول a است. پس در این حالت ربات چپ حداکثر مسیری به طول ۳a طی می‌کند تا به نقطه‌ای برسد که از نقطه آغاز فاصله‌ای به اندازه a دارد. همین استدلال برای حالتی که ربات چپ در جهت صحیح حرکت می‌کند و ربات راست در حال حرکت در جهت مسیر انحرافی است نیز صادق است. ■

در زیر قضیه‌ای را اثبات می‌کنیم که نشان می‌دهد ضریب رقابتی برای مسیر طی شده توسط دو ربات حداکثر دو برابر طول کوتاه‌ترین مسیر بین نقطه آغاز s و نقطه هدف t می‌باشد.

قضیه ۱۵.۴ الگوریتم ارائه شده برای حالتی که هدف کمینه کردن طول کوچکترین مسیر حرکت دو ربات باشد، دارای ضریب رقابتی دو است.

اثبات. جهت اثبات فرض می‌کنیم طول کوتاه‌ترین مسیر بین نقطه آغاز s و نقطه هدف t برابر با α باشد. جهت افزایش کمترین مسیر بین دو ربات، طول مسیری که هر دو ربات طی می‌کنند را برابر در نظر می‌گیریم. حال جهت اینکه طول دو مسیر را حداکثر کنیم، فرض می‌کنیم نیمی از مسیر را ربات چپ بر روی کوتاه‌ترین مسیر بوده است یعنی $\frac{\alpha}{۲}$ و نیمی از مسیر را اشتباه رفته است، که با توجه به لم ۱۴.۴ حداکثر مسیری که طی می‌کند برابر با $\frac{۳\alpha}{۲}$ می‌باشد. یعنی ربات چپ مسیری به طول $۲\alpha = \frac{\alpha}{۲} + \frac{۳\alpha}{۲}$ را طی می‌کند. ربات راست نیز مسافتی به همین اندازه را طی می‌کند و ضریب رقابتی دو می‌باشد. ■

۶.۴ جمع‌بندی

در این فصل مسئله هدایت ربات نقطه‌ای مجهز به حسگر شکاف مطرح شد با این فرض که عوامل محیطی مثل دما، رطوبت، نور و غیره تاثیری بر عملکرد ربات ندارند. ربات با سرعت ثابت در خیابان از نقطه شروع s شروع به حرکت می‌کند تا به هدفش در نقطه t برسد. یک راهبرد برخط برای حرکت این ربات ارائه شد که ربات با کمک گرفتن از یک نشانه‌گذار می‌تواند مسیری را که حداکثر ۱۱ برابر طول کوتاه‌ترین مسیر است، برای رسیدن به هدف طی می‌کند. بر اساس الگوریتم قطعی ارائه شده یک الگوریتم تصادفی ارائه دادیم. ضریب رقابتی مورد انتظار الگوریتم ۶/۵۹ است. این ضریب تقریبا دو برابر ضریب رقابتی الگوریتم قطعی را بهبود می‌دهد. همچنین الگوریتمی قطعی برای هدایت دو ربات برای رسیدن به یک هدف مشترک ارائه شده به طوری که مسیری که ربات‌ها طی می‌کنند از ۲ برابر طول کوتاه‌ترین مسیر کمتر است. همچنین الگوریتمی برای هدایت ربات‌ها ارائه شده به طوری که مسیری که ربات‌ها طی می‌کنند از ۲ برابر طول کوتاه‌ترین مسیر کمتر است.

فصل ۵

هدایت یک ربات ساده در خیابان در امتداد مسیری با کمترین تعداد چرخش

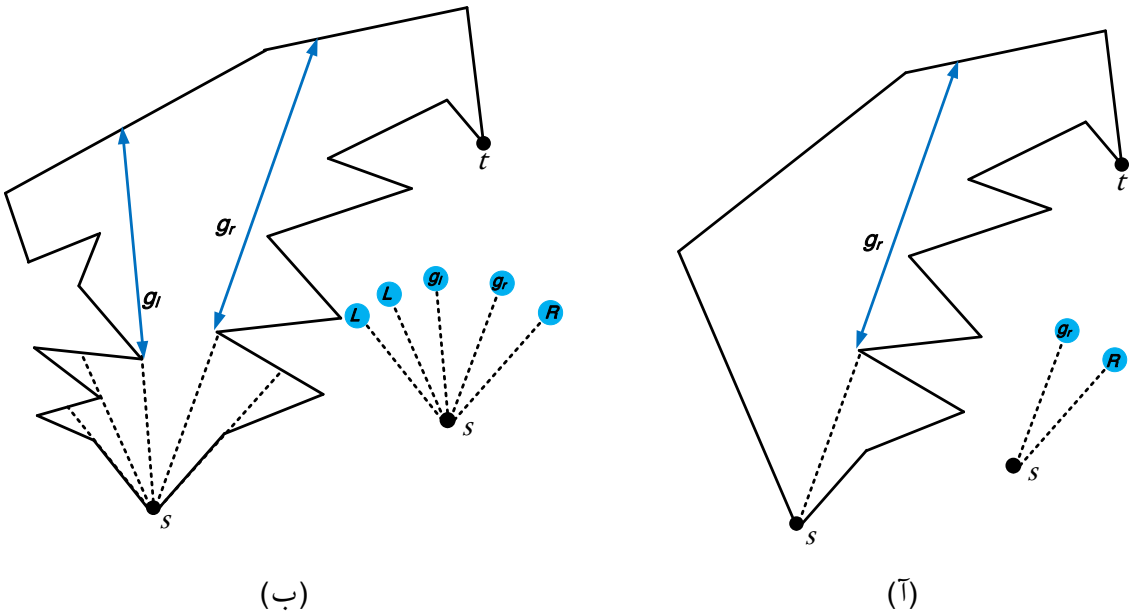
در این فصل نیز مسئله هدایت همان ربات ساده در یک خیابان را طرح می کنیم. تفاوت این ربات با ربات مطرح شده در فصل قبل این است که چرخش برای این ربات هزینه دارد در حالی حرکت در راستای مستقیم هزینه زیادی ندارد. به عبارت دیگر هدف این است که برای ربات ساده مجهز به حسگر شکاف یک راهبرد جستجو ارائه شود که ربات را از نقطه شروع s به نقطه هدف t برساند به طوری‌که مسیر تولید شده برای ربات کم‌ترین تعداد پیوند را داشته باشد. ثابت می‌شود، راهبرد ارائه شده در این فصل یک مسیر بهینه که دارای کمترین تعداد پیوند[۱] است تولید می‌کند. همچنین ثابت می‌کنیم ضریب رقابتی برای این مسیر $۲ - ۱/m$ است. در ضریب فوق m تعداد پیوندها در مسیر بهینه‌ای است که با در دست داشتن نقشه محیط به دست می‌آید. در انتهای فصل، برای حالت خاصی که چندضلعی متعامد است و ربات فقط قادر به حرکت روی یک مسیر افقی-عمودی است، نیز یک راهبرد رقابتی بهینه ارائه شده است.

۱.۵ مفاهیم مورد نیاز

ربات از نقطه s در خیابان ناشناخته شروع به حرکت می‌کند و با استفاده از اطلاعاتی که از طریق حسگرهایش به دست می‌آورد، می‌خواهد با طی کردن مسیری با کمترین تعداد چرخش به نقطه هدف t

[۱] Link

برسد. شکل ۱.۸ را ببینید.



(ب) (آ)

شکل ۱.۵: خیابان و آنچه ربات از این محیط درک می‌کند. (آ) ربات فقط یک راه دارد و بدون ابهامی آن را طی می‌کند. (ب) ربات دو راه دارد که شکاف پیشرفته چپ را انتخاب می‌کند.

در نقطه شروع دو حالت پیش می‌آید: حالت اول این است که ربات نقطه هدف را می‌بیند، بدیهی است که در این حالت ربات به سمت هدف حرکت می‌کند. حالت دوم این است که هدف قابل دید نیست و ربات با مجموعه‌ای از شکاف‌ها با برچسب‌های L و R روبرو می‌شود. همانطور که در فصل قبل اشاره شد، ساختار خیابان به گونه‌ای است که هدف فقط می‌تواند پشت سمت چپ‌ترین شکاف راست (g_r) مخفی باشد، یا فقط می‌تواند پشت سمت راست‌ترین شکاف چپ (g_l) مخفی باشد [۴۵]. در حالتی که فقط یکی از دو شکاف پیشرفته فوق وجود دارد، هر راهبرد منطقی چنین رباتی را به سمت آن شکاف برای دیدن ناحیه مخفی پشت آن هدایت می‌کند (شکل ۱.۵آ). حالتی که هر دو شکاف g_l و g_r وجود دارند، وضعیت قیف نامیده می‌شود. ربات نمی داند که هدف پشت کدام یک از دو شکاف پیشرفته چپ یا راست قرار دارد (شکل ۱.۵ب).

از آنجایی که در کل مسیر حرکت تا رسیدن به هدف، همواره هدف پشت یکی از این دو شکاف قرار دارد مکان این دو شکاف در داخل ساختار درختی که به صورت پویا تغییر می‌کند، نگهداری می‌شود. این درخت فقط دارای دو شاخه است که در هر نقطه از مسیر حرکت، این دو شاخه، آدرس دو شکاف پیشرفته چپ و راست فعلی را نگه می‌دارند. مکان این دو شکاف (ساختار درخت) با حرکت ربات تغییر می‌کند. در نقطه شروع حرکت، نقطه s ریشه درخت و دو شکاف پیشرفته برگ‌های درخت هستند.

ریشه درخت همواره مکان فعلی ربات است و رخدادهای پدیدار شدن شکاف، تقسیم شکاف، ناپدید شدن شکاف و ادغام شکاف، برگ‌های این درخت را به صورت زیر تغییر می‌دهد.

- وقتی که ربات از روی مماس مشترک g_r (g_l) و یک شکاف راست (شکاف چپ) عبور کند، g_r (g_l) تقسیم می‌شود و با آن شکاف راست (شکاف چپ) جایگزین می‌شود. نقطه ۱ در شکل ۲.۵ را ببینید.

- وقتی که ربات از روی مماس مشترک g_r (g_l) و یک شکاف چپ (شکاف راست) عبور کند، g_r (g_l) تقسیم می‌شود و g_l (g_r)با آن شکاف چپ (شکاف راست) جایگزین می‌شود. نقطه ۲ در شکل ۲.۵ و نقطه ۲ در شکل ۳.۵آ را ببینید.

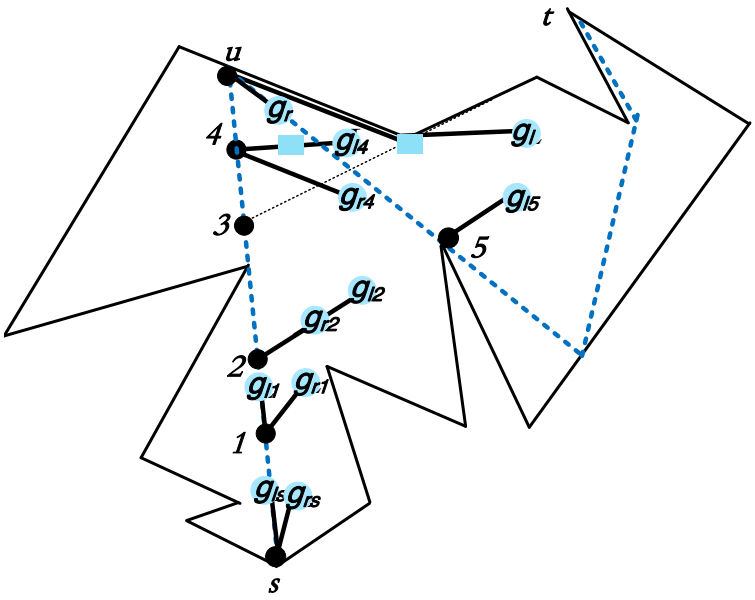
- هر رخداد پدیدار شدن یک شکاف (عبور از روی یک شعاع انعکاسی) باعث مخفی شدن ناحیه‌ای می‌شود که می‌شود که ربات قبلا آن ناحیه را می‌دیده است. چنین شکاف‌هایی نادیده گرفته می‌شود مگر اینکه این شکاف‌ها باعث مخفی شدن شکاف‌های پیشرفته فعلی شود یا به عبارت دیگر شکاف پیشرفته‌ای با آن‌ها ادغام شود که در این صورت شکاف پیشرفته به صورت فرزندی از این شکاف پدیدار شده در درخت در نظر گرفته می‌شود. نقطه ۱ در شکل ۲.۵ را ببینید.

- هرگاه ربات از روی شعاع انعکاسی یک شکاف پیشرفته عبور کند که باعث ناپدید شدن آن شود، آن شکاف پیشرفته از درخت حذف می‌شود. نقطه ۵ در شکل ۲.۵ و نقطه ۱ در شکل ۳.۵آ را ببینید.

در نقطه شروع، حرکت ربات به سمت هر یک از شکاف‌های پیشرفته چپ یا راست به گونه‌ای است که با ادامه حرکت ناحیه مخفی پشت این دو شکاف کاهش می‌یابد. ایده اصلی ما در اینجا حداکثر استفاده از یک راستای انتخاب شده است به عبارت دیگر تا وقتی که حرکت در امتداد راستای انتخاب شده باعث کاهش ناحیه مخفی پشت حداقل یکی از شکاف‌ها شود، ربات به حرکتش در این راستا ادامه می‌دهد. به همین دلیل شکاف فعال[۲] را تعریف کرده‌ایم.

تعریف ۱.۵ در طول مسیر حرکت ربات، یک شکاف پیشرفته فعال است، اگر ادامه حرکت در راستای فعلی باعث کاهش ناحیه مخفی پشت آن شکاف پیشرفته شود. در غیر اینصورت آن شکاف یک شکاف

[۲] Active gap



شکل ۲.۵: نمایش تغییرات ساختار داده‌ای که مکان g_l و g_r را نگه می‌دارد. در نقطه u زنجیر ساعتگرد از s به u به طور کامل از روی مسیر نقطه چین که s را به u وصل می‌کند، قابل دید است و شکاف پیشرفته چپ تا قبل از رسیدن به نقطه u در نقطه ۲ غیر فعال شده است. ربات در این نقطه به سمت شکاف پیشرفته راست می‌چرخد.

غیر فعال[۳] است.

در نقطه شروع حرکت ربات، هر یک از شکاف‌های پیشرفته موجود، با توجه به تعریف، فعال هستند. برخی از رخدادهای پدیدار شدن شکاف، تقسیم شکاف، ناپدید شدن شکاف و ادغام شکاف می‌تواند وضعیت شکاف‌های پیشرفته را از فعال به غیر فعال تغییر دهد. در ادامه توضیح می‌دهیم که در هر یک از وضعیت‌های قیف و غیر قیف چگونه وضعیت فعال بودن شکاف‌ها عوض می‌شود.

۱. در حالت قیف هر دو شکاف پیشرفته فعال هستند تا وقتی که این وضعیت به پایان برسد. همان‌طور که قبلا اشاره شده ربات وقتی با حالت قیف روبرو است که هر دو شکاف پیشرفته وجود داشته باشند. بنابراین این حالت زمانی به پایان می‌رسد که:

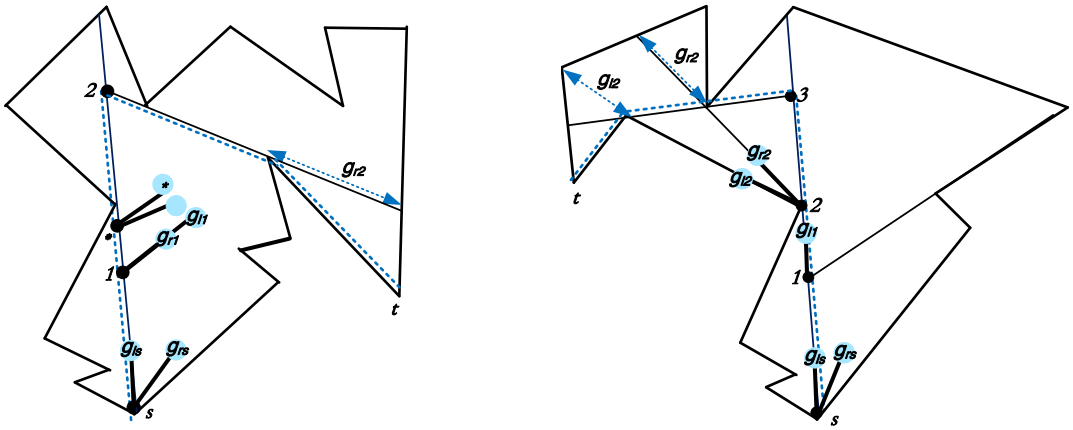
الف) دوشکاف در یک راستا قرار بگیرند. با ادامه حرکت یک وضعیت قیف جدید ایجاد می‌شود که یکی از شکاف‌های پیشرفته آن متعلق به قیف قبلی است که همچنان فعال باقی می‌ماند و دیگری شکاف جدید است که با تقسیم شکاف پیشرفته متعلق به قیف قبلی ایجاد شده است. در راستای این حرکت این شکاف جدید فعال نیست. نقطه * در شکل ۳.۵ب را ببینید.

ب) ربات به نقطه‌ای برسد که یکی از شکاف‌های پیشرفته ناپدید شود. در این وضعیت با ادامه

[۳] Inactive gap

حرکت در راستای کنونی شکاف پیشرفته کنونی فعال باقی می‌ماند. نقطه ۱ در شکل ۳.۵آ را ببینید.

