حرکت برخط ربات در چندضلعی‌های با موانع دایره‌ای

عین‌الله پیرا   محمّد قدسی

دانشکدهٔ مهندسی کامپیوتر
دانشگاه صنعتی شریف

چکیده

یک چندضلعی با موانع داخل آن مفروض است. هدف این است که یک ربات نقطه‌ای از نقطه مبدأ s حرکت کند و به نقطه هدف t برسد. ربات موقعیت مبدأ و مقصد را می‌داند ولی موقعی می‌تواند هدف را تشخیص دهد که آن را ببیند. ربات قابلیت جابجایی و چرخش در هر زاویه‌ای را داراست. همچنین ربات در هر نقطه‌ای می‌تواند موقعیت خود را تعیین کند و فاصله خود را تا یک نقطهِ دیگر محاسبه کند. ربات همه اطلاعات مربوط به چندضلعی را ندارد و باید بصورت برخط حرکت کند و الگوریتمی که برای حرکت آن در نظر گرفته می‌شود بعنوان یک الگوریتم برخط تلقی می‌شود. در این مقاله یک الگوریتم برخط برای حرکت ربات در چندضلعی‌هایی که شامل موانع دایره‌ای می‌باشند ارائه خواهیم کرد و سپس آن را تحلیل خواهیم کرد.

کلمات کلیدی: برنامه‌ریزی حرکت، تحلیل تقریبی، الگوریتم برخط ، مسیر بهینه

۱ مقدمه

پیدا کردن مسیر و مسیر بهینه در دامنه هندسی یکی از مسائل پایه‌ای در هندسه محاسباتی به شمار می‌آید و کاربردهای زیادی در سیستمهای اطلاعاتی جغرافیایی (GIS[۱])، مسیریابی سیم[۲]، رباتیک و غیره دارد و رایج‌ترین فرم پایه این مسأله، پیداکردن کوتاهترین فاصله اقلیدسی بین دو نقطه می‌باشد که با موانع داده شده برخورد نداشته باشد. اکثر کارهای انجام شده بر روی حرکت ربات به دو دسته تقسیم می‌شوند: ۱ــ حرکت ربات در یک فضای شناخته شده که در آن ربات نقشه کاملی از محیط را دارد و می‌تواند از الگوریتم‌های DFS[۳] یا BFS[۴] استفاده کرده و یک مسیر بهینه بدست آورد. ۲ــ حرکت ربات در یک فضای ناشناخته که ربات اطلاعات همه نقشه را ندارد و در واقع اطلاعات ربات شامل نقشه‌های نمایان محلی می‌باشد. همچنان که ربات حرکت می‌کند می‌تواند اطلاعات جزئی از محیط را بدست آورده و آنها را پردازش کند. ربات بعلت فقدان اطلاعات کامل چندضلعی نمی‌تواند یک مسیر بهینه پیدا کند. در واقع اطلاعات ربات شامل نقشه‌های

---

[۱] Geographic Information Systems
[۲] Wire Routing
[۳] Depth First Search
[۴] Breadth First Search

نمایان[5]محلی می‌باشد بنابراین استراتژی حرکت باید بصورت برخط انجام شود و بصورت تقریبی تحلیل شود.

در این مقاله فرض می‌کنیم ناحیه S، ناحیه دوبعدی شامل نقطه شروع s و نقطه هدف t با مجموعه‌ای از موانع دایره‌ای باشد بطوریکه این موانع با هم تماس ندارند و هیچکدام از موانع شامل نقطه s و t نمی‌باشند. هدف t ممکن است یک نقطه یا یک دیوار نامتناهی باشد. رباتِ نقطه‌ای، مجبور است از s تا t پیمایش کند و در طی این پیمایش، باید بتواند موانع را در S تشخیص دهد. ربات موقعیت و اندازه موانع را نمی‌داند و زمانی می‌تواند موقعیت و اندازه آنها را یاد بگیرد که حداقل سه نقطه از یک مانع را ببیند (چون اگر ربات سه نقطه از یک دایره را داشته باشد می تواند مرکز و شعاع دایره را معین کند).