۲. وقتی که رباتی در وضعیت غیر قیف قرار دارد (فقط یکی از شکاف‌های پیشرفته وجود دارد). فرض کنید فقط شکاف پیشرفته چپ وجود داشته باشد. در این حالت یک وضعیت قیف می‌تواند از طریق تقسیم شدن این شکاف پیشرفته چپ به دو شکاف چپ و راست به وجود بیاید. این دو شکاف با هم سازنده یک قیف جدید می‌شوند. واضح است، با ادامه حرکت در راستای قبلی شکاف پیشرفته چپی که از قبل موجود بود، فعال باقی می‌ماند ولی شکاف پیشرفته راست ایجاد شده فعال نیست. نقطه ۲ در شکل ۳.۵آ را ببینید. در حالتی که فقط شکاف پیشرفته راست موجود باشد هم به همین ترتیب وضعیت قیف می‌تواند ایجاد شود.



(ب) (آ)

شکل ۳.۵: مسیر نقطه‌چین از s به t مسیر حرکت ربات را نشان می‌دهد، در هر یک از نقاط تیره رنگ شکاف‌های پیشرفته تغییر می‌کنند که این تغییر باعث تغییر در وضعیت فعال بودن آن‌ها نیز می‌شود.

بنابراین هریک از رخدادهای پدیدار شدن شکاف، تقسیم شکاف، ناپدید شدن شکاف و ادغام شکاف همانطور که مکان شکاف‌های پیشرفته را تغییر می‌دهد، وضعیت فعال بودن یا فعال نبودن شکاف‌ها را می‌تواند عوض کند.

۲.۵ الگوریتم

در این بخش راهبردی برای هدایت ربات بر اساس اطلاعات به دست آمده از طریق حسگر شکاف ارائه می‌دهیم. اگر ربات در نقطه شروع حرکت، نقطه هدف را نبیند با مجموعه‌ای از شکاف‌های چپ و

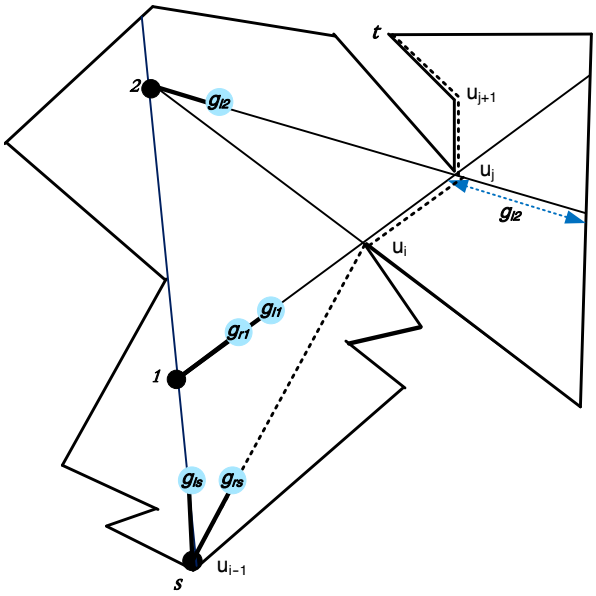
راست روبرو می‌شود. هدف پشت یکی از شکاف‌های پیشرفته چپ یا راست قرار دارد. در حالی که فقط یکی از شکاف‌ها وجود دارد ربات به سمت آن حرکت می‌کند. چنانچه هر دو شکاف پیشرفته چپ و راست وجود داشته باشد، ربات به سمت شکاف پیشرفته چپ حرکت می‌کند. همانطور که در بخش قبل اشاره شد، ایده اصلی ما برای کم کردن تعداد چرخش‌ها در مسیر، حداکثر استفاده از یک راستای انتخاب شده است. بنابراین ربات بعد از اینکه یک راستا را برای حرکت انتخاب کرد به حرکتش ادامه می‌هد تا وقتی که یکی از رخدادهای زیر پیش بیاید.

۱. ربات به یک نقطه‌ی u روی یک دیوار برخورد کند به طوری که قادر به ادامه حرکت در راستای فعلی نباشد. در این نقطه یکی از زنجیرهای ساعت‌گرد چندضلعی از s به u یا پادساعت‌گرد چندضلعی از s به u از هر مسیری که s را به u وصل می‌کند،به طور کامل قابل دید است [۲۳]. با توجه به مطلب اخیر واضح است که شکاف مربوط به آن زنجیر تا رسیدن به این نقطه غیر فعال شده است. ربات در این نقطه به سمت شکاف پیشرفته‌ای که تا نقطه u هنوز فعال باقی مانده است، می‌چرخد. در این نقطه نیز مانند نقطه شروع حرکت، هر یک از دو شکاف پیشرفته که موجود باشند، دوباره فعال می‌شوند. نقطه u در شکل ۲.۵ را ببینید.

۲. ربات به نقطه‌ای برسد که هیچ شکاف فعالی وجود نداشته باشد. از آنجائی که با ادامه حرکت در راستای کنونی هیچ یک از نواحی مخفی پشت شکاف‌های پیشرفته کاهش نمی‌یابد، ربات ناگزیر از چرخیدن است. در چنین نقطه‌ای یکی از حالت‌های زیر رخ داده است:

- ربات به نقطه‌ای رسیده است که تنها شکاف پیشرفته موجود با یک شکاف غیر فعال که تازه ایجاد شده است ادغام شود. نقطه ۳ در شکل ۳.۵آ و نقطه ۲ در شکل ۳.۵ب را ببینید. در این نقطه ربات به سمت شکاف ادغام شده حرکت می‌کند و این شکاف به عنوان یک شکاف فعال در نظر گرفته می‌شود.
- ربات به نقطه‌ای رسیده است که تنها شکاف پیشرفته فعال موجود ناپدید شود. نقطه ۲ در شکل ۴.۵ را ببینید. در این نقطه ربات به سمت شکاف پیشرفته موجود حرکت می‌کند و این شکاف را به عنوان یک شکاف فعال در نظر می‌گیرد.

فرایند فوق ادامه پیدا می‌کند تا اینکه ربات به نقطه‌ای برسد که هدف از این نقطه قابل دید است.



شکل ۴.۵: در نقطه ۱ مکان شکاف پیشرفته چپ عوض می‌شود و غیر فعال می‌شود. در نقطه ۲ شکاف پیشرفته راست (تنها شکاف پیشرفته موجود) ناپدید می‌شود. مسیر نقطه چین $SP(s,t)$ است و u_iu_j یک گوش روی این مسیر است.

۱.۲.۵ تحلیل الگوریتم

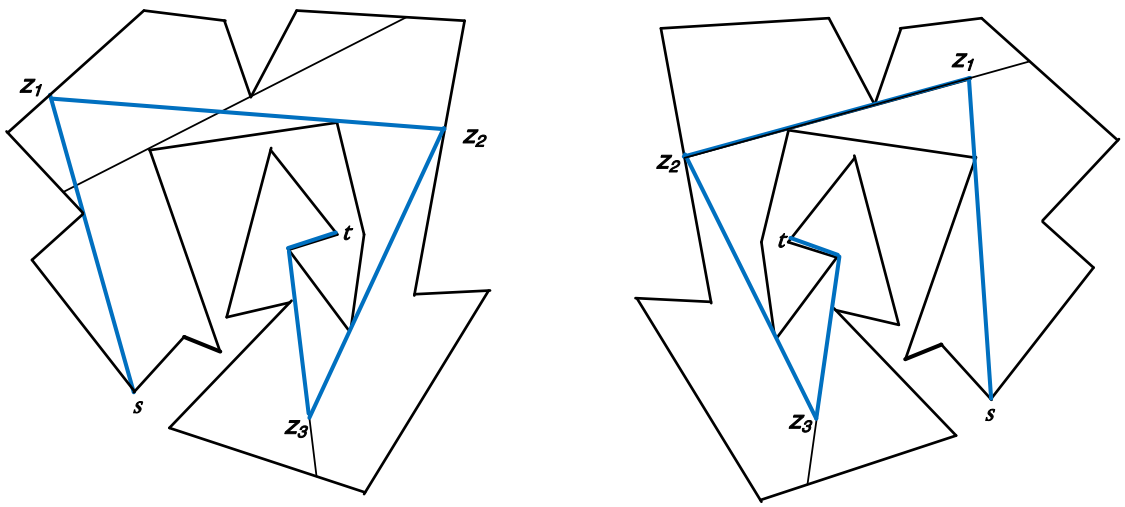
در این بخش تعداد چرخش‌هایی را که ربات برای رسیدن به هدف انجام می‌دهد، می‌شماریم. علاوه بر این، یک ضریب رقابتی ثابت برای راهبرد فوق برای رسیدن به هدف ارائه می‌دهیم. فرض کنید $SP(s,t)$ کوتاه‌ترین مسیر اقلیدسی از s به t باشد. یک یال u_iu_j از $SP(s,t)$ یک گوش[۴] نامیده می‌شود اگر u_{i-1} و u_{j+1} دو طرف خطی که u_i به u_j وصل می‌کند قرار بگیرند [۲۴]، شکل ۴.۵ را ببینید. برای شمارش تعداد چرخش‌های مسیری که راهبرد ارائه شده تولید می‌کند، ابتدا حالت ساده‌ای که در آن، $SP(s,t)$ فقط چپ‌گردی یا فقط راست‌گردی دارد، بررسی می‌کنیم. به عبارت دیگر در این حالت ساده $SP(s,t)$ گوشی ندارد.

لم ۲.۵ در حالتی که نقطه هدف از نقطه شروع قابل دیدن نیست و $SP(s,t)$ فقط چپ‌گردی دارد، راهبرد ما مسیری را برای رسیدن ربات به نقطه t تولید می‌کند که تعداد چرخش‌های آن بهینه است.

اثبات. چون نقطه هدف از نقطه شروع قابل دیدن نیست، ربات به سمت g_l حرکت می‌کند و در یکی از دو حالتی که در الگوریتم شرح داده شد، ربات می‌چرخد. ۱) در حالتی که به دیوار برخورد کند به سمت g_l کنونی می‌چرخد. ۲) در حالتی که به نقطه‌ای برسد که هیچ شکاف فعالی وجود ندارد، به

[۴] Eave

سمت شکاف موجود می‌چرخد. از آنجایی که $SP(s,t)$ فقط چپ‌گردی دارد، شکاف موجود یک g_l است. واضح است که در هر یک از دو حالت فوق اگر از نقطه چرخش دو مماس بر $SP(s,t)$ رسم کنیم، هر دو مماس در داخل خیابان می‌افتد. به همین ترتیب کل نقاط چرخش روی مسیر این ویژگی را دارند که اگر دو مماس از آن‌ها بر $SP(s,t)$ بکشیم، هر دو مماس در داخل خیابان می‌افتد. این نقاط را $z_1, z_2, ..., z_n$ می‌نامیم، شکل ۵.۵آ را ببینید. توجه کنید که هر مسیر با کمترین تعداد پیوند باید تمام مماس‌های چپ بر $SP(s,t)$ از نقطه‌هایی مانند $z_1, z_2, ..., z_n$ قطع کند [۲۴]. واضح است مسیری با کمترین تعداد چرخش نمی‌تواند بیش از یکی از خطوط مسیر ساخته شده توسط الگوریتم را قطع کند. بنابراین مسیر به دست آمده توسط راهبرد ما کمترین تعداد چرخش را دارد. ■



(ب) (آ)

شکل ۵.۵: دو شکل ساده از خیابان. (آ) $SP(s,t)$ فقط چپ‌گردی دارد. (ب) $SP(s,t)$ فقط راست‌گردی دارد.

لم ۳.۵ در حالتی که نقطه هدف از نقطه شروع قابل دید نیست و $SP(s,t)$ فقط راست‌گردی دارد، راهبرد ما مسیری را برای رسیدن ربات به نقطه t تولید می‌کند که تعداد چرخش‌های آن حداکثر یک چرخش بیشتر از مسیر بهینه است.

اثبات. اگر ربات در نقطه شروع حرکت هم وجود شکاف راست و هم وجود شکاف چپ را گزارش دهد، با توجه به راهبرد ارائه شده، ربات به سمت شکاف پیشرفته چپ حرکت می کند. در حالتی که به دیوار برخورد کند، به سمت g_r کنونی می‌چرخد. در حالتی که به نقطه‌ای برسد که هیچ شکاف فعالی

وجود ندارد، به سمت شکاف موجود می‌چرخد. از آنجایی که $SP(s,t)$ فقط چپ‌گردی دارد، شکاف موجود یک g_r است. به همین ترتیب از این به بعد ربات همواره به سمت یک g_r می‌چرخد. مانند اثبات لم قبل، در همه خطوط سازنده مسیرش، با خطوط مسیر با کمترین تعداد چرخش برخورد دارد مگر در اولین خط سازنده اش. بنابراین ربات با طی کردن مسیری با حداکثر یک چرخش بیش از مسیر بهینه به هدف می‌رسد. شکل ۳.۵ب و شکل ۵.۵ب را ببینید. ■

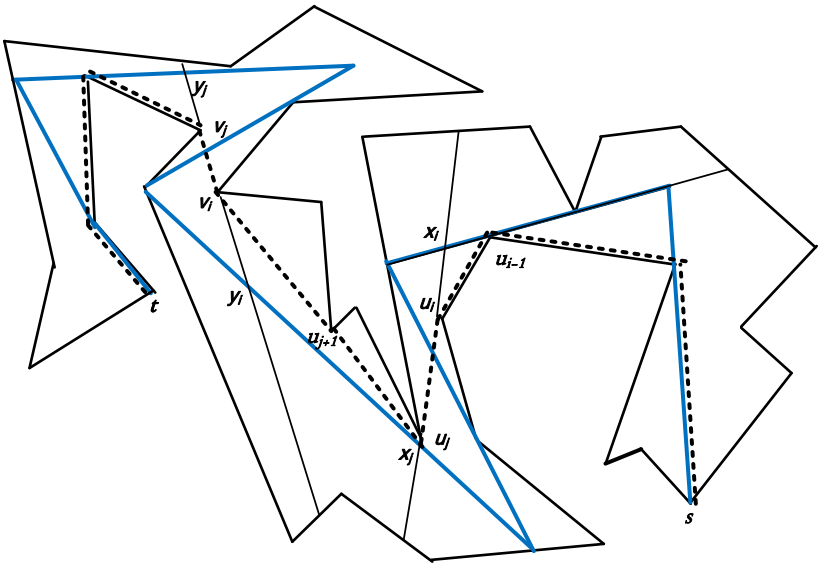
اکنون حالت کلی را در نظر می‌گیریم که در آن $SP(s,t)$ هم چپ‌گردی و هم راست‌گردی دارد. به عبارت دیگر $SP(s,t)$ تعدادی گوش دارد. از قضیه زیر برای مقایسه تعداد چرخش‌های ربات با مسیر با کمترین تعداد چرخش استفاده می‌کنیم.

قضیه ۴.۵ [۲۴] مسیری با کمترین تعداد چرخش وجود دارد که حاوی تمام گوش‌ها روی $SP(s,t)$ است.

قضیه ۵.۵ با راهبرد ارائه شده ربات با شروع از نقطه s در خیابان به نقطه هدف t با عبور از مسیری با حداکثر $m+1+e$ پیوند می‌رسد. m تعداد پیوندها در مسیر با کمترین تعداد چرخش از s به t است و e تعداد گوش‌ها در $SP(s,t)$ است. همچنین راهبرد ارائه شده برای ربات یک راهبرد بهینه است.

اثبات. در حالت کلی که تعدادی گوش در $SP(s,t)$ وجود دارد، فرض کنید که u_iu_j اولین گوش در این مسیر باشد. فرض کنید که $SP(s,t)$ از s تا u_i فقط چپ‌گردی داشته باشد و در نقطه u_j چرخش به سمت راست داشته باشد. این گوش را از دو طرف ادامه دهیم تا مرزهای چندضلعی را قطع کند. با راهبرد ارائه شده در این بخش، ربات وقتی که از نقطه s شروع به حرکت می‌کند با حداکثر یک چرخش اضافه به نقطه‌ای مانند x_i روی امتداد گوش می‌رسد. با توجه به اینکه $SP(s,t)$ در نقطه u_i به سمت چپ می‌چرخد، در راهبرد ارائه شده، شکاف پیشرفته راست غیر فعال شده و ربات در نقطه‌ی چرخش به سمت شکاف پیشرفته چپ می‌چرخد و با گوش برخورد می‌کند و چون $SP(s,t)$ در نقطه‌ی u_j به سمت راست می‌چرخد، در راهبرد ارائه شده، شکاف پیشرفته چپ غیر فعال شده و ربات در نقطه‌ی چرخش به سمت شکاف پیشرفته راست می‌چرخد. $SP(s,t)$ از این نقطه تا امتداد گوش بعدی چپ‌گردی دارد، ربات از این نقطه تا امتداد گوش بعدی را با تعداد چرخش بهینه طی می‌کند. ربات با حداکثر طی کردن دو پاره خط دو امتداد هر گوش را قطع می کند و فاصله بین دو گوش متوالی را هم با تعداد بهینه چرخش طی می‌کند، شکل ۶.۵ را ببینید. از آنجایی که مسیری وجود دارد که دربرگیرنده تمام گوش‌ها است (طبق قضیه ۳.۷) پس تعداد پیوندها در مسیر ربات $m+1+e$ است. راهبرد ارائه شده به این

دلیل بهینه است که تعداد پیوندها در مسیری که طی می‌کند با تعداد پیوندهایی که مسیر بهینه ربات به کار برده شده در [۲۳] با دید کامل طی می‌کند، برابر است. ■



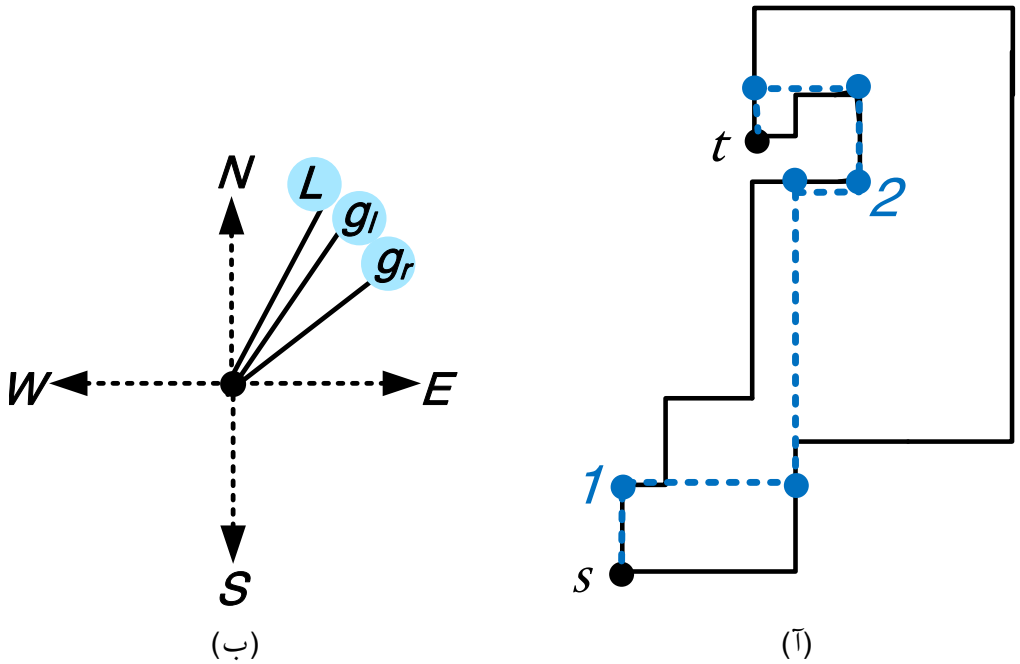
شکل ۶.۵: حالت کلی که کوتاه‌ترین مسیر از s به t هم راست‌گردی و هم چپ‌گردی دارد. مسیر پر رنگ مسیر حرکت ربات را نشان می‌دهد.

قضیه ۶.۵ راهبرد ارائه شده با استفاده از یک حافظه ثابت در حالی که ربات را به هدف می‌رساند متوقف می‌شود. ضریب رقابتی این راهبرد $۲ - ۱/m$ است.

اثبات. تمام آنچه ما برای هدایت ربات نیاز داریم، مکان شکاف‌های پیشرفته و وضعیت فعال بودن آن‌ها است. بر خلاف راهبردهای قبلی ارائه شده ما نیازی به نگهداری نقشه‌ای از محیط تاکنون پیمایش شده، نداریم. مکان این دو شکاف و وضعیت فعال بودن آن‌ها را می‌توان در یک حافظه ثابت نگه داشت. در طول مسیر حرکت مکان و وضعیت شکاف‌ها می‌تواند عوض شود که تعداد این تغییرات متناهی است و هر یک مربوط به عبور از روی شعاع انعکاسی یا عبور از روی مماس مشترک راس‌های انعکاسی است. هر بار که ربات به سمت یک شکاف پیشرفته حرکت می‌کند یک گام به هدف نزدیک می‌شود. بنابراین ربات در زمان متناهی به هدف می‌رسد. طبق قضیه ۳.۷ می‌دانیم $e \leq m - ۲$، پس تعداد پیوندها در مسیری که ربات طی می‌کند $۲m - ۱$ است. مطلب اخیر نشان می‌دهد ضریب رقابتی $۲ - ۱/m$ است. ■

۳.۵ هدایت ربات در یک خیابان متعامد در امتداد یک مسیر متعامد با کمترین تعداد چرخش

اکنون حالت خاص از مسئله را که ربات مجبور است در امتدا یک مسیر متعامد در یک خیابان متعامد حرکت کند تا به هدف برسد را بررسی می‌کنیم. به عبارت دیگر ربات مجهز به حسگر شکاف به یک قطب‌نما هم تجهز شده که جهت‌های شرق، غرب، شمال و جنوب را به او نشان می‌هد. ربات در حالی که مکان شکاف‌ها را تشخیص می‌دهد، فقط قادر است در امتداد جهت‌های فوق حرکت کند (شکل ۷.۵).

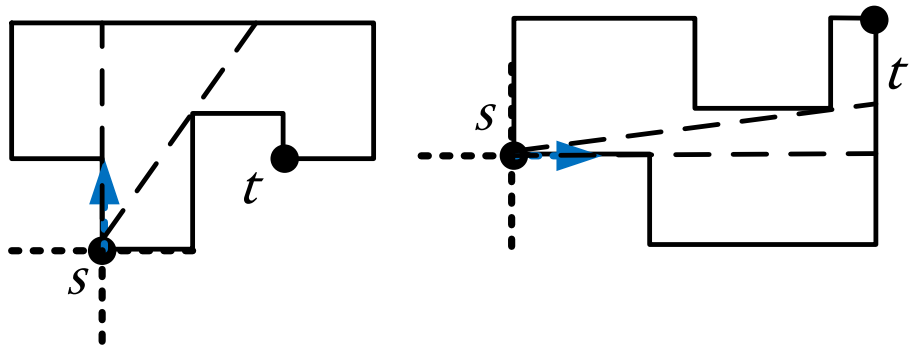


(ب) (آ)

شکل ۷.۵: یک خیابان متعامد. (آ) مسیر نقطه‌چین، مسیری که ربات برای رسیدن به هدف طی می‌کند. (ب) نمایش مکان شکاف‌هایی که ربات در نقطه شروع درک می‌کند و جهت‌هایی که ربات قادر به حرکت در راستای آن‌ها است.

ربات از نقطه s در خیابان متعامد شروع به حرکت می‌کند و تا وقتی که نقطه هدف را ندیده است، هدف پشت یکی از شکاف‌های پیشرفته است. در طول حرکت، شکاف‌های پیشرفته در یک درخت دو شاخه‌ای ذخیره می‌شود، این درخت در طول مسیر به صورت پویا همان‌طور که در بخش قبل شرح داده شد، به‌روز می‌شود. در نقطه شروع حرکت، مکان شکاف‌های پیشرفته در یک خیابان متعامد به صورت یکی از حالت‌هایی است که در شکل‌های ۸.۵، ۹.۵ و ۱۰.۵ نشان داده شده است. ربات بر اساس اینکه موقعیت شکاف‌های پیشرفته به چه صورت است، جهت حرکتش را مطابق زیر انتخاب می‌کند.

۱. وقتی که یکی از شکاف‌های پیشرفته متعامد باشند (در راستای جهت‌هایی باشد که قطب‌نما نشان



شکل ۸.۵: در نقطه شروع حرکت یک شکاف متعامد وجود دارد. ربات راستای این شکاف را برای حرکت انتخاب می‌کند.

می‌دهد)، ربات به سمت آن شکاف پیشرفته حرکت می‌کند (شکل ۸.۵)

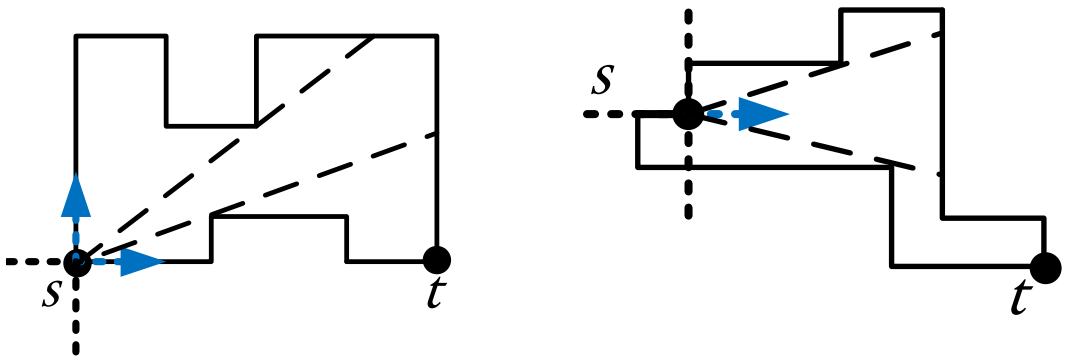
۲. وقتی که دو شکاف پیشرفته در دو ربع متوالی از دستگاه مختصات باشند. به عبارت دیگر یکی از جهت‌هایی که قطب‌نما نشان می‌دهد میان مکان دو شکاف پیشرفته چپ و راست باشد (شکل ۹.۵آ).

۳. وقتی که فقط یک از شکاف‌های پیشرفته وجود داشته باشد. اگر این شکاف پیشرفته g_l باشد ربات در اولین راستایی از قطب‌نما که در جهت عقربه‌های ساعت بعد از شکاف قرار دارد حرکت می‌کند. به همین ترتیب اگر این شکاف پیشرفته g_r باشد ربات در اولین راستایی از قطب‌نما که در جهت خلاف عقربه‌های ساعت بعد از شکاف قرار دارد، حرکت می‌کند. شکل ۱۰.۵ را ببینید.

۴. وقتی که هر دو شکاف پیشرفته در یک ربع از دستگاه مختصات قرار دارند. به عبارت دیگر هیچ یک از راستاهایی که قطب‌نما نشان می‌دهد، ما بین آن‌ها قرار نگیرد. در این حالت ربات به صورت دلخواه یکی از دو راستای در برگیرنده آن‌ها را انتخاب می‌کند (شکل ۹.۵ب).

ربات به حرکت در امتداد راستای انتخاب شده ادامه می‌دهد تا یکی از رخدادهای زیر اتفاق بیفتد. لازم به ذکر است که در کل مسیر تا قبل از اینکه هدف دیده شود مکان شکاف‌های پیشرفته چپ و راست به یکی از چهار صورت فوق است.

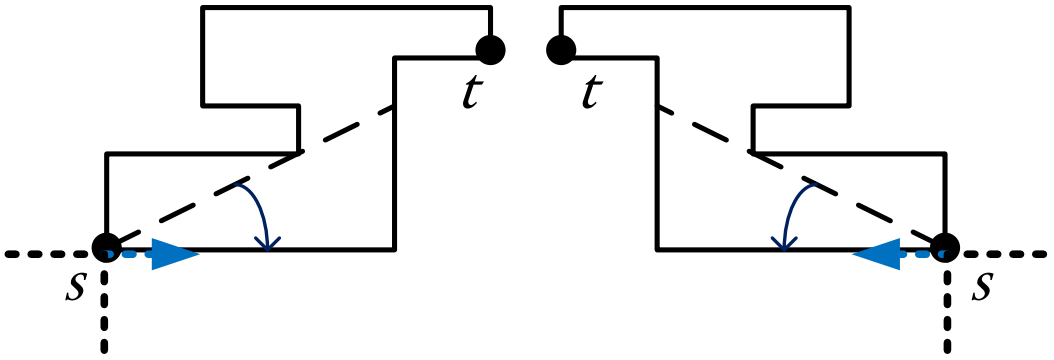
- ربات به یک دیوار برخورد کند به طوری که قادر به ادامه حرکت در راستای کنونی نباشد. در این نقطه ربات بر اساس مکان فعلی شکاف‌های پیشرفته مانند نقطه شروع جهتی را برای حرکت انتخاب می‌کند. این جهت عمود بر جهت حرکت قبلی‌اش است (نقطه ۱ در شکل ۷.۵آ).



(ب) (آ)

شکل ۹.۵: (آ) یک راستای متعامد در بین دو شکاف قرار دارد و ربات این راستا را برای حرکت از نقطه شروع انتخاب می‌کند. (ب) ربات یکی از راستاهایی که بلافاصله قبل یا بعد از شکاف‌های پیشرفته قرار دارد را برای حرکت از نقطه شروع انتخاب می‌کند.

- ربات به نقطه‌ای برسد که مکان شکاف‌های پیشرفته بر راستای حرکت کنونی‌اش عمود باشد. به عبارت دیگر مکان یکی از شکاف‌ها پیشرفته منطبق با یک از جهت‌های قطب‌نما که با جهت حرکش فعلی‌اش متفاوت است، باشد (نقطه ۲ در شکل ۷.۵آ). ربات در این نقطه به سمت آن شکاف پیشرفته می‌چرخد.



شکل ۱۰.۵: در نقطه شروع فقط یکی از شکاف‌های پیشرفته چپ یا راست وجود دارد. فلش پر رنگ جهت حرکت ربات را نشان می‌دهد.

فرایند فوق ادامه پیدا می‌کند تا ربات به نقطه‌ای برسد که از آن نقطه، هدف t قابل دیدن باشد. در این نقطه ربات به حرکت در امتداد کنونی ادامه می‌دهد تا اینکه به نقطه‌ای برسد که نقطه هدف در امتداد یکی از جهت‌هایی که قطب‌نما نشان می‌دهد و متفاوت با جهت حرکت فعلی است، قابل دید باشد. در این نقطه ربات، به سمت هدف در جهتی که قطب‌نما نشان می‌دهد، می‌چرخد. مسیر حرکت

ربات در شکل در شکل ۷.۵آ نشان داده شده است.

اکنون نشان می‌دهیم که ربات با طی کردن مسیری که دارای حداکثر $m+1$ پیوند افقی-عمودی است با شروع از نقطه s به نقطه هدف t می‌رسد.

لم ۷.۵ یک مسیر با کمترین تعداد پیوند از s به t وجود دارد که تمام پیوندهای آن مگر اولی بر پیوندهای مسیری که ربات ما طی می‌کند، منطبق است.

اثبات. در نقطه شروع حرکت، ربات بر اساس مکان شکاف‌های پیشرفته جهتی را برای حرکت انتخاب می‌کند. چنانچه در نقطه شروع حرکت هر یک از شرایط ۱ تا ۳ اتفاق بیفتد، ربات جهتی را انتخاب می‌کند که با حرکت در این راستا شکاف‌های پیشرفته موجود فعال هستند. چه نقطه‌ی هدف پشت g_l چه پشت g_r مخفی باشد، با حرکت در این راستا یک گام به سمت آن نزدیک می‌شویم. واضح است که مسیری با کمترین تعداد پیوند وجود دارد که پیوند اولش بر مسیر حرکت ربات منطبق است. اگر حالت چهارم در نقطه شروع حرکت رخ بدهد، هر دو شکاف پیشرفته در یک ربع از دستگاه مختصات قرار دارند. ربات یکی از جهت‌های متعامد سازنده آن ربع را انتخاب می‌کند و در راستای آن حرکت می‌کند. در طول حرکت یکی از شکاف‌های پیشرفته فعال است و دیگری فعال نیست. چنانچه هدف پشت شکاف فعال باشد، واضح است که مسیری با کمترین تعداد پیوند وجود دارد که پیوند اولش بر مسیر حرکت ربات منطبق است، در غیر اینصورت ربات یک پیوند اضافه را طی می‌کند. در شکل ۷.۵آ در نقطه شروع، در راستای حرکت ربات، شکاف پیشرفته چپ غیر فعال است و چون هدف پشت این شکاف است، این راستا منجر به یک پیوند اضافه در مسیر حرکت ربات می‌شود. در طول حرکت هم ربات زمانی می‌چرخد که یک شکاف در یکی از راستاهای قطب‌نما قرار بگیرد. در این نقطه وضعیت شکاف‌ها مانند حالت ۱ در الگوریتم است. این مطالب نشان می‌دهد که ربات فقط ممکن است یک پیوند اضافه در طول مسیرش برای رسیدن به هدف داشته باشد.

■

نتیجه اصلی این بخش در قضیه زیر خلاصه شده است.

قضیه ۸.۵ راهبرد ارائه شده در این بخش یک مسیر جستجوی متعامد بهینه با ضریب رقابتی $1+1/m$ تولید می‌کند.

اثبات. بحثی مشابه آنچه برای قضیه ۶.۵ مطرح شد، اثبات می‌کند که ربات در زمان متناهی با استفاده از یک حافظه ثابت به هدف می‌رسد. ضریب رقابتی هم از روی تعداد پیوندها که در لم قبل محاسبه

شد، به دست می‌آید. از آنجایی که تعداد پیوندهای مسیر ربات با تعداد پیوندهای مسیر بهینه‌ای که یک ربات با دید کامل از محیط دارد برابر است، پس این مسیر هم بهینه است.

■

۱.۳.۵ جمع‌بندی

در این فصل برای هدایت ربات ساده‌ای که چرخش برایش هزینه دارد در حالی که حرکت در راستای مستقیم هزینه زیادی ندارد، یک راهبرد جستجو ارائه شد. با راهبرد ارائه شده ربات از نقطه شروع s در خیابان به نقطه هدف t می‌رسد به طوری که تعداد پیوندهای مسیر بهینه است.

فصل ۶

پیمایش یک چندضلعی ستاره‌ای توسط یک ربات ساده

در این فصل مسئله پیمایش یک چندضلعی ستاره‌ای توسط ربات مجهز به حسگر شکاف را مطرح می‌کنیم. یک چندضلعی ستاره‌ای یک چندضلعی ساده است که ناحیه‌ای در آن وجود دارد که هریک از نقاط چندضلعی از یکی نقاط داخل این ناحیه قابل دیدن باشد. به این ناحیه، هسته چندضلعی گفته می‌شود. یک ربات ساده از یک نقطه s درون چندضلعی شروع به حرکت می‌کند، هدف این است که ربات با استفاده از اطلاعاتی که از طریق حسگرهایش از محیط به دست می‌آورد به یک نقطه از هسته برسد. نکته مهمی که در هدایت ربات برای رسیدن به هسته وجود دارد این است که طول مسیری که ربات برای رسیدن به هسته طی می‌کند از طول کوتاه‌ترین مسیر خیلی بیشتر نباشد. بعلاوه در این فصل الگوریمی برای هدایت ربات برای رسیدن به هر نقطه دلخواه از چندضلعی ستاره‌ای ارائه می‌دهیم. نشان می‌دهیم که طول این مسیر هم ضریب ثابتی از طول کوتاه‌ترین مسیر است. به عبارت دیگر نسبت طول مسیر طی شده تا هسته به طول کوتاه‌ترین مسیر یک عدد ثابت است. همانطور که قبلا ذکر شد، چنین راهبردی یک راهبرد رقابتی نامیده می‌شود.