فرض کنیم $R(S)$ مسافت پیموده شده بوسیله ربات R از مبدأ s تا هدف t در فضای دوبعدی S باشد، $d(S)$ طول کوتاهترین مسیر(حالت برون‌خط[6]) بین s و t در فضای دوبعدی S باشد و $S(n)$ مجموعه نواحی دوبعدی باشد که فاصله بین s و t، n باشد. شکل دیگری از متریک برای ربات، ضریب

$$\rho(R,n) = \sup \frac{R(S)}{d(S)}$$

می‌باشد که رشد[7] $\rho(R,n)$ یک تابعی از n (فاصله بین مبدأ و هدف) می‌باشد و نشان دهنده اینست که مسافت پیموده شده توسط ربات در حالت برخط در حدود $\rho(R,n)$ برابر مسافت پیموده شده در حالت برون‌خط می‌باشد.

در این مقاله حرکت ربات را در چندضلعی هایی بررسی خواهیم کرد که در آنها هدف t بصورت یک خط عمودی در فاصله n از مبدأ s قرار دارد (که به مسأله دیوار[8] معروف است.) و موانع بصورت دایره می‌باشند. الگوریتمی که برای حل این مسأله در نظر گرفته‌ایم از ایده جاروب استفاده می‌کند. الگوریتم پنجره‌ای در اندازه‌های متغیر نگهداری می‌کند و ربات تعداد $\sqrt{n}$ جاروب در جهت‌های متناوب ما بین شمال و جنوب برای هر مقدار اندازه پنجره انجام می‌دهد. با کامل شدن این $\sqrt{n}$ جاروب، اندازه پنجره دو برابر می شود.

اگر اندازه پنجره در یک زمان W باشد (که از $y = \frac{+W}{2}$ تا $y = \frac{-W}{2}$ قرار گرفته‌باشد)، مسافت پیموده شده توسط ربات در طی جاروب $O(W\sqrt{n})$ خواهد بود. فرض کنید موقعی که ربات به t می‌رسد اندازه پنجره، W_f باشد ثابت خواهیم کرد که کل مسافت پیمایش شده توسط ربات $O(W_f\sqrt{n})$ می‌باشد در حالی که طول کوتاهترین مسیر(عاری از مانع) برابر $d(S) = \Omega(W_f)$ می‌باشد.

در بخش دوم به نمونه‌ای از الگوریتمهای برخط اشاره خواهیم کرد، در بخش سوم، خلاصه‌ای از نتایج کارهای قبلی آورده می‌شود و در نهایت الگوریتمی برای مسأله دیوار با موانع دایره‌ای ارائه و سپس آن را تحلیل خواهیم کرد.

۲ الگوریتمهای برخط

در طراحی و تحلیل الگوریتمها، فرض می کنیم الگوریتمی که خروجی تولید می‌کند، دانش کاملی نسبت به کلیه ورودی‌هایش دارد. البته این در بعضی از مسائل کاربردی، غیرقابل تحقّق[9] می‌باشد. برخی از مسائلی که در عمل رخ می‌دهند برخط[10] هستند. در این نوع مسائل، اطلاعات ورودی، بطور جزئی در دسترس می‌باشند. یک الگوریتم برخط، باید خروجی را بدون داشتن دانش کاملی از کل ورودیها تولید کند. مسائل برخط در حوزه‌هایی مانند تخصیص منابع[11] در سیستم‌های عامل، داده ساختار، محاسبات توزیع شده، زمانبندی و رباتیک رخ می‌دهند. برای روشن شدن موضوع، مثالهای زیر را ارائه می‌کنیم.

صفحه‌بندی[12]: در سیستم حافظه‌ایِ دو سطحه که در آن، حافظه شامل حافظه سریع با ظرفیّت کم و حافظه کند با ظرفیّت زیاد می‌باشد یک الگوریتم صفحه‌بندی باید بطور فعالانه، صفحات رجوع شونده در حافظه سریع را نگهداری کند بدون اینکه بداند در آینده چه صفحاتی مورد نیاز خواهند بود.