برای رساندن رباتی که دید کامل از محدوده رویت‌پذیرش دارد، به هسته چندضلعی، راهبردهایی ارائه شده است که بهترین آن دارای ضریب رقابتی ۳/۱۲۶ است. همچنین ثابت شده که $\sqrt{۲}$ کران پایین ضریب رقابتی برای این مسئله است [۳۰، ۴۷]. در این فصل ما برای ربات ساده‌ای که مجهز به حسگر شکاف است و یک قطب‌نما برای تشخیص جهت‌های جغرافیایی همراه دارد یک راهبرد رقابتی ارائه می‌دهیم که ربات با حرکت از هر نقطه درون چندضلعی بتواند به هسته برسد.

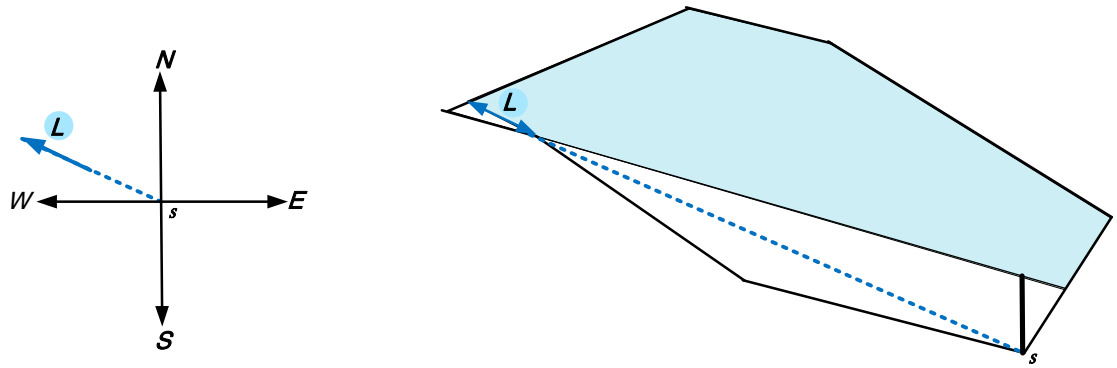
۱.۶ ضرورت وجود قطب‌نما

پیمایش یک محیط ناشناخته به این ترتیب است که یک ربات بدون اینکه از قبل شناختی از محیط داشته باشد، در محیط حرکت می کند. با استفاده از اطلاعاتی از طریق حسگرهایش به دست می‌آورد برای رسیدن به هدف تلاش می کند. در این فصل هدف این است که ربات ساده ما که مجهز به حسگر شکاف است به نقطه‌ای از یک چندضلعی ستاره‌ای برسد که بتواند از آن‌جا کل محدوده را زیر نظر بگیرد. به عبارت دیگر قادر باشد کل محدوده کاری را از آن نقطه ببیند. یک چندضلعی ستاره‌ای است اگر نقطه‌ای در آن وجود داشته باشد که از آن نقطه کل مرزهای چندضلعی قابل دیدن باشد. به مجموعه چنین نقاطی هسته چندضلعی گفته می شود. نمونه‌هایی از چندضلعی‌های ستاره‌ای در شکل‌های ۱.۶آ و ۲.۶ آمده است. در پژوهش‌های قبلی، رباتی که برای رسیدن به چنین نقطه‌ای از محیط انتخاب شده بود، رباتی است که دید کامل از محیط دارد و نقشه‌ی کاملی از محدوده قابل دیدش را درک می‌کند. این ربات پس از پیمایش محیط کل نقشه محیط را ضبط می‌کند. در این پژوهش هدف این است که به دلیل هزینه کمتر، انعطاف‌پذیری بیشتر، استفاده از حافظه کمتر و سرعت بالاتر از ربات‌های ساده استفاده کنیم.

ربات مجهز به حسگر شکاف در محیط حرکت می‌کند و قادر به درک مکان ناپیوستگی‌های عمقی یا شکاف‌ها است و فقط می‌تواند به سمت این شکاف‌ها حرکت کند. برای چنین رباتی می‌توان مثال‌هایی زد که این ربات با قابلیت محدودش برای رسیدن به هسته به اندازه‌ی بسیار زیادی از کوتاه‌ترین برای رسیدن به هسته منحرف شود به طوری که نمی‌توان برای مسیر طی شده برای رسیدن به هدف ضریب رقابتی ثابتی به دست آورد. برای نمونه شکل ۱.۶آ را ببینید. برای حل این مشکل ما ربات را مجهز به یک قطب‌نما کردیم. بنابراین رباتی که در این‌جا استفاده می‌کنیم علاوه براین که قادر است موقعیت شکاف ها را تشخیص دهد و به سمت آن‌ها حرکت کند می‌تواند جهت‌های افقی و عمودی را تشخیص دهد و در امتداد جهت‌های افقی و عمودی حرکت کند (۱.۶ب).

۲.۶ مدل حسگر و محدودیت‌های حرکتی ربات

همان‌طور که در فصل‌های قبلی ذکر شد، ربات مجهز به حسگر شکاف ناپیوستگی‌های عمقی یا شکاف‌ها در اطرافش را به ترتیب گزارش می‌دهد و به هر یک از آن‌ها یک برچسب چپ یا راست اختصاص می‌دهد. ربات می‌تواند به سمت این شکاف‌ها حرکت کند. در طول حرکت ربات به سمت شکاف‌ها،



(ب) (آ)

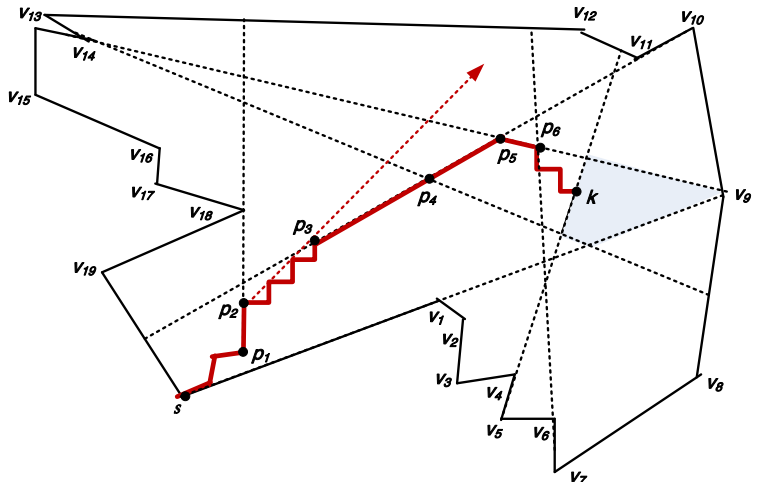
شکل ۱.۶: (آ)یک چندضلعی بسیار ساده ستاره‌ای که فقط ربات در نقطه شروع، فقط یک شکاف با برچسب L در آن حس می‌کند. ناحیه رنگی هسته آن است. ربات مجهز به حسگر ساده شکاف در نقطه شروع، با حرکت به سمت تنها شکاف موجود (مسیر نقطه‌چین) به اندازه زیادی از کوتاه‌ترین مسیر به سمت هسته دور می‌شود. (ب) آنچه ربات مجهز به حسگر شکاف و قطب‌نما در نقطه شروع از محیط درک می‌کند.

رخدادهایی پیش می‌آید که موقعیت شکاف‌ها عوض می‌شود. بعضی از شکاف‌ها ممکن است از بین بروند، شکاف‌ها می توانند در هم ادغام شوند، می‌توانند به دو قسمت تقسیم شوند و ممکن است یک شکاف جدید در حین حرکت ایجاد شود. رباتی که در فصل قبل از آن استفاده شد، فقط قادر بود به سمت این شکاف‌ها حرکت کند. اما همان‌طور که در بخش قبل توضیح داده شد، چنین رباتی برای رسیدن به هسته چندضلعی ممکن است، به اندازه‌ی خیلی زیادی از کوتاه‌ترین مسیر منحرف شود. بنابراین نمی توان برای چنین رباتی برای رسیدن به هدف یک راهبرد رقابتی ارائه داد. برای حل این مشکل سعی کرده‌ایم قدرت این ربات را افزایش دهیم به طوری که ربات را مجهز به یک قطب‌نما کرده‌ایم که می‌تواند جهت‌های شمال،جنوب، شرق و غرب را تشخیص دهد و قادر است در دو راستای متعامد افقی و عمودی یعنی همان جهت‌های جغرافیایی حرکت کند، شکل ۱.۶ب را ببینید. لازم به ذکر است که این ربات می‌تواند به سمت شکاف‌ها و در راستای افقی و عمودی به تعداد دلخواهی گام حرکت کند و هر وقت رخدادی اتفاق افتاد که وضعیت شکاف‌ها را تغییر می‌دهد، می‌تواند تغییر جهت بدهد و در راستای دیگری یا به سوی شکاف دیگری حرکت کند.

۳.۶ الگوریتم

در این بخش به بیان الگوریتمی جهت تولید مسیر برای ربات فوق می‌پردازیم با این هدف که مسیری که ربات طی می‌کند از کوتاه‌ترین مسیر خیلی منحرف نشود. باتوجه به این که ربات قادر به تشخیص

جهت‌های جغرافیایی است، فرض می‌کنیم این جهت‌ها همان محورهای مختصات X و Y هستند. بنابراین ربات قادر به تشخیص این است که شکاف‌ها در کدام یک از نواحی مثلثاتی قرار می‌گیرند. ربات را بر اساس اینکه موقعیت شکاف‌ها در کدام یک از نواحی مثلثاتی است، هدایت می‌کنیم. در بین مجموعه شکاف‌های چپ، سمت راست‌ترین (g_l) و در بین شکاف‌های راست، سمت چپ‌ترین (g_r) را مورد نظر قرار می‌دهیم. همان‌طور که در فصل قبل شرح داده شد، این شکاف‌ها برای ربات قابل شناسایی است.



شکل ۲.۶: ناحیه رنگی هسته چندضلعی ستاره‌ای است. مسیر تیره رنگ مسیر حرکت ربات از نقطه شروع s تا نقطه‌ای از هسته است.

وقتی که ربات از نقطه دلخواه درون چندضلعی ستاره‌ای شروع به حرکت می‌کند تا وقتی که به هسته‌ی چندضلعی نرسیده است، با مجموعه‌ای از شکاف‌های چپ و راست روبر است. واضح است که وقتی که به هسته می‌رسد، همه‌ی شکاف‌ها ناپدید شده‌اند. فرض کنیم که آخرین شکافی که ناپدید می‌شود تا ربات به هسته برسد، یک شکاف راست باشد. بنابراین تا زمانی که شکاف چپی وجود دارد، حتما شکاف راستی هم وجود دارد. بدون از دست رفتن کلیت مسئله فرض می‌کنیم که در ابتدای حرکت، پیشرفته‌ترین شکاف راست در ربع اول قرار گرفته باشد. وضعیت شکاف‌ها در سرتاسر مسیر حرکت به یکی از صورت‌های زیر است، در ادامه توضیح می‌دهیم که ربات در هر حالت چه جهتی را برای حرکت انتخاب می‌کند.

- چنانچه هر دو شکاف در ربع اول باشند، ربات یک گام به سمت شکاف پیشرفته راست و یک گام به سمت شکاف پیشرفته چپ حرکت می کند. نقطه s در شکل ۲.۶ را ببینید.
- چنانچه موقعیت دو شکاف طوری باشد که یکی در ربع اول و دیگری در ربع دوم باشد، به عبارت دیگر مختصات Y هر دو شکاف مثبت باشد در این حالت ربات به سمت شمال حرکت می کند.

نقطه p_1 شکل ۲.۶ را ببینید.

- چنانچه موقعیت دو شکاف طوری باشد که یکی در ربع سوم و دیگری در ربع چهارم باشد، به عبارت دیگر مختصات Y هر دو شکاف منفی است، در این حالت ربات باز هم به سمت شمال حرکت می‌کند.

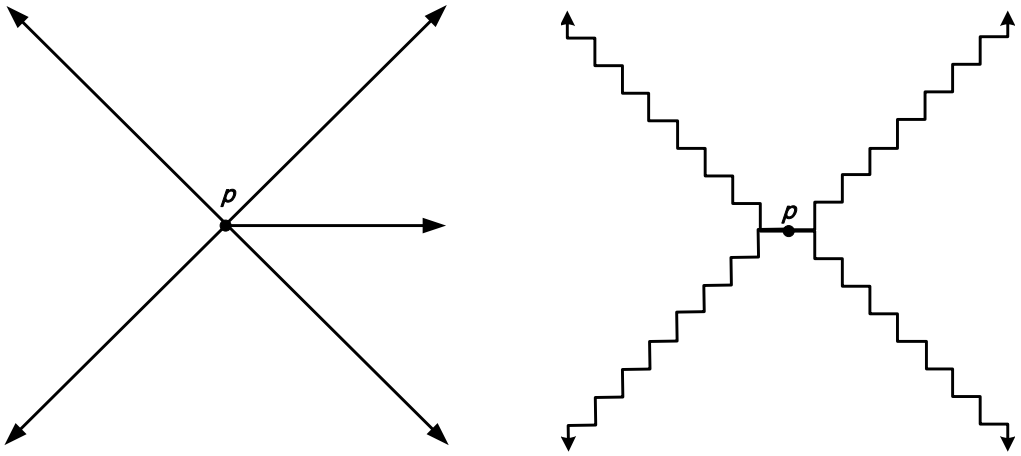
- در حالتی که شکاف پیشرفته راست در ربع اول یا دوم باشد و شکاف پیشرفته چپ در ربع سوم یا چهارم باشد، به عبارت دیگر شکاف پیشرفته راست دارای مختصات Y مثبت و شکاف پیشرفته چپ دارای مختصات Y منفی باشد، ربات یک گام در میان به سمت شمال و شرق حرکت می‌کند. نقطه p_2 شکل ۲.۶ را ببینید.

- در حالتی که شکاف پیشرفته راست در ربع سوم یا چهارم باشد و شکاف پیشرفته چپ در ربع اول یا دوم باشد، به عبارت دیگر شکاف پیشرفته راست دارای مختصات Y منفی و شکاف پیشرفته چپ دارای مختصات Y مثبت باشد، ربات یک گام در میان به سمت شمال و غرب حرکت می‌کند.

- در حالتی که فقط شکاف پیشرفته راست باقی مانده است، ربات دو محور افقی عمودی که در خلاف عقربه‌های ساعت بعد از این شکاف قرار دارند، در نظر می‌گیرد و یک گام در جهت یکی و یک گام در جهت دیگری می‌رود. ربات یکی از جهت‌های پلکانی که در شکل ۳.۶آ نشان داده شده را انتخاب می‌کند. در شکل ۲.۶ در نقطه p_6 فقط شکاف پیشرفته راست متصل به راس v_4 در ربع چهارم وجود دارد و ربات به سمت افقی عمودی شرق و جنوب حرکت می‌کند.

ربات در هر نقطه بسته به اینکه در کدام یک از وضعیت‌های فوق قرار دارد، مسیری را برای حرکت انتخاب می‌کند تا اینکه به نقطه‌ای برسد که وضعیت شکاف‌ها عوض شود. گاهی اوقات ادامه دادن به مسیر انتخاب شده باعث ایجاد یک شکاف جدید در مسیر حرکت ربات می‌شود. در این حالت ربات به جای حرکت در امتداد مسیر انتخاب شده، در راستای شکاف ایجاد شده حرکت می‌کند، نقطه p_3 در شکل ۲.۶ را ببینید. ربات به حرکت خود در راستای این شکاف ادامه می‌دهد تا اینکه به نقطه‌ای برسد که وضعیت شکاف‌ها عوض شود. سپس ربات براساس وضعیت مکانی جدید شکاف‌ها مسیر حرکتش را انتخاب می‌کند.

این فرایند ادامه پیدا می‌کند تا ربات به نقطه‌ای برسد که هیچ شکافی در مقابلش وجود نداشته باشد. واضح است که در این لحظه به نقطه‌ای از هسته رسیده است. در این نقطه هیچ شکافی یا به عبارت دیگر هیچ مکان غیر قابل دیدی برای ربات وجود ندارد.



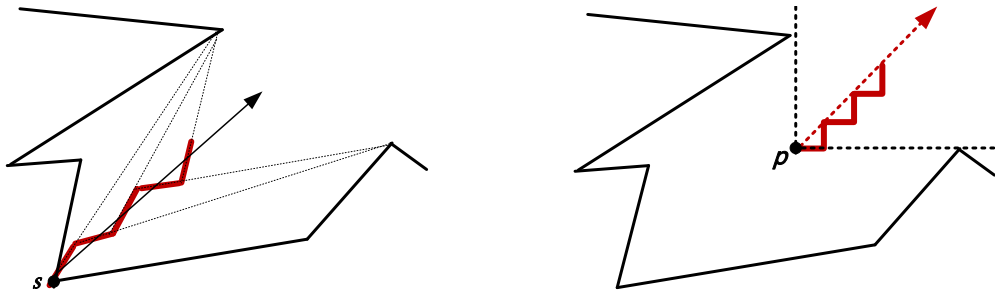
شکل ۳.۶: (آ)ربات ساده یکی از جهت‌های افقی عمودی که بعد از شکاف قرار دارد را برای حرکت انتخاب می‌کند. (ب) ربات با دید کامل، اولین جهتی که بعد از شکاف قرار دارد و زاویه آن تا شکاف بیش از ۴۵ درجه است را انتخاب می‌کند.

۴.۶ تحلیل درستی و محاسبه ضریب رقابتی الگوریتم

در این بخش ابتدا ثابت می‌کنیم که الگوریتم ارائه شده در بخش قبل ربات را در زمان متناهی به هسته چندضلعی می‌رساند. سپس ثابت می‌کنیم این الگوریتم مسیر حرکتی برای ربات می‌سازد که طول آن ضریب ثابتی از طول کوتاه‌ترین مسیر برای رسیدن به هسته چندضلعی است. نشان می‌دهیم ضریب رقابتی راهبرد ارائه شده ۱۲/۹۲ است. ربات از نقطه s شروع به حرکت می‌کند و بسته به مکان شکاف‌های پیشرفته به گونه‌ای حرکت می‌کند که نواحی مخفی پشت آن‌ها را ببیند. هر بار که این شکاف‌ها وضعیتشان به‌روز می‌شود، ربات ممکن است جهت حرکتش را عوض کند. از آنجائی که فقط زمانی که ربات از روی مماس مشترک رئوس انعکاسی عبور می‌کند، مکان پیشرفته‌ترین شکاف‌ها می‌تواند عوض شود حداکثر از مرتبه $O(n^۲)$ بار مکان این شکاف‌ها ممکن است تغییر کند. هر حرکت ربات در راستای آشکار کردن ناحیه پشت دو تا از این شکاف‌ها است. بنابراین ربات در زمان متناهی به هدف که همان هسته چندضلعی است می‌رسد. مطالب فوق قضیه زیر را ثابت می‌کند.

قضیه ۱.۶ ربات با استفاده از راهبرد ارائه شده، در زمان متناهی در حالی که به هسته چندضلعی ستاره‌ای رسیده است، متوقف می‌شود.

اکنون به محاسبه ضریب رقابتی می‌پردازیم. برای اینکه طول مسیری که ربات طی می‌کند با طول کوتاه‌ترین مسیر مقایسه کنیم، طول مسیری را که ربات خودمان طی می‌کند را با مسیری که ربات لی و



(آ) (ب)

شکل ۴.۶: (آ)ربات ساده در جهت افقی عمودی حرکت می‌کند در حالی که که فلش جهت حرکت ربات با دید کامل را نشان می‌دهد. (ب) ربات ساده در جهت یک گام به سمت شکاف راست و یک گام به سمت شکاف چپ گام بر می‌دارد در حالی که فلش جهت حرکت ربات با دید کامل را نشان می‌دهد.

همکارانش [۴۰] طی می‌کند، مقایسه می‌کنیم. آن‌ها ثابت کردند که راهبردشان دارای ضریب رقابتی $۲\sqrt{۲}+۱$ است. اکنون هر جزء از مسیر رباتمان را با همان قسمت از مسیر ربات لی و همکاران مقایسه می‌کنیم.

۱. هر وقت که ربات ما هر دو شکاف پیشرفته را دارای مختصات Y یکسان می‌بیند، در راستای شمال حرکت می‌کند، ربات آن‌ها نیز همین مسیر را طی می‌کند. به عبارت دیگر حرکت ربات‌ها در این حالت روی هم منطبق است.

۲. در حالتی که ربات ما یک گام به سمت شمال و یک گام به سمت شرق حرکت می‌کند ربات آن‌ها در جهت $+X+Y$ حرکت می کند. در شکل ۴.۶آ نمایش داده است. به راحتی اثبات می‌شود اختلافی که این مسیر با مسیر حرکت ربات ما دارد این است که طول مسیر ربات ما $\sqrt{۲}$ برابر طول مسیر آن ربات است.

۳. در حالتی که ربات ما یک گام به سمت شمال و یک گام به سمت غرب حرکت می‌کند ربات آن‌ها در جهت $-X+Y$ حرکت می کند. مانند حالت قبل، به راحتی اثبات می‌شود، اختلافی که این مسیر با حرکت ربات ما دارد این است که طول مسیر ربات ما $\sqrt{۲}$ برابر طول مسیر آن ربات است.

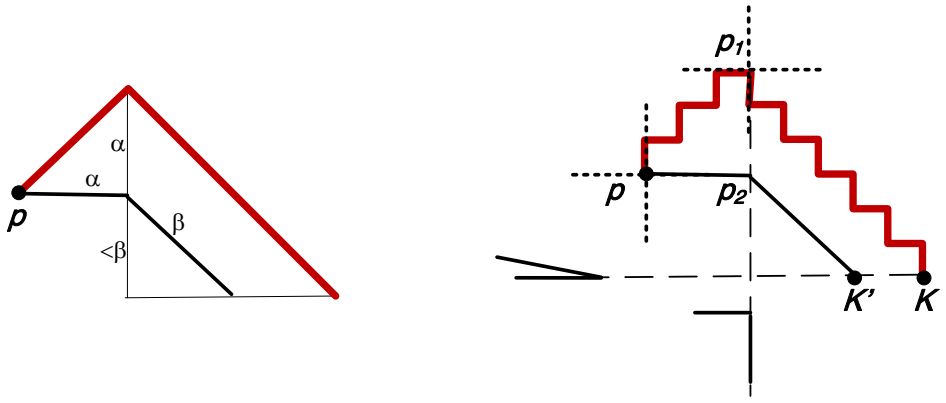
۴. در حالتی که ربات ما یک گام به سمت شکاف پیشرفته راست و یک گام به سمت شکاف پیشرفته چپ حرکت می‌کند، ربات آن‌ها این قابلیت را دارد که در یک راستای مابین آن‌ها حرکت کند. در این حالت نیز اثبات می‌شود مسیری که ربات ما طی می‌کند کمتر از $\sqrt{۲}$ برابر طول مسیر آن ربات است، شکل ۴.۶ب را ببینید.

۵. در حالتی که همه‌ی شکاف‌های یک سمت ناپدید شده‌اند (همان‌طور که از ابتدا فرض کردیم شکاف‌های سمت چپ) ربات ما در یکی از جهت‌هایی که در شکل ۳.۶آ نمایش داده شده است، حرکت می‌کند در حالی که ربات لی و همکاران در یکی از جهت های که در شکل ۳.۶ب نمایش داده شده است، حرکت می‌کند. ربات آن‌ها یکی از جهت‌های $\{+X+Y, +X-Y, -X+Y, -X-Y, +X\}$ را که در شکل نشان داده شده، انتخاب می‌کند. به طوری که این راستا، اولین راستایی از مجموعه فوق است که زاویه‌ی شکاف راست موجود در آن بیشتر از ۴۵ درجه است (در خلاف جهت عقربه‌های ساعت). در این حالت برای مقایسه طول مسیر ربات ما با طول مسیر آن ربات، به بررسی دقیق‌تری نیاز داریم. اگر شکاف راست موجود در ربع اول قرار دارد باز مسیر ربات ما $\sqrt{۲}$ برابر آن مسیر است. در طول مسیر حرکت ربات، شکاف راست می‌تواند از ربع اول وارد ربع چهارم (اکتانت هشتم) شود که در این حالت نیز مسیر ربات ما $\sqrt{۲}$ برابر آن مسیر است. اگر با ادامه حرکت، شکاف پیشرفته راست موجود از اکتانت هشتم وارد اکتانت هفتم شود، ربات ما به حرکت قبلی خود ادامه می‌دهد در حالی که ربات آن‌ها تغییر جهت حرکت می‌دهد و در راستای $+X$ حرکت می‌کند. با توجه به راهبرد حرکتی هر دو ربات، شکاف پیشرفته راست موجود برای هر دو ربات می‌تواند از ربع چهارم وارد ربع سوم شود که این باعث تغییر جهت حرکت هر دو ربات می‌شود، شکل ۵.۶ رابببنید. ربات‌ها به حرکت خود ادامه می‌دهند تا به نقطه‌ای برسند که در آن هیچ شکافی وجود نداشته باشد. با استفاده از شکل ۵.۶ب می‌توان دید، از نقطه‌ای به بعد، مسیر دو ربات از هم جدا می‌شود. مسیری که ربات لی و همکاران طی می‌کند $\alpha+\beta$ است ولی مسیری که ربات ما طی می‌کند برابر است با $\sqrt{۲}\alpha+\sqrt{۲}(\sqrt{۲}/۲\beta+\alpha)$. واضح است که مقدار اخیر از $(\sqrt{۲}+۱)(\alpha+\beta)$ کمتر است و از آنجایی که ربات ما هر حرکت مورب را با مجموعه‌ای از حرکات افقی-عمودی می سازد، مسیری که ربات ما طی می‌کند از $\sqrt{۲}$ برابر مقدار فوق کمتر است. بنابراین مسیری که ربات طی می‌کند از $(\sqrt{۲}+۲)(\alpha+\beta)$ کمتر است.

مطالب فوق اثبات کننده قضیه زیر است.

قضیه ۲.۶ راهبرد ارائه شده در این فصل، مسیری با ضریب رقابتی ۱۲/۹۲ برای ربات ساده‌ی مجهز به قطب‌نما، جهت رسیدن به هسته‌ی چندضلعی ستاره‌ای تولید می‌کند.

اثبات.



(ب) (آ)

شکل ۵.۶: (آ)ربات ساده در جهت افقی عمودی حرکت می‌کند در حالی که مسیر پیوسته از p به k مسیر حرکت ربات با دید کامل را نشان می‌دهد. (ب) مسیر افقی عمودی ربات برای ساده‌تر شدن مقایسه به صورت مورب نشان داده شده است.

مطالب فوق اثبات می‌کند، ضریب رقابتی راهبرد ارائه شده برای این ربات ساده $(۲ + \sqrt{۲})$ برابر راهبرد ارائه شده برای ربات لی و همکارانش است. از آنجائی که ضریب رقابتی راهبرد آن‌ها نیز $(۱ + ۲\sqrt{۲})$ است، ضرب این دو عدد ضریب رقابتی الگوریتم ما می‌شود. ■

۵.۶ جمع بندی

در این فصل از ربات ساده مجهز به حسگر شکاف برای رسیدن به هسته یک چندضلعی استفاده کردیم. به این نتیجه رسیدیم که این ربات ساده نمی‌تواند در راستای یک مسیر مناسبی به هسته برسد. بنابراین ربات را به یک قطب‌نما نیز تجهیز کردیم که قادر است، جهت‌های جغرافیایی را به ربات نشان دهد. برای ربات تقویت شده یک راهبرد رقابتی با ضریب ۱۲/۹۲ ارائه دادیم. لازم به ذکر است که مسئله مطرح شده در این فصل، در قالب یک پایان‌نامه کارشناسی ارشد نیز در دانشکده ارائه شد [۸۱] که در این جا تکمیل شده است.

فصل ۷

راهبرد رقابتی بهینه برای هدایت ربات با استفاده از اطلاعات محلی

در این فصل برای هدایت ربات راهبردی ارائه می‌شود که در آن ربات با استفاده از اطلاعات محلی[۱] که توسط حسگرهایش گزارش می‌شود و بدون نگهداری اطلاعات محیط به جستجو در محدوده خیابان می‌پردازد. در الگوریتم قبلی که ارائه شده بود، ربات بخشی از اطلاعات محیط را نگهداری می‌کرد و همچنین از تعدادی نشانه‌گذار[۲] برای علامت‌گذاری محیط استفاده می‌کرد. در ابتدا یک الگوریتم قطعی برای هدایت ربات با ضریب رقابتی ۹ ارائه می‌شود. اثبات می‌شود که ربات یک مسیر بهینه را طی می‌کند. سپس الگوریتمی تصادفی براساس الگوریتم فوق ارائه می‌شود. ضریب رقابتی مورد انتظار الگوریتم تصادفی ۵/۳۳ است [۵۸].

لازم به ذکر است که اخیرا راهبردی رقابتی برای مسیریابی بین نقاط در صفحه با وجود محدودیت‌ها توسط بوز[۳] و همکارانش ارائه شده است [۷].

---

[۱] Local Information

[۲] Pebble

[۳] Bose

۱.۷ مفاهیم اولیه مورد نیاز

در این بخش به مفاهیم مورد نیاز برای هدایت ربات می‌پردازیم. از این مفاهیم قبلا برای ارائه راهبرد قطعی جستجو استفاده کرده‌ایم [۲۵، ۵۴، ۵۵].

در نقطه شروع حرکت، ربات با مجموعه‌ای از شکاف‌ها با برچسب‌های L و R روبرو می‌شود. با توجه به ساختار محدوده کاری خیابان، نیازی نیست که برای رسیدن به هدف کل شکاف‌ها پیموده شود. بین شکاف‌هایی که دارای برچسب R هستند، هدف فقط می‌تواند پشت سمت چپ‌ترین (پیشرفته‌ترین) شکاف راست که با g_r نشان می‌دهیم، مخفی باشد و از بین شکاف‌هایی که دارای برچسب L هستند، فقط می‌تواند پشت سمت راست‌ترین (پیشرفته‌ترین) شکاف چپ که با g_l نشان می‌دهیم مخفی باشد [۴۵]. در هر نقطه از مسیر حرکت اگر فقط یکی از شکاف‌ها وجود داشته باشد، هدف پشت این شکاف است و ربات بدون هیچ ابهامی به سمت آن حرکت می‌کند . اگر هر دو شکاف پیشرفته وجود داشته باشند، هدف پشت یکی از آن‌ها قرار دارد. در این حالت که یک قیف نامیده می‌شود، ربات برای رسیدن به هدف مجبور است پشت هر دو شکاف را ببیند.

از آنجایی که ربات هیچ اطلاعی در مورد محیط به جز مکان شکاف‌ها ندارد، انحراف از کوتاه‌ترین مسیر غیر قابل اجتناب است.

۲.۷ اطلاعات اساسی

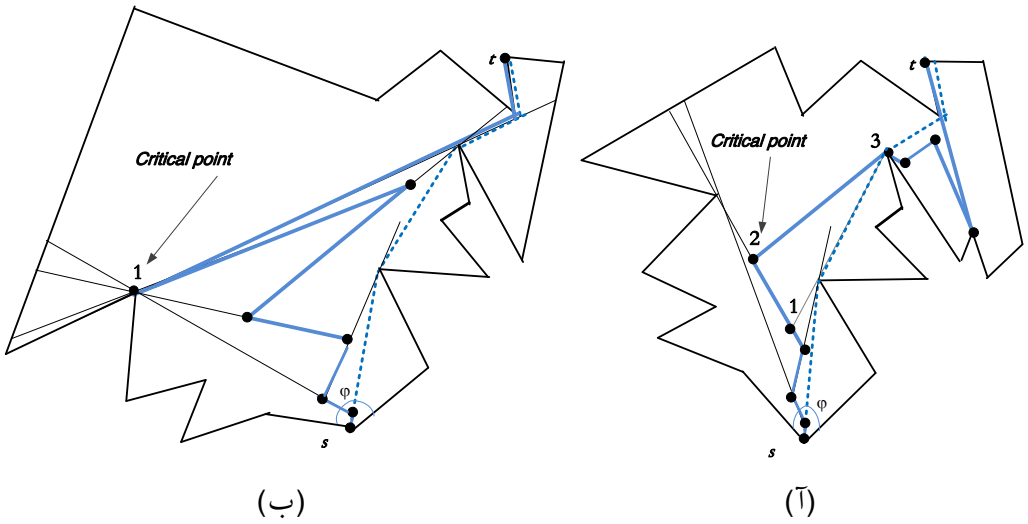
تمام آنچه ربات در طول مسیر حرکت نگه می‌دارد مکان g_r و g_l است. در طول مسیر حرکت ربات همان رخدادهای اساسی که باعث تغییر ناحیه رویت‌پذیری آن می‌شود، مکان g_r و g_l را می‌تواند عوض کند. همچنین یک وضعیت قیف ممکن است پایان یابد یا اینکه یک قیف جدید شروع شود. به نقطه‌ای که در آن وضعیت قیف تغییر می‌کند، نقطه بحرانی[۴] گفته می‌شود. رخدادهایی که در ادامه آمده است، این تغییرات را نشان می‌دهد.

- وقتی که g_r (g_l) تقسیم می‌شود به g_r (g_l) و یک شکاف راست (شکاف چپ)، g_r (g_l) با آن شکاف راست (شکاف چپ) جایگزین می‌شود. نقطه ۱ در شکل ۱.۷آ را ببینید.

[۴] Critical Point

- وقتی که g_r (g_l) تقسیم می‌شود به g_r (g_l) و یک شکاف چپ (شکاف راست)، g_l (g_r) با آن شکاف چپ (شکاف راست) جایگزین می‌شود. نقطه ۲ در شکل ۱.۷آ را ببینید.
- هرگاه رخداد ناپدید شدن g_r (g_l) پیش آید، ربات ممکن است در نقطه ای باشد که یک وضعیت قیف پایان می‌پذیرد. نقطه ۱ در شکل ۱.۷ب را ببینید.

توجه کنید که رخدادهای فوق ممکن است همزمان پیش آیند. علاوه بر این با حرکت به سمت g_r یا g_l این شکاف‌ها با شکاف دیگری ادغام نمی‌شوند.



(ب) (آ)

شکل ۱.۷: نمایش مسیر حرکت ربات در دو حالت زاویه باز کوچک و زاویه باز بزرگ.

۳.۷ الگوریتم

اکنون الگوریتم را بر اساس اطلاعاتی که در مورد g_r و g_l داریم، شرح می‌دهیم.

۱.۳.۷ الگوریتم قطعی

در هر نقطه از مسیر حرکت ربات به خصوص، در نقطه شروع دو حالت پیش می‌آید.