[5] Visibility
[6] الگوریتم برون‌خط، الگوریتمی است که به همه اطلاعات ورودی دسترسی دارد و می‌تواند یک خروجی بهینه تولید کند.
[7] growth
[8] Wall Problem
[9] Unrealistic
[10] Online
[11] Resource Allocation
[12] Paging

زمانبندی چندپردازنده‌ای[۱۳]: دنباله‌ای از کارها[۱۴] باید بر روی یک مجموعه از ماشینها زمانبندی بشوند. کارها یکی‌یکی به سیستم می‌رسند و فوراً باید زمانبندی شوند بدون اینکه بدانیم درآینده چه کارهای دیگری خواهند رسید.

مسائل پیمایش در رباتیک[۱۵]: رباتی در یک ناحیه نامشخص قرار گرفته است و مجبور است یک مسیر کوتاه از نقطه s به نقطه t پیدا کند. ربات همچنان که حرکت می‌کند می‌تواند اطلاعاتی در مورد ناحیه بدست آورد.

در سالهای اخیر، نشان داده شده است که تحلیل رقابتی [۱۶] یک ابزار قوی در تحلیل کارایی الگوریتمهای برخط می‌باشد. ایده رقابت[۱۷] اینست که خروجی تولید شده بوسیله الگوریتم برخط را با خروجی تولید شده بوسیله الگوریتم برون‌خط[۱۸] مقایسه کنیم. یک الگوریتم برون‌خط، الگوریتمی است که به همه اطلاعات ورودی دسترسی دارد و می‌تواند یک خروجی بهینه تولید کند[۱].

۳ خلاصه نتایج کارهای قبلی

در [۲]، Yannakakis و Papadimitriou نشان دادند که اگر موانع، بصورت مربع هایی با اضلاع موازی با کران چندضلعی باشند و n فاصله اقلیدسی بین s و t باشد آنگاه الگوریتم برخط با ضریب رقابتی $\rho(R,n) = \frac{\sqrt{26}}{3} + O(1)$ برای همه n ها وجود دارد (اگر مختصات x یا y نقاط s و t باهم برابر باشند آنگاه $\rho(R,n) = \frac{5}{3}$ برای همه n ها خواهد بود). اگر موانع، مربعهای واحد با اضلاع موازی با کران چندضلعی باشند استراتژی با ضریب رقابتی $\rho(R,n) = 1.5 + O(1)$ برای همه n ها خواهد بود. علاوه‌براین، آنها نشان دادند که اگر موانع، مستطیلهایی باریک با اضلاع موازی با کران چندضلعی باشند هیچ استراتژی با ضریب رقابتی محدود برای یک ربات با دید خط وجود ندارد. در حقیقت، در [۲، ۳]، نشان داده شد که اگر هدف t یک خط عمودی نامحدود «دیوار» در فاصله n از s باشد و موانع، مستطیلهایی با اضلاع موازی با کران چندضلعی باشند پس $\rho(R,n) = \Omega(\sqrt{n})$ خواهد بود. Blum، Raghavan و Schieber[۴] یک الگوریتم جاروب برای این مسأله دیوار ارائه کردند و نشان دادند که $\rho(R,n) = O(\sqrt{n})$ خواهد بود.

Blum در [۴] «مسأله خانه»[۱۹]، P شامل یک مربع $n \times n$ با موانع مستطیلی با اضلاع موازی با کران چندضلعی، را مورد مطالعه قرار داد. برای این مسأله خانه، الگوریتمی با ضریب رقابتی $2^{\sqrt{3\log n}}$ ارائه داد. Bar-Eli در [۵] این ضریب رقابتی را بهبودش کرد و ضریب رقابتی $\rho(R,n) = \theta(\log n)$ ارائه کرد. مسأله دیوار و خانه می‌توانند باهم ترکیب شده و یک استراتژی با ضریب رقابتی $\rho(R,n) = O(\sqrt{n})$ برای مسأله پیمایش نقطه‌به‌نقطه ما بین موانع مستطیلی با اضلاع موازی با کران چندضلعی ارائه شود. اگر اضلاع موانع مستطیلی موازی کران چندضلعی نباشند Blum [۴] نشان داد که در این حالت نیز $\rho(R,n) = \Omega(\sqrt{n})$ خواهد بود.