- وقتی که فقط یکی از g_r یا g_r وجود دارد (شکل ۱.۵آ)، یا اینکه هر دو با هم در یک راستا قرار دارند. در این حالت ربات به سمت آن شکاف حرکت می‌کند تا وقتی که هدف به دست آید یا اینکه وضعیت قیف ایجاد شود.

- در یک وضعیت قیف برای پرهیز از انحراف از کوتاه‌ترین مسیر رویات به صورت تناوبی به سمت g_r و g_l حرکت می‌کند (شکل ۱.۷). در هر مرحله $i \in \{۱, ۳, ۵, ...\}$، ربات a_i گام به سمت g_r حرکت می‌کند و در هر مرحله $i \in \{۲, ۴, ۶, ...\}$، ربات a_i به سمت g_l حرکت می‌کند به طوری که $a_۱ = ۱$، $a_۲ = ۳$ و $a_i = ۲a_{i-۱}$ برای $i \geq ۳$. ربات به حرکت تناوبی خود ادامه می‌دهد تا اینکه به نقطه‌ای برسد که در آن وضعیت قیف تمام شده است یا اینکه هر دوی g_r و g_l در یک راستا قرار بگیرند که حالت قبل پیش می‌آید. نقطه ۲ در شکل ۱.۷آ را ببینید.

فرایند فوق ادامه پیدا می‌کند تا اینکه ربات به نقطه هدف برخورد کند، همان‌طور که در شکل ۱.۷ نمایش داده شده است.

۲.۳.۷ الگوریتم تصادفی

در هر نقطه از مسیر حرکت که فقط یکی از g_r و g_l وجود دارند، ربات به سمت آن حرکت می‌کند. در وضعیت قیف، ربات یک متغیر تصادفی حقیقی ε از بازه $(۰, ۱]$ انتخاب می‌کند و طول گام ربات را با ۲^{ε} تنظیم می‌کند. سپس برای انتخاب جهت حرکت، یک متغیر تصادفی X به طور یکنوا از مجموعه $\{۰, ۱\}$ انتخاب می‌کند. اگر متغیر تصادفی انتخاب شده ۱ (۰) باشد، ربات در هر گام زوج، a_i گام به سمت $g_l(g_r)$ حرکت می کند و در هر گام فرد، a_i گام به سمت $g_r(g_l)$ حرکت می‌کند. مانند حالت قطعی ربات به حرکت تناوبی خود به سمت شکاف‌های راست و چپ ادامه می‌دهد تا اینکه به نقطه‌ای می‌رسد که وضیت قیف به پایان می‌رسد. هر بار که وضعیت قیف ایجاد می‌شود، ربات با استفاده از متغیرهای تصادفی، طول گام و جهت‌های حرکت در گام‌های زوج و فرد را انتخاب می‌کند. فرآیند فوق ادامه پیدا می‌کند تا اینکه به هدف برسد.

۳.۳.۷ تحلیل الگوریتم

در طی فرایند جستجو، مسیر حرکت ربات با کوتاه‌ترین مسیر منطبق است تا اینکه وضعیت قیف ایجاد شود. بنابراین برای اینکه طول مسیر حرکت ربات را با طول کوتاه‌ترین مسیر مقایسه کنیم، آن‌ها را در یک وضعیت قیف با هم مقایسه می‌کنیم. برای این تحلیل از راهبرد دوگانه‌ای که بایزا و همکارنش [۴] برای جستجو روی خط استفاده کردند، ایده می‌گیریم. در آن راهبرد ربات برای پیدا کردن نقطه‌ای از خط به سمت عقب و جلو حرکت می‌کند به طوری که فاصله‌اش را تا نقطه شروع هر بار دو برابر می‌کند

تا اینکه به هدف برسد.

قضیه ۱.۷ [۴] جستجوی دوگانه برای پیدا کردن نقطه‌ای از خط دارای ضریب رقابتی ۹ است و این ضریب رقابتی بهینه است.

زاویه بین g_l یا g_r زاویه بازه[۵] نامیده می‌شود که اندازه آن از ۱۸۰ درجه بیشتر نمی‌شود. یک خاصیت مهم آن که ما برای اثبات ضریب رقابتی از آن استفاده کرده‌ایم، در لم زیر آمده است.

لم ۲.۷ در راهبرد رقابتی ارائه شده ما، انحراف از کوتاه‌ترین مسیر، برای زاویه‌های باز کوچک‌تر کمتر از انحراف برای زاویه‌های باز بزرگ‌تر است.

اثبات.

در راهبرد ما، ربات گام‌هایی را به سمت راست و گام‌هایی را به سمت چپ حرکت می‌کند. یکی از حرکت‌های تناوبی فوق حرکت در امتداد صحیح و دیگری انحراف از کوتاه‌ترین مسیر است. واضح است میزان این انحراف برای زاویه‌های بزرگ بیشتر است، همان‌طور که در شکل ۱.۷ نمایش داده شده است . ■

اکنون می‌توانیم ضریب رقابتی راهبرد ارائه شده را محاسبه کنیم.

قضیه ۳.۷ راهبرد قطعی ما مسیری برای جستجوی ربات تولید می‌کند که طول آن حداکثر ۹ برابر طول کوتاه‌ترین مسیر است. علاوه بر این، مسیر تولید شده یک مسیر بهینه است.

اثبات. ربات همواره به سمت یکی از جهت‌های g_l یا g_r حرکت می‌کند در حالی که آن‌ها با رخداد تقسیم شدن به‌روز می‌شوند. تعداد این رخدادها متناهی است. بنابراین ربات در زمان متناهی به هدف می‌رسد. با توجه به لم ۲.۷ می‌دانیم که انحراف برای زاوایای بزرگ‌تر بیشتر است. بنابراین برای به دست آوردن ضریب رقابتی فرض می‌کنیم، زاویه باز بیشترین مقدار خود را دارد، یعنی ۱۸۰ درجه است. می‌توان نشان داد که گام‌های انتخابی ما برای زاویه ۱۸۰ درجه همان گام‌های راهبرد دوگانه معروف برای جستجوی دوگانه روی خط است. این نشان می‌دهد، ضریب رقابتی راهبرد ما برای جستجو در خیابان ۹ است. ■

[۵] Opening Angle

کاو ریف و تات [۳۱] برای جستجو روی یک خط یک الگوریتم تصادفی ارائه دادند. در آن الگوریتم گام اول توسط یک متغیر تصادفی انتخاب می‌شد و در گام‌های بعدی تعداد گام‌ها هر بار در r ضرب می‌شد. اثبات می‌شود که ضریب رقابتی مورد انتظار الگوریتم فوق $1 + (1 + r)/\ln r$ است. بحثی شبیه به اثبات قضیه ۳.۷ نشان می‌دهد که راهبرد تصادفی ما برای زاویه ۱۸۰ درجه در خیابان مسیری منطبق بر مسیر تولید شده توسط راهبرد کاو و همکارانش با $r = 2$ تولید می‌کند. بحث اخیر قضیه زیر را ثابت می‌کند.

قضیه ۴.۷ راهبرد تصادفی ما مسیری برای جستجوی ربات تولید می‌کند که طول مورد انتظار آن حداکثر ۵/۳۳ برابر طول کوتاه‌ترین مسیر است.

۴.۷ جمع‌بندی

در این فصل برای هدایت ربات در یک خیابان راهبردی ارائه شد که ربات بدون نگهداری اطلاعات محیط به هدفش می‌رسد. ربات فقط بر اساس اطلاعات محلی که از طریق حسگرش به دست می‌آورد، یک مسیر بهینه را که طول آن ۹ برابر طول کوتاه‌ترین مسیر است، برای رسیدن به هدف طی می‌کند. بر اساس راهبرد ارائه شده فوق، یک الگوریتم تصادفی ارائه دادیم که ضریب رقابتی مورد انتظار آن ۵/۳۳ است.

فصل ۸

راهبرد رقابتی برای هدایت ربات در یک خیابان تعمیم یافته

در این فصل مسئله هدایت ربات ساده در خیابان تعمیم یافته[۱] مورد بررسی قرار می‌گیرد [۱۰، ۱۱]. ربات پایه همان ربات ساده مجهز به حسگر شکاف است که الگوریتم‌هایی برای هدایت آن در خیابان در فصل‌های قبل شرح داده شد. در این بخش ربات فوق را با اضافه کردن یک قطب‌نما تقویت کرده‌ایم. ربات تقویت شده قادر است جهت‌های شمال جنوب و شرق و غرب را تشخیص دهد و در امتداد آن‌ها حرکت کند. یک راهبرد رقابتی برای حرکت ربات در یک خیابان تعمیم یافته متعامد ارائه می‌شود. ربات با استفاده از اطلاعات محلی که از طریق حسگرهایش به دست می‌آورد و همچنین با استفاده از تعدادی نشانه‌گذار، محدوده را جستجو می‌کند تا با طی کردن یک مسیر متعامد به نقطه هدف برسد. طول مسیری که ربات برای رسیدن به هدف طی می‌کند، حداکثر ۹/۰۶ برابر طول کوتاه‌ترین مسیر است. ضریب رقابتی ارائه شده یک ضریب بهینه است [۵۹].

[۱] Generalized Street

۱.۸ مفاهیم مورد نیاز

۱.۱.۸ محدوده کاری

در این بخش محدوده کاری، کلاسی از چندضلعی‌ها به نام چندضلعی خیابان تعمیم یافته است که همه چندضلعی‌های خیابان را شامل می‌شود. در این بخش برخی از تعاریف و مفاهیم مربوط به آن را به اختصار شرح می‌دهیم.

در یک چندضلعی ساده با دو راس مجزای s و t، زنجیر چندضلعی ساعتگرد از s به t را زنجیر چپ و زنجیر چندضلعی پادساعتگرد از s به t را زنجیر راست می‌نامند. همان‌طور که در قسمت‌های قبل شرح داده شد، در یک چندضلعی خیابان هر نقطه از روی زنجیر راست حداقل از یک نقطه روی زنجیر چپ قابل دید است و برعکس.

تعریف ۱.۸ هر خط افقی در درون چندضلعی که هر دو راس آن روی اضلاع چندضلعی باشد یک وتر[۲] نامیده می‌شود. چنانچه یک راس از وتر روی زنجیر راست و یک راس آن روی زنجیر چپ باشد یک LR–وتر است.

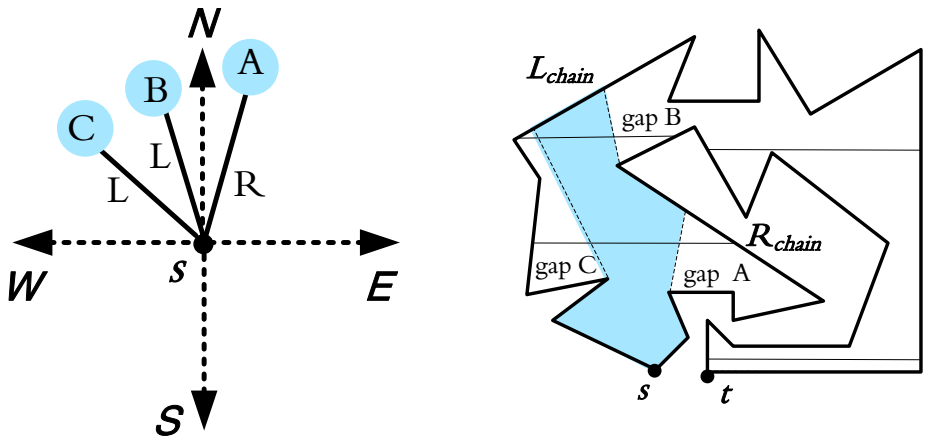
تعریف ۲.۸ یک چندضلعی ساده یک خیابان تعمیم یافته است در صورتی که هر نقطه روی مرز چندضلعی از روی حداقل یک LR–وتر دیده شود.

در شکل ۱.۸ چندضلعی خیابان و تعدادی از وترهای آن نمایش داده شده است.

۲.۱.۸ مدل حسگر و محدودیت حرکتی ربات

ربات پایه همان ربات مجهز به حسگر شکاف است. ربات از نقطه s در خیابان شروع به حرکت می‌کند و هدفش رسیدن به نقطه t است. با حرکت ربات همان‌طور که قبلا شرح داده شد وضعیت شکاف‌ها تغییر می‌کند. ربات قادر به حرکت به سمت شکاف‌ها است در حالی که رخدادهایی را که وضعیت شکاف‌ها را تغییر می‌دهد، تشخیص می‌دهد. اکنون ربات با استفاده از یک قطب‌نما تقویت شده [۱۴] و قادر است جهت‌های جغرافیایی را تشخیص دهد و در امتداد آن‌ها حرکت کند، شکل ۱.۸ را ببینید. علاوه بر این، تعدادی نشانه‌گذار همراه خود دارد که قادر است با استفاده آن‌ها هر جایی از محیط را علامت‌گذاری کند.

[۲] Chord



(ب) (آ)

شکل ۱.۸: خیابان تعمیم یافته. (آ) برخی از وترها در خیابان تعمیم یافته. (ب) آنچه ربات در نقطه شروع از محیط درک می‌کند.

۲.۸ راهبرد اصلی برای هدایت ربات در یک خیابان تعمیم یافته متعامد

اکنون برای هدایت ربات در یک خیابان تعمیم یافته که همه اضلاع آن به صورت افقی یا عمودی هستند یک راهبرد برخط ارائه می‌دهیم.

برخلاف راهبردهای قبلی ارائه شده برای هدایت ربات، در این جا ربات نقشه محدوده‌ای را که تاکنون دیده است، نگه نمی‌دارد. فقط مکان شکاف‌های فعلی و جهت‌های جغرافیایی در دسترس ربات است. متناظر با هر شکاف یک راس انعکاسی وجود دارد. به وتری که از این شکاف عبور می‌کند وتر شکاف[۳] می‌گوییم. هر وتر چندضلعی را به دو یا سه ناحیه تقسیم می‌کند، همان‌طور که در شکل ۲.۸ نمایش داده شده است. همچنین یالی از چندضلعی را که در امتداد یک وتر شکاف است، یک یال شکاف[۴] می‌نامیم.

تعریف ۳.۸ یک شکاف، شکاف اساسی نامیده می‌شود چنانچه هدف پشت وتر شکاف بعد از آن قرار داشته باشد. اگر این شکاف راست باشد آن را با $R-EG$ و اگر یک شکاف چپ باشد با $L-EG$ نمایش می‌دهیم.

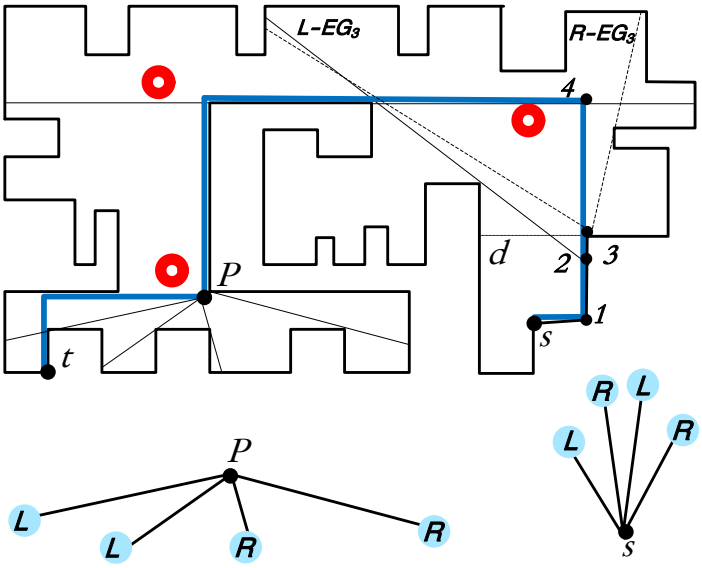
مثال‌هایی از شکاف اساسی، وتر شکاف و یال شکاف در شکل ۲.۸ قرار داده شده است. از تعریف خیابان تعمیم یافته لم زیر به طور مستقیم نتیجه می‌شود.

[۳] Gap Chord

[۴] Gap Edge

لم ۴.۸ در نقطه شروع حرکت، یا نقطه t قابل دید است، یا حداقل یک شکاف اساسی وجود دارد.

نکته اساسی در این جا این است که چگونه ربات ساده ما می‌تواند تشخیص دهد، کدام شکاف از بین شکاف‌های موجود یک شکاف اساسی است.



شکل ۲.۸: یک خیابان تعمیم یافته متعامد و مکان شکاف‌ها در دو نقطه s و p.

قضیه ۵.۸ وقتی که ربات در حال حرکت در داخل چندضلعی تعمیم یافته است، به محض اینکه به نقطه‌ای برسد که در آن نقطه، شکاف هم‌راستا با جهت‌های شمال یا جنوب قطب‌نما وجود دارد، یک شکاف اساسی را گزارش می‌دهد.

اثبات. فرض کنید که خط d وتر شکاف مربوط به آن شکاف باشد، این وتر چندضلعی را به دو یا سه ناحیه تقسیم می‌کند. نقاط s و t باید در دو ناحیه مختلف باشد، چنانچه این دو نقطه در یک ناحیه باشند نقاط پشت شکاف توسط هیچ LR–وتر قابل دید نیست. بنابراین نقطه‌ی هدف پشت این شکاف قرار دارد. پس این شکاف عمودی، یک شکاف اساسی است که ربات می‌تواند تشخیص دهد. نقطه ۱ در شکل ۲.۸ را ببینید. ■

به نقطه‌ای که در آن، یک شکاف اساسی توسط ربات گزارش می‌شود، نقطه‌ی چرخش[۵] می‌گوییم. توجه کنید که در این نقطه یک شکاف عمودی دیده شده است.

[۵] Turn Point

قضیه ۶.۸ اگر شکاف گزارش شده در نقطه‌ی چرخش یک $R-EG$ باشد، شکاف چپی که بلافاصله پادساعتگرد بعد از آن قرار دارد $L-EG$ است. اگر شکاف گزارش شده در نقطه چرخش یک $L-EG$ باشد، شکاف راستی که بلافاصله ساعتگرد بعد از آن قرار دارد $R-EG$ است.