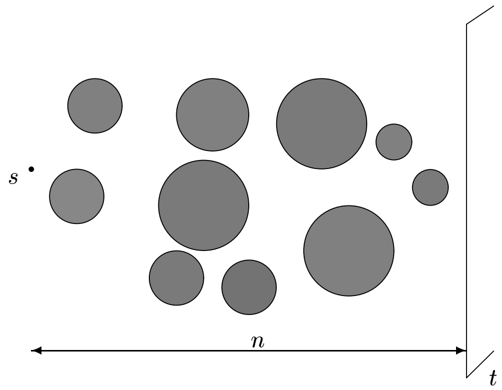
Coffman و Gilbert در سال ۱۹۹۳ کارایی الگوریتمهای ساده مکاشفه‌ای[۲۰] را برای حالتی که موانع، بطور تصادفی در مکانهای نامعلوم قرار گرفته‌باشند مورد مطالعه قرار دادند [۶]. Lumelsky و Stepanov در سال ۱۹۸۶ الگوریتمی ارائه کردند که تضمین می‌کند R(S) توسط مجموع d(S) و مجموع قطرهای تمام موانع محدود شده باشد[۷].

برای الگوریتمهای معین، حد محکم[۲۱] $\rho(R,n) = \theta(\sqrt{n})$ برای نسخه‌های دیوار و کاوش نقطه‌به‌نقطه مسأله وجود دارد. درحالیکه، Berman در [۸] نشان داد که الگوریتمهای تصادفی، قدرتمند هستند و برای مسأله دیوار و کاوش نقطه‌به‌نقطه، الگوریتم تصادفی با ضریب رقابتی $O(n^{\frac{4}{9}}\log n)$ ارائه کرد. Berman و Karpinski در [۹] یک الگوریتم تصادفی برای موانع محدب با ضریب رقابتی $O(n^{\frac{3}{4}})$ ارائه کردند.

۴ مسأله دیوار برای موانع دایره‌ای

همانطوریکه در بخش قبلی ذکر شد Blum مسأله دیوار را برای حالتی که موانع بصورت مستطیلهایی با اضلاع موازی با کران

[۱۳] Multiprocessor Scheduling
[۱۴] Job
[۱۵] Navigation Problems in Robotics
[۱۶] Competitive Analysis
[۱۷] Competitiveness
[۱۸] Offline
[۱۹] Room Problem
[۲۰] heuristic
[۲۱] Tight



شکل ۱: محیط با موانع دایره‌ای برای مسأله دیوار

محیط باشند را حل کرد. همچنین Blum ثابت کرده است که هیچ راه‌حلی برای پیمایش[۲۲] نقطه‌به‌نقطه با موانع محدب وجود ندارد (منظور از پیمایشِ نقطه به نقطه، اینست که ربات از هر نقطه دلخواه شروع به حرکت کرده تا به یک نقطه دلخواه دیگر برسد). الگوریتمی که در اینجا ارائه می‌کنیم برای حل مسأله دیوار می‌باشد بطوریکه موانع بصورت دایره‌ای می‌باشند (شکل ۱ را ببینید).

برای راحتی کار، چندضلعی را با استفاده از جهت جنوب/شمال برای کاهش/افزایش مقدار y، شرق/غرب برای افزایش/کاهش مقدار x در نظر می‌گیریم. همچنین خط مماس بر بالاترین نقطه دایره را «مماس بالایی» و پایین‌ترین نقطه را «مماس پایینی» نام‌گذاری می‌کنیم.

برای حل مسأله دیوار، از روش جاروب که توسط Blum در سال ۱۹۹۷ ارائه شده، ایده گرفته‌ایم. الگوریتم چهار تا متغیر بصورت زیر نگهداری می‌کند: ۱– اندازه پنجره W ۲– آستانه[۲۳] τ ۳– جهت جاروب ۴– شمارنده جاروب. که در ابتدای الگوریتم، متغیرها را بصورت زیر مقداردهی اولیه می‌کنیم: ۱– $W := n$ ۲– $\tau = W/(\pi\sqrt{n})$ ۳– جهت جاروب را بطرف جنوب در نظر می‌گیریم ۴– شمارنده جاروب را به صفر مقداردهی می‌کنیم.

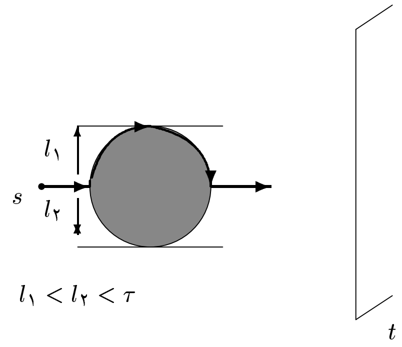
الگوریتم، پنجره‌ای در اندازه‌های متغیر نگهداری می‌کند و ربات تعداد $\sqrt{n}$ جاروب در جهت‌های متناوب بین شمال و جنوب برای هر مقدار اندازه پنجره انجام می‌دهد. با کامل شدن این $\sqrt{n}$ جاروب، اندازه پنجره دو برابر می‌شود. اگر اندازه پنجره در یک زمان W باشد (که از $y = \frac{+W}{2}$ تا $y = \frac{-W}{2}$ قرار گرفته‌باشد)، مسافت پیموده شده توسط ربات در طی $\sqrt{n}$ جاروب، $O(W\sqrt{n})$ می‌باشد. همچنین کوتاهترین مسیر در طی $\sqrt{n}$ جاروب دارای طول $\Omega(W)$ می‌باشد. فرض کنید موقعی که ربات به t می‌رسد اندازه پنجره، W_f باشد ثابت خواهیم کرد که کل مسافت پیمایش شده توسط ربات $O(W_f\sqrt{n})$ می‌باشد و طول کوتاهترین مسیر(عاری از مانع) برابر $d(S) = \Omega(W_f)$ خواهد بود.

شکل کلی الگوریتم بصورت زیر می‌باشد:

ربات از نقطه شروع s بطرف شرق شروع به حرکت می‌کند تا موقعی که یا به t برسد و یا با یک مانع در نقطه‌ای مثل (x, y) برخورد کند. فرض کنیم جهت جاروب فعلی بطرف جنوب باشد(برای حالتی که جهت جاروب بطرف شمال باشد برعکس این می‌باشد) مراحل بعدی الگوریتم بوسیله قوانین زیر معین می‌شود:

- قاعده ۱: اگر فاصله ما بین موقعیت فعلی ربات تا نزدیکترین خط مماس بر مانع (بالایی یا پایینی) کمتر از τ باشد پس ربات مطابق شکل ۲ از مانع عبور خواهد کرد.
- قاعده ۲: اگر مماس بالایی و پایینی مانع از پنجره بیرون زده باشد (یعنی اندازه y یال شمالی آن بزرگتراز $y = \frac{+W}{2}$ و اندازه y یال جنوبی آن کوچکتر از $y = \frac{-W}{2}$ باشد) پس الگوریتم اندازه پنجره W و آستانه τ را دو برابر می‌کند، شمارنده جاروب را به صفر مقداردهی می‌کند و جهت جاروب را بطرف جنوب قرار می‌دهد.

Navigation[۲۲]
threshold[۲۳]



شکل ۲: حرکت ربات دور یک مانع دایره‌ای

- قاعده ۳: در غیر اینصورت، ربات بطرف جنوب حرکت می‌کند تا یا به مماس پایینی مانع برسد و یا به کران پایین پنجره برسد. در اولین حالت، ربات بطرف شرق حرکت می‌کند تا به مانع بعدی برسد(یا به t برسد)(قسمت (ii) شکل ۳ راببینید). در دومین حالت، ربات مقدار شمارنده جاروب را یک واحد افزایش داده و جهت جاروب را برعکس می‌کند و اگر اندازه شمارنده جاروب از $\sqrt{n}$ بیشتر باشد آنگاه ربات شمارنده جاروب را به صفر مقداردهی می‌کند و اندازه پنجره و آستانه را دو برابر می‌کند (قسمت (i) شکل ۳ راببینید).