اثبات. فرض کنید که در نقطه‌ی چرخش یک $R-EG$ کشف شود و شکاف چپی بلافاصله پادساعتگرد بعد از آن قرار داشته باشد. با حرکت ربات به سمت چپ، آن شکاف به یک شکاف عمودی تبدیل می‌شود بنابراین آن شکاف چپ، $L-EG$ بوده است. برای حالتی که در نقطه‌ی چرخش $L-EG$ کشف شود به همین ترتیب اثبات می‌شود. ■

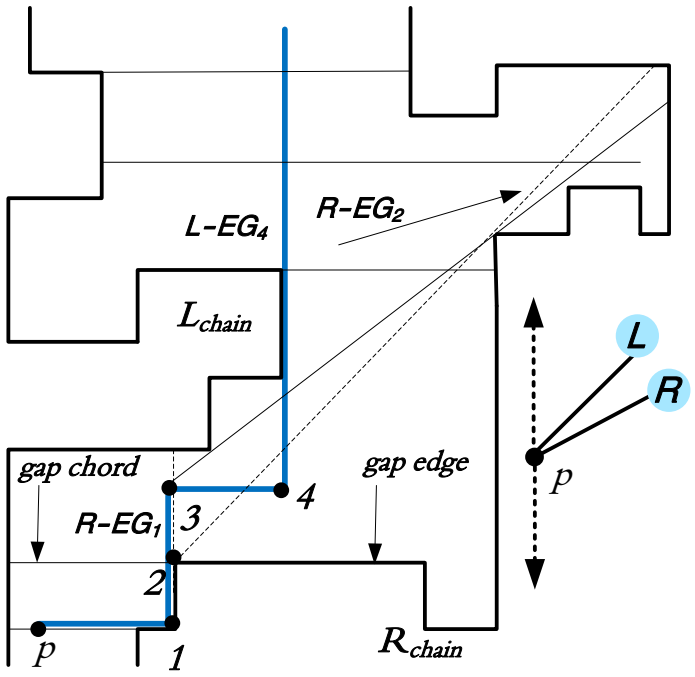
اکنون با استفاده از مطالب فوق می‌توانیم الگوریتم را شرح دهیم.

۱.۲.۸ الگوریتم

در نقطه شروع، حسگر ربات مکان شکاف‌های موجود را گزارش می‌دهد. سه حالت پیش می‌آید.

- همه شکاف‌ها در سمت راست خط قائم حاوی نقطه شروع باشند. با استفاده از قطب‌نما ربات به سمت راست حرکت می‌کند تا به یک نقطه چرخش برسد، شکل ۳.۸ رابببینید.
- همه شکاف‌ها در سمت چپ خط قائم حاوی نقطه شروع باشند. با استفاده از قطب‌نما ربات به سمت چپ حرکت می‌کند تا به یک نقطه چرخش برسد.
- شکاف‌ها در دو طرف خط عمودی حاوی نقطه شروع باشند. در این حالت ربات با اجرای فرایند جستجوی دوگانه (حرکت متناوب به سمت چپ و راست) نقطه چرخش را پیدا می‌کند، شکل ۲.۸ را ببینید.

بعد از پیدا کردن نقطه چرخش، با توجه به قضیه ۶.۸ ربات می‌تواند هر دو شکاف اساسی موجود را در صورت وجود تشخیص دهد. فرض کنید در نقطه چرخش، شکاف عمودی گزارش شده یک $R-EG$ باشد. ربات به سمت آن شکاف در راستای عمودی حرکت می‌کند در حالی که مکان $L-EG$ را هم نگه داشته است. در طول مسیر، $L-EG$ می‌تواند به‌روز شود (ناپدید شود یا تقسیم شود)، نقطه ۲ در شکل ۲.۸ را ببینید. ربات به حرکت عمودی خود در راستای $R-EG$ ادامه می‌دهد تا وقتی که $R-EG$ ناپدید شود. در لحظه رخداد اخیر ۳ حالت پیش می‌آید.



شکل ۳.۸: مسیر پیموده شده ربات بین دو وتر شکاف متوالی.

۱. هر دو شکاف $R-EG$ و $L-EG$ وجود داشته باشند (مانند نقطه ۳ در شکل ۲.۸)، در این حالت ربات به حرکت عمودی خود ادامه می‌دهد تا اینکه به نقطه‌ای برسد که هر دو شکاف فوق ناپدید شوند که حالت سوم است.

۲. یکی از دو شکاف فوق وجود دارد (مانند نقطه ۲ در شکل ۳.۸). ربات به حرکت در امتداد کنونی ادامه می‌دهد تا اینکه شکاف فوق با یک شکاف دیگر ادغام شود یا اینکه ناپدید شود که همان حالت سوم است. در صورت ادغام شدن، ربات در امتداد افقی در ناحیه‌ای که حاوی شکاف‌ها بود، حرکت می‌کند تا اینکه در یک نقطه‌ی چرخش یک شکاف اساسی قائم را بیابد (نقطه ۳ در شکل ۳.۸).

۳. شکاف اساسی وجود نداشته باشد (نقطه ۴ در شکل ۲.۸). ربات در ناحیه‌ای که از آن آمده است یک نشانه‌گذار قرار می‌دهد و فرایند جستجوی دوگانه را برای پیدا کردن نقطه چرخش که در آن یک شکاف اساسی قائم دیده می‌شود، به کار می‌گیرد. توجه کنید که شکاف اساسی گزارش شده باید در ناحیه‌ای باشد که در آن نشانه‌گذار وجود نداشته باشد.

توجه کنید یک حالت خاص زمانی رخ می‌دهد که ربات بعد از پیدا کردن اولین نقطه‌ی چرخش بخواهد بدون هیچ حرکت افقی دوباره فرایند جستجوی دوگانه را انجام دهد که در این حالت

همان جستجوی دوگانه قبلی را از سر می‌گیرد.

فرایند فوق ادامه پیدا می‌کند تا ربات به نقطه‌ای برسد که در آن نقطه‌ی هدف t قابل دید باشد. ربات با استفاده از قطب‌نما در امتداد یک مسیر افقی عمودی به آن می‌رسد.

۲.۲.۸ تحلیل الگوریتم

اکنون نشان می‌دهیم ربات ساده ما با استفاده از اطلاعات محلی که از طریق حسگرهایش به دست می‌آورد در امتداد مسیری که طول آن قابل مقایسه با طول کوتاه‌ترین مسیر است به هدف می‌رسد. اگر چه ربات ما قویا از ربات داتا و ایکینگ [۱۰] ضعیف‌تر است و مسیر جستجوی آن‌ها متفاوت است ولی ضریب رقابتی هر دو راهبرد یکی است.

قضیه ۷.۸ راهبرد ما برای هدایت ربات در حالی که یک مسیر جستجو از s به t تولید کرده است، خاتمه می‌یابد.

اثبات. در نقطه شروع، اگر نقطه‌ی هدف قابل دید نباشد قطعا پشت یکی از شکاف‌های موجود مخفی است. ربات در امتداد افقی برای پیدا کردن یک شکاف اساسی جستجو می‌کند. سپس در امتداد شکاف عمودی که یک راستای درست هم است، حرکت می‌کند. ربات به حرکت در این راستای عمودی درست ادامه می‌دهد تا اینکه هر دو شکاف اساسی ناپدید شوند. دوباره به جستجوی دوگانه برای یافتن یک شکاف اساسی دیگر می‌پردازد. دوباره یک جهت عمودی درست انتخاب می‌کند که یک گام به هدف نزدیک‌تر می‌شود. بنابراین فرایند خاتمه‌پذیر است.

■

قضیه ۸.۸ [۴۴] کران پایین ضریب رقابتی برای یک راهبرد جستجو در یک خیابان تعمیم یافته ۹/۰۶ است.

قضیه ۹.۸ ضریب رقابتی برای راهبرد جستجوی ما در یک خیابان تعمیم یافته ۹/۰۶ است. این ضریب بهینه است.

اثبات. برای مقایسه طول مسیر حرکت ربات با طول کوتاه‌ترین مسیر، ابتدا آن را با کوتاه‌ترین مسیر با متریک L_1 مقایسه می‌کنیم. همان‌طور که در الگوریتم ذکر شد، ربات ما همواره یک جهت عمودی

درست را برای حرکت انتخاب می‌کند. ربات برای تشخیص یک شکاف اساسی یا در جهت درست حرکت می‌کند یا اینکه فرایند جستجوی دوگانه را اجرا می‌کند. پس مسیر افقی طی شده توسط ربات حداکثر ۹ برابر طول کوتاه‌ترین مسیر است. پس جستجو در یک خیابان تعمیم یافته حداقل به سختی جستجو روی خط است. بنابراین ضریب رقابتی در متریک L_1، ۹ است. فاصله بین دو نقطه چرخش متوالی در متریک L_2 برابراست با $\sqrt{x_1^2 + y_1^2}$. با راهبرد ما طول مسیری که ربات طی می‌کند $9x_1 + y_1$ است. مقدار بشینه $\frac{9x_1+y_1}{\sqrt{x_1^2+y_1^2}}$ برابراست با ۹/۰۶. قضیه ۸.۸ نشان می‌دهد که راهبرد ما بهینه است. ■

۳.۸ جمع‌بندی

در این فصل ربات ساده مجهز به حسگر شکاف را تقویت کردیم و از آن برای پیمایش چندضلعی خیابان تعمیم یافته استفاده کردیم. در حالتی که چندضلعی خیابان تعمیم یافته متعامد است یک الگوریتم با ضریب رقابتی ۹/۰۶ ارائه دادیم.

فصل ۹

نتیجه‌گیری و پیشنهادهایی برای ادامه کار

در این رساله، الگوریتم‌های مسیریابی در محیط‌های ناشناخته مطرح شدند. به عبارت دیگر ربات در محیطی که نقشه‌ای از آن در دسترس نیست، حرکت می‌کند و با استفاده از اطلاعاتی که توسط حسگرهایش از محیط به‌دست می‌آورد مسیری را برای رسیدن به هدف پیدا می‌کند. آنچه این پژوهش را از پژوهش‌های قبلی متمایز می‌کند، توجه به قدرت حسگرهای ربات است. از آنجایی که حسگرهای محدود هم از نظر هزینه و هم از جنبه انعطاف پذیری (از نظر قابل استفاده بودن در محیط‌های متنوع) نسبت به حسگرهای قوی بهتر عمل می‌کنند، در این پژوهش هم سعی می‌شود از حسگرهای محدود استفاده شود. این حسگرها ویژگی‌های هندسی محدوده کاری مثل زاویه و اندازه را درک نمی‌کنند و اغلب ویژگی‌های ترکیبیاتی محیط را گزارش می‌دهند.

۱.۹ خلاصه‌ای از نتایج بدست آمده

در این پژوهش بسیاری مسائل مسیریابی در محیط ناشناخته که قبلا برای ربات‌های با دید کامل طرح شده بود، برای ربات‌های ساده، بررسی شده است. در هر یک از مسائل میزان تاثیر محدودیت درک ربات از محیط، روی مسیری که طی می‌کند، تحلیل شده است. این تحلیل توسط مفهوم ضریب رقابتی انجام شده است. در جدول زیر، میزان انحراف از کوتاه‌ترین مسیر بر اساس قدرت حسگر ربات، در مسائل مختلف نمایش داده شده است. در هر مسئله، بهترین ضریب رقابتی برای حالتی که از ربات با حسگر دید کامل استفاده شود، نیز در جدول آمده است.

جدول ۱.۹: نتایج به دست آمده برای مسئله‌های مختلف و مقایسه با بهترین نتایج قبلی برای حسگر دید کامل

مسئله	ضریب رقابتی با حسگر دید کامل	ضریب رقابتی با حسگر محدود	نوع حسگر محدود
هدایت ربات در خیابان	$\sqrt{۲}$	۱۱ و ۹	شکاف و نشانه‌گذار
هدایت ربات در خیابان متعامد	$\sqrt{۲}$	$\sqrt{۲}$	شکاف و قطب‌نما
هدایت ربات در خیابان در امتداد مسیر با کمترین تعداد چرخش	$۲ - ۱/m$	$۲ - ۱/m$	شکاف
یافتن هسته در چندضلعی ستاره‌ای	۳/۱۲۶	۱۲/۹۲	شکاف و قطب‌نما
هدایت دو ربات	—	۲	شکاف
هدایت در خیابان با استفاده از اطلاعات محلی	—	۹	شکاف
راهبرد تصادفی هدایت ربات در خیابان با استفاده از اطلاعات محلی	—	۵/۳۳	شکاف
هدایت ربات در خیابان تعمیم‌یافته	۹/۰۶	۹/۰۶	شکاف، قطب‌نما و نشانه‌گذار

۱.۱.۹ مزایا و معایب الگوریتم‌های ارائه شده

عمده مزیت این پژوهش استفاده از ربات ساده با حسگر محدود در مسیریابی است. در پژوهش قبلی که ربات ساده به کار گرفته شده بود، از درخت پیمایش شکاف که در فصل ۳.۳ برای شناسایی محیط‌های ناشناخته معرفی شده است، استفاده می‌شود. این اشکال را می‌توان به آن‌ها وارد کرد که ربات برای ساخت چنین درختی مجبور است، مسیر بسیار طولانی طی کند و در کل محدوده کاری حرکت کند. علاوه بر این، اطلاعات کل محدوده کاری در درخت نگهداری می‌شود که نیاز به حافظه دارد. در صورتی که برای بسیاری از اهداف از جمله مسئله هدایت ربات در خیابان نیازی به شناسایی کل محیط نیست. در الگوریتم ارائه شده ما در فصل ۴، ربات فقط به نواحی که احتمال وجود هدف است سر می‌زند و فقط همان نواحی را در S-GNT ذخیره می‌کند. مسیری که ربات برای رسیدن به هدف طی می‌کند، دارای انحراف محدود شده از کوتاه‌ترین مسیر است. در فصل‌های بعدی، ربات فقط از اطلاعات محلی برای مسیریابی استفاده می‌کند. این در حالی است که در همه پژوهش‌های قبلی که ربات دارای حسگر دید کامل است، کل محدوده‌ای که ربات تاکنون دیده است، ذخیره می‌شود و در مسیریابی مورد استفاده قرار می‌گیرد.

اشکالی که به استفاده از حسگرهای محدود می‌توان وارد کرد، این است که به هر حال ربات برای رسیدن به هدف، کوتاه‌ترین مسیر را طی نمی‌کند. علاوه بر این، برای این‌که بتوان از این روش برای مسیریابی در چندضلعی‌های عمومی‌تر استفاده کرد باید آن را متناسب با آن چندضلعی‌ها اصلاح کرد.

۲.۹ پیشنهادها

با توجه به عوامل دخیل در مسئله مسیریابی در محیط ناشناخته، ما حیطه‌هایی که قابلیت ادامه کار در آن‌ها وجود دارد را به سه دسته اصلی تقسیم می‌کنیم.

۱.۲.۹ توسعه محدوده کاری

ما برای چندضلعی خیابان، چندضلعی ستاره‌ای و چندضلعی خیابان تعمیم یافته در حالت متعامد راهبردهای رقابتی ارائه دادیم. گسترش محیط کاری به چندضلعی‌های دیگر یا قرار دادن حفره‌هایی درون محیط‌های بررسی شده، فرصتهایی برای ادامه این پژوهش هستند.

با تامل در الگوریتم‌های پیشنهادی در این پژوهش می‌توان دریافت که الگوریتم‌ها بر اساس رخداد کار می‌کنند. چنانچه بتوانیم تغییری در شکل چندضلعی‌ها ایجاد کنیم به گونه‌ای که رخدادی که باعث تغییر مسیر حرکت ربات شود ایجاد نشود، می‌توان چندضلعی‌هایی را به دست آورد که الگوریتم‌های پیشنهادی در این رساله، در آن چندضلعی‌ها نیز مسیر رسیدن به هدف برای ربات تولید کند. بسیاری از روش‌های ارائه شده براساس خواص خیابان به خصوص وجود قیف است. اگر چندضلعی را به گونه‌ای تغییر دهیم که ساختار قیف در آن‌ها تغییر نکند، می‌توان تحت شرایطی انتظار داشت که الگوریتم‌های پیشنهادی جوابگو باشند.

۲.۲.۹ تغییر مدل حسگر

ما از ربات مجهز به حسگر شکاف استفاده کردیم. همچنین ربات را با مجهز کردن به یک قطب‌نما تقویت کردیم و توانستیم نتایج بهتری بگیریم. تعریف حسگرهای محدود جدید و ترکیب حسگرها با هم نیز یک فرصت دیگر برای پژوهش در این حوزه است.

۳.۲.۹ محدودیت‌های حرکتی برای ربات و تغییر شکل ربات

رباتی که ما در این پژوهش استفاده کردیم، می‌تواند در هر لحظه تغییر جهت حرکت بدهد و به اندازه دلخواه بچرخد. استفاده از رباتی که محدودیت در میزان چرخش خود دارد کابردی‌تر است. استفاده از چنین ربات‌هایی نیز مجالی برای ادامه این پژوهش است. در سرتاسر این پژوهش فرض بر استفاده از یک ربات نقطه‌ای است در حالی‌که استفاده از ربات با شکل‌های دیگر کاربردی‌تر است.