فرض کنید موقعی که ربات به t می‌رسد اندازه پنجره، W_f باشد.

قضیه ۱. کل مسافت پیموده شده بوسیله ربات، $O(W_f\sqrt{n})$ می‌باشد.

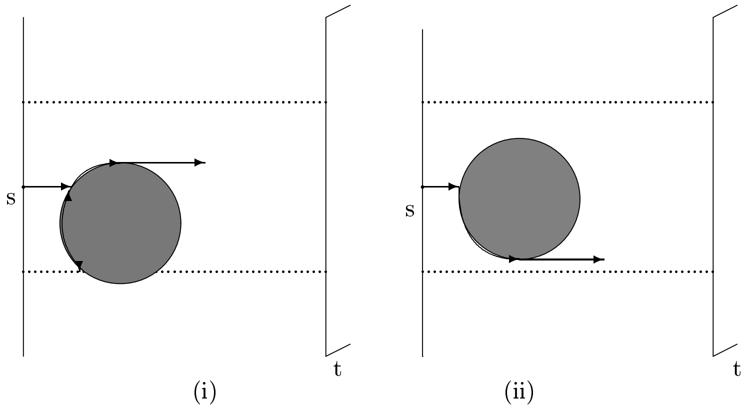
اثبات. برای اینکه ثابت کنیم کل مسافت پیموده شده بوسیله ربات با $O(W_f\sqrt{n})$ محدود شده‌است، مسیر پیموده شده بوسیله ربات را به سه قطعه بصورت زیر تقسیم می‌کنیم: (۱) قطعه‌های افقی (۲) قطعه‌های پیمایش شده بطرف جنوب و شمال حول موانع با استفاده از قاعده ۱ (۳) قطعه‌های پیموده شده بطرف جنوب و شمال با استفاده از قاعده ۳. توجه داشته‌باشید که (i) کل مسافت پیمایش شده بطرف شرق (طول قطعه‌های افقی) برابر n است که $n \leq W_f\sqrt{n}$ می‌باشد چون $n \leq W_f$ و (ii) کل مسافت پیمایش شده بطرف جنوب و شمال با استفاده از قاعده ۱ در بدترین حالت برابر با $W_f\sqrt{n}$ خواهد بود (شکل ۴ را ببینید.) و $n\pi\tau_f \leq W_f\sqrt{n}$ که در آن، τ_f آستانه نهایی می‌باشد.

حال کافی است ثابت کنیم طول کل سومین قطعه‌ها نیز با $O(W_f\sqrt{n})$ محدود می‌شود. برای اینکار، اندازه پنجره را در یک زمان، W در نظر می‌گیریم مسافت پیمایش شده بوسیله ربات با استفاده از قاعده ۳ برای کامل‌کردن یک جاروب، $O(W)$ می‌باشد و چون ربات حداقل $\sqrt{n}$ جاروب برای هر اندازه پنجره انجام می‌دهد پس کل مسافت پیموده شده برای اندازه پنجره W برابر $O(W\sqrt{n})$ می‌باشد. موقعی که اندازه پنجره عوض می‌شود دو برابر می‌شود و بنابراین مسافت نهایی پیموده‌شده برای همه اندازه‌های پنجره برابر $O(W_f\sqrt{n})$ خواهد بود.□

قضیه ۲. طول کوتاهترین مسیر از s تا t، $d(S)$، برابر $\Omega(W_f)$ می‌باشد.