۳.۹ مسئله‌های باز مشابه

علاوه بر مسیریابی در محیط ناشناخته، ربات می‌تواند با استفاده از اطلاعاتی که از طریق حسگرهایش به دست می‌آورد به استنتاج در مورد مشخصات محیط از نظر هندسی، یا تعداد و چگونگی پراکندگی اشیا در محیط بپردازد. نمونه‌هایی از این قبیل مسئله‌ها در فصل ۳.۳ معرفی شدند. به عنوان مثال، یک ربات پس از عبور کردن از روی یک مسیر خاص از پیش تعریف شده به استنتاج در مورد حداقل و حداکثر تعداد اشیایی که در محیط شناسایی نشده‌اند، می‌پردازد. از آنجایی که مسیر ربات از قبل مشخص است

این مسئله به یک مسئله غیرفعال[۱] معروف است. حالت فعال[۲] این مسئله یعنی طرح مسیری که بتوان با حرکت روی آن به استنتاج مطلوب در مورد محیط رسید، یک مسئله باز در این زمینه است.

مسئله جالب دیگری که می‌توان در این رابطه معرفی کرد این است که اگر از دو یا چند ربات برای شناسایی محیط استفاده شود، به چه نتایجی می‌توان رسید. در راهبرد ارائه شده در [۶۵]، برای استنتاج در مورد چگونگی توزیع موجودات رویت نشده توسط ربات از یک گراف دوبخشی استفاده می‌شود. یک ایده برای به دست آوردن اطلاعات در مورد پراکندگی موجوداتی که توسط هیچ یک از ربات‌ها در محیط دیده نشده‌اند، ساخت گراف دوبخشی مربوط به مسیر حرکت هر یک از ربات‌ها و سپس ترکیب اطلاعات موجود در گراف‌ها است به گونه‌ای که بتوان به اطلاعات کلی در محیط دست یافت.

[۱] Passive

[۲] Active

کتاب‌نامه

[1] Abouei Mehrizi, Mohammad, Mohammad Ghodsi, and Azadeh Tabatabaei. “Robots’ Cooperation for Finding a Target in Streets.” In International Conference on Topics in Theoretical Computer Science (TTCS). Springer International Publishing, (2015): 30-43.

[2] Albers, Susanne, Klaus Kursawe, and Sven Schuierer. “Exploring unknown environments with obstacles.” Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, (1999).

[3] Amzajerdian, Farzin; Pierrottet, Diego F.; Petway, Larry B.; Hines, Glenn D.; Roback, Vincent E. “Lidar Systems for Precision Navigation and Safe Landing on Planetary Bodies”. Langley Research Center. NTRS. Retrieved May 24, (2011).

[4] Baezayates, Ricardo A., Joseph C. Culberson, and Gregory JE Rawlins. “Searching in the plane.” Information and Computation, 106.2 (1993): 234-252.

[5] Berman, Piotr. “On-line searching and navigation.” Online Algorithms (1998): 232-241.

[6] Blum, Avrim, Prabhakar Raghavan, and Baruch Schieber. “Navigating in unfamiliar geometric terrain.” SIAM Journal on Computing 26.1 (1997): 110-137.

[7] Bose, Prosenjit, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. “Competitive local routing with constraints.” In International Symposium on Algorithms and Computation. Springer Berlin Heidelberg, (2015): 23-34.

[8] Brunner, Jan, Matúš Mihalák, Subhash Suri, Elias Vicari, and Peter Widmayer. “Simple robots in polygonal environments: A hierarchy.” In International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics. Springer Berlin Heidelberg, (2008): 111-124.

[9] Das, Gautam, Paul J. Heffernan, and Giri Narasimhan. “LR-visibility in polygons.” Computational Geometry 7.1 (1997): 37-57.

[10] Datta, Amitava, and Christian Icking. “Competitive searching in a generalized street.” Proceedings of the tenth annual symposium on Computational geometry. ACM (1994).

[11] Datta, Amitava, Christoph Hipke, and Sven Schuierer. “Competitive searching in polygons—Beyond generalised streets.” Algorithms and Computations (1995): 32-41.

[12] Datta, Amitava, and Subbiah Soundaralakshmi. “Motion planning in an unknown polygonal environment with sounded performance guarantee.” Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on. Vol. 2. IEEE, (1999).

[13] Deng, Xiaotie, Tiko Kameda, and Christos Papadimitriou. “How to learn an unknown environment. I: the rectilinear case.” Journal of the ACM (JACM) 45.2 (1998): 215-245.

[14] Disser, Yann, Subir Kumar Ghosh, Matúš Mihalák, and Peter Widmayer “Mapping a polygon with holes using a compass.” Theoretical Computer Science 553 (2014): 106-113.

[15] Dudek, Gregory, Kathleen Romanik, and Sue Whitesides. “Localizing a robot with minimum travel.” SIAM Journal on Computing 27.2 (1998): 583-604.

[16] European Roadmap Smart Systems for Automated Driving. “European Technology Platform on Smart Systems Integration (EPoSS)”, (2015).

[17] Gfeller, Beat, Matúš Mihalák, Subhash Suri, Elias Vicari, and Peter Widmayer. “Counting targets with mobile sensors in an unknown environment.” Algorithmic Aspects of Wireless Sensor Networks (2008): 32-45.

[18] Google Self-Driving Car Project Monthly Report, March 2016. Google. Retrieved 23 March (2016).

[19] González-Baños, Héctor, Eric Mao, Jean-Claude Latombe. “Planning robot motion strategies for efficient model construction.” ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM (2000): 345-352.

[20] Guibas, Leonidas J., Rajeev Motwani, and Prabhakar Raghavan. “The robot localization problem.” SIAM Journal on Computing 26.4 (1997): 1120-1138.

[21] Guilamo, Luis, Benjamin Tovar, and Steven M. LaValle. “Pursuit-evasion in an unknown environment using gap navigation trees.” Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 4. IEEE, (2004).

[22] Ghosh, Subir Kumar. “Visibility algorithms in the plane.” Cambridge University Press (2007).

[23] Ghosh, Subir Kuma , and Sanjeev Saluja. “ Optimal on-line algorithms for walking with minimum number of turns in unknown streets.” Computational Geometry 8.5 (1997): 241-266.

[24] Ghosh, Subir Kumar. “Computing the visibility polygon from a convex set and related problems.” Journal of Algorithms 12.1 (1991): 75-95.

[25] Ghosh, Subir Kumar, and Rolf Klein. “Online algorithms for searching and exploration in the plane.” Computer Science Review 4.4 (2010): 189-201.

[26] Guizzo, Erico. “How google’s self-driving car works.” IEEE Spectrum Online, October 18 (2011).

[27] Hoffmann, Frank, Klaus Kriegel, Christian Icking, and Rolf Klein. “A competitive strategy for learning a polygon.” Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics (1997).

[28] Hoffmann, Frank, Christian Icking, Rolf Klein, and Klaus Kriegel. “The polygon exploration problem: A new strategy and a new analysis technique.” Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics: the algorithmic perspective: the algorithmic perspective. AK Peters, Ltd. (1998).

[29] Icking, Christian, Rolf Klein, and Elmar Langetepe. “An optimal competitive strategy for walking in streets.” STACS 99. Springer Berlin/Heidelberg (1999).

[30] Icking, Christian, and Rolf Klein. “Searching for the kernel of a polygon—a competitive strategy.” Proceedings of the Eleventh Annual Symposium on Computational geometry ACM (1995).

[31] Kao, Ming-Yang, John H. Reif, and Stephen R. Tate. “An optimal randomized algorithm for the cow-path problem.” Information and Computation 131.1 (1996): 63-79.

[32] Katsev, Max, Anna Yershova, Benjamin Tovar, Robert Ghrist, and Steven M. LaValle. “Mapping and pursuit-evasion strategies for a simple wall-following robot.” IEEE Transactions on Robotics 27.1 (2011): 113-128.

[33] Kessler, Aaron M. “Elon Musk Says Self-Driving Tesla Cars Will Be in the US by Summer.” The New York Times (2015): B1.

[34] Klein, Rolf. “Walking an unknown street with bounded detour.” Computational Geometry 1.6 (1992): 325-351.

[35] Kleinberg, Jon M. “On-line search in a simple polygon.” Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics (1994).

[36] Kostenko, A. A., A. I. Nosich, and I. A. Tishchenko. “Radar prehistory, Soviet side: three-coordinate L-band pulse radar developed in Ukraine in the late 30’s.” In Antennas and Propagation Society International Symposium, 2001. IEEE 4 (2001): 44-47.

[37] Latombe, Jean-Claude. “Robot motion planning.” Academic Publishers 25 (1991): 100-107.

[38] LaValle, Steven M. “Planning algorithms.” Cambridge University Press (2006).

[39] LaValle, Steven M. “Sensing and filtering: A tutorial based on preimages and information spaces.” Foundations and Trends in Robotics 1 (2012).

[40] Lee, Jae-Ha, Chan-Su Shin, Jae-Hoon Kim, Sung Yong Shin, and Kyung-Yong Chwa. “New competitive strategies for searching in unknown star-shaped polygons.” In Proceedings of the Thirteenth Annual Symposium on Computational geometry, ACM (1997): 427-429.

[41] Lewis, Jeremy S., and Jason O’Kane. “Planning for provably reliable navigation using an unreliable, nearly sensorless robot.” The International Journal of Robotics Research (2013): 0278364913488428.

[42] Lindemann, Stephen R., and Steven M. LaValle. “Simple and efficient algorithms for computing smooth, collision-free feedback laws over given cell decompositions.” The International Journal of Robotics Research 28.5 (2009): 600-621.

[43] López-Ortiz, Alejandro, and Sven Schuierer. “Simple, efficient and robust strategies to traverse streets.” Proc. 7th Canad. Conf. on Computational Geometry (1995).

[44] Lopez-Padilla, Rigoberto, Rafael Murrieta-Cid, and Steven M. LaValle. “Optimal gap navigation for a disc robot.” Algorithmic Foundations of Robotics X. Springer Berlin Heidelberg (2013): 123-138.

[45] Mitchell, Joseph SB. “Geometric shortest paths and network optimization.” Handbook of computational geometry 334 (2000): 633-702.

[46] O’Leary, Beth Laura; Darrin, Ann Garrison. Handbook of Space Engineering, Archaeology, and Heritage. Hoboken: CRC Press. (2009): 239–240.

[47] Palios, Leonidas. “A new competitive strategy for reaching the kernel of an unknown polygon.” In Scandinavian Workshop on Algorithm Theory. Springer Berlin Heidelberg (2000): 367-382.

[48] Park, Sang-Min, Jae-Ha Lee, and Kyung-Yong Chwa. “Visibility-based pursuit-evasion in a polygonal region by a searcher.” Automata, Languages and Programming (2001): 456-468.

[49] Pizarro, Daniel, Manuel Mazo, Enrique Santiso, Marta Marron, David Jimenez, Santiago Cobreces, and Cristina Losada. “Localization of mobile robots using odometry and an external vision sensor.” Sensors 10. 4 (2010): 3655-3680.

[50] Ramsey, John. “Self-driving cars to be tested on Virginia highways”. Richmond Times-Dispatch (1 June 2015).

[51] Rao, Nageswara SV, et al. “Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms.” Oak Ridge National Laboratory (1993).

[52] Shrivastava, Nisheeth, R. Mudumbai U. Madhow, and S. Suri. “Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms.” Proceedings of the 4th International Conference on Embedded Networked Sensor Systems. ACM (2006).

[53] Suri, Subhash, Elias Vicari, and Peter Widmayer. “Simple robots with minimal sensing: From local visibility to global geometry.” The International Journal of Robotics Research 27.9 (2008): 1055-1067.

[54] Tabatabaei, Azadeh, and Mohammad Ghodsi. “Walking in Streets with Minimal Sensing.” The 7th Annual International Conference on Combinatorial Optimization and Applications (COCOA’13). Springer International Publishing (2013): 361-372.

[55] Tabatabaei, Azadeh, and Mohammad Ghodsi. “Walking in Streets with Minimal Sensing.” J. Comb. Optim, 30.2 (2015): 387-402.

[56] Tabatabaei, Azadeh, and Mohammad Ghodsi. “Optimal Strategy for Walking in Streets with Minimum Number of Turns for a Simple Robot.” The 8th Annual International Conference on Combinatorial Optimization and Applications (COCOA’14). Springer International Publishing (2014): 101-112.

[57] Tabatabaei, Azadeh, and Mohammad Ghodsi. “A Randomized Strategy for Walking in Streets for a Simple Robot.” The 31th Annual European Workshop on Computational Geometry (EuroCG 2015), March 15-18, Ljubljana, Slovenia (2015).

[58] Tabatabaei, Azadeh, and Mohammad Ghodsi. “A Randomized Strategy for Walking in Streets for a Simple Robot Using Local Information .” Submitted to IPL (2016).

[59] Tabatabaei, Azadeh, Mohammad Ghodsi, and Fardin Shapouri. “A Competitive Strategy for Walking in Generalized Streets for a Simple Robot.” The 28th Canadian Conference on Computational Geometry (2016).

[60] Teller, Seth: “Development of a self-driving car as a mobile sensing platform.” In: CPS Week Keynote Presentation at the International Conference on Information Processing in Sensor Networks (IPSN) (April 2008).

[61] Tovar, Benjamin, Fred Cohen, and Steven LaValle. “Sensor beams, obstacles, and possible paths.” Algorithmic Foundation of Robotics VIII (2009): 317-332.

[62] Tovar, Benjamin, Fred Cohen, and Steven LaValle. “Optimal Navigation and Object Finding without Geometric Maps or Localization.” Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on. Vol. 1. IEEE (2003).

[63] Tovar, Benjamın, Rafael Murrieta-Cid, and Steven M. LaValle. “Distance-optimal navigation in an unknown environment without sensing distances.” Robotics, IEEE Transactions on 23.3 (2007): 506-518.

[64] Tseng, L. H., P. Heffernan, and D. T. Lee. “Two-guard walkability of simple polygons.” International Journal of Computational Geometry Applications 8.01 (1998): 85-116.

[65] Yu, Jingjin, and Steven M. LaValle. “Shadow Information Spaces: Combinatorial Filters for Tracking Targets.” Robotics, IEEE Transactions on 28.2 (2012): 440-456.

[66] http://custom-build-robots.com/robot-cars/discoverer-remote-controlled-robot-with-a-raspberry-pi/48

[67] http://cvrr.ucsd.edu/

[68] https://earth.boisestate.edu/cryogars/research/robots/

[69] http://www.geology.smu.edu/ dpa-www/robo/jbot/

[70] https://jwinburnnc.wordpress.com/tag/odometry/

[71] https://learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor

[72] http://www.lidarindonesia.com/kutipan-email-raharjo/?lang=

[73] http://www.mogi.bme.hu/TAMOP/jarmurendszerek$_i ranyitasa_a ngol/math-ch03.html$

[74] https://www.ri.cmu.edu/research$_guide/vision.html$

[75] http://www.sensorsmag.com/about-sensors

[76] http://spectrum.ieee.org/automaton/robotics/robotics-hardware/sweep-lidar-for-robots-and-drones

[77] http://www.theoldrobots.com/BeetleRobot.html

[78] https://www.utwente.nl/ctw/aida/education/Rapport$_MP.pdf$

[79] https://www.wired.com/2016/08/how-tesla-autopilot-works/

[۸۰] ابویی مهریزی، محمد. الگوریتمی جهت مسیریابی چند ربات با حسگرهای ضعیف، پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی شریف (۱۳۹۳).

[۸۱] هراتیان، سعید. پیدا کردن هسته چندضلعی توسط یک ربات ساده. پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی شریف (۱۳۹۲).

واژه‌نامه

ادغام	Merge
اطلاعات	Information
انعکاسی	Inflection
بازخورد	Feedback
برخط	Online
بدوی	Primitive
پدیدار شدن	Appear
پیکربندی	Configuration
پیوند	Link
تعقیب کننده	Pursuer
تعمیم‌یافته	Generalized
تقسیم	Split
حالت	State
حرکت	Walking
حسگر	Sensor

Beam Sensor	حسگر پرتوی
Gap Sensor	حسگر شکاف
Minimal Sensor	حسگر محدود
Street	خیابان
Strategy	راهبرد
Simple Robot	ربات ساده
Event	رخداد
Competitive	رقابتی
Chain	زنجیر
Navigation Tree	درخت پیمایش
Adversary	دشمن
Data Structure	ساختار داده
Shadow	سایه
Star	ستاره
Gap	شکاف
Object	شی
Ratio	ضریب
Inactive، Passive	غیر فعال
Space	فضا
Active	فعال
Compass	قطب‌نما
Funnel	قیف

Eave	گوش
Rectilinear	متعامد
Local	محلی
Detour	مسیر انحرافی
Art Gallery	موزه هنر
Disappear	ناپدید شدن
Unknown	ناشناخته
Critical Point	نقطه‌ی بحرانی
Turn Point	نقطه‌ی چرخش
Chord	وتر
Target	هدف
Kernel	هسته

Abstract

In this research, path planning in unknown environment is considered. It is assumed that the robot's sensors are the only tools to collect information from the scene. Volume of the information gathered from the environment depends on the capability of the sensor. So, a detour from the optimal path is unavoidable. In this research, the basic robot is equipped with a minimal sensing system that only detects the discontinuities in depth information (gaps). We present some online search strategies that guide such a robot to navigate unknown streets from a start point s to reach a target point t. Then, we empower the robot by adding a compass to patrol more general classes of polygons. We present some online strategies that generate search paths for the empowered robot in Generalized streets, a more general class of polygons that contains all streets properly, and in Star shapes. In order to evaluate the efficiency of our online search strategies for the robot, we use the notation of the competitive analysis.

Keywords: Computational Geometry, Robot Path planning, Minimal Sensing, Unknown Environment, Competitive Analysis

Sharif University of Technology
Computer Engineering

PhD Thesis

Topic

Path Planning in Unknown Environment with Minimal Sensing

By
Azadeh Tabatabaei

Supervisor
Mohammad Ghodsi

September 2016