اثبات. فرض می‌کنیم مسیر، شامل قطعه‌های افقی و قطاع های دایره‌ای باشد. واضح است که طول قطعه‌های افقی حداقل n می‌باشد. اگر $W_f = n$ باشد آنگاه درستی قضیه نتیجه می‌شود. همچنین اگر W_f توسط قاعده ۲ تعیین گردد (مانعی وجود دارد که مماس پایینی و بالایی دایره از پنجره بیرون زده باشد) آنگاه درستی قضیه نتیجه می‌شود. بنابراین فرض می‌کنیم هیچکدام از این دو حالت اتفاق نیافتد که در نتیجه، ربات حداقل باید $\sqrt{n}$ جاروب کامل برای بعضی اندازه پنجرهٔ $W \leq ۰.۵W_f$ کامل کند. حال نشان می‌دهیم که طول کل قطاع های دایره‌ای موجود در کوتاهترین مسیر برابر $\Omega(W) = \Omega(W_f)$ می‌باشد.

یک نقطه‌ای بر روی کوتاهترین مسیر بطوریکه دارای بیشترین مقدار مطلق روی محور y ها باشد در نظر بگیرید، اگر این مقدار مطلق حداقل $\frac{W}{۲}-\tau$ باشد آنگاه واضح است که کوتاهترین مسیر دارای حداقل طول $\frac{W}{۲}-\tau = \Omega(W)$ خواهد بود. فرض کنید این حالت رخ ندهد. یک کوتاهترین مسیری را در نظر بگیرید، برای هر $\sqrt{n}$ جاروب، اولین نقطه ورودی را



شکل ۳: اجرای یک مرحله از الگوریتم

اولین نقطه بر روی کوتاهترین مسیر که مختصات x آن مشابه مختصات x از نقطه شروعِ جاروب باشد در نظر بگیرید بطور مشابه، اولین نقطه خروجی را اولین نقطه بر روی کوتاهترین مسیر که مختصات x آن مشابه مختصات x از نقطه پایانیِ جاروب باشد در نظر بگیرید. توجه داشته باشید که چون مختصات x جاروبها باهم برابر نیستند پس نقطه خروجی جاروبِ i ام قبل از نقطه ورودی جاروبِ $i+1$ ام می‌باشد. بنابراین برای محاسبه حد پایین طول کوتاهترین مسیر می‌توانیم برای هر جاروب i ، قطاع دایره‌ای از کوتاهترین مسیر، از نقطه ورودی i ام تا نقطه خروجی i ام را در نظر بگیریم. موانع لمس شده بوسیله ربات در طی بعضی جاروبها– فرضاً جهت جاروب بطرف جنوب باشد– نیاز به استفاده از قاعده ۳ دارد. هر چنین مانعی حداقل $\pi\tau = \frac{2\pi\tau}{2}$ واحد از مماس بالایی مانع بطرف شمالتر قرار دارد بنابراین پیمایش از نقطه ورودیِ جاروب تا نقطه خروجیِ جاروب نیاز به پیمایش مسافت قطاع حداقل $\pi\tau$ دارد و چون $\sqrt{n}$ جاروب وجود دارد پس طول کل قطعات عمودی حداقل $W = \pi\tau\sqrt{n}$ خواهد بود.□

نتیجه. الگوریتم جاروبِ ارائه شده در اینجا برای مسأله دیوار، دارای ضریب رقابتیِ $\rho(R,n) = O(\sqrt{n})$ می‌باشد.

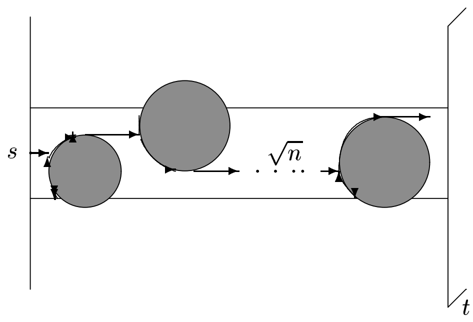
با یک تغییر جزئی روی الگوریتم ارائه شده، می‌توان ضریب رقابتی $O(\sqrt{n})$ را برای حالتی که هدف(دیوار) بصورت عمودی نباشد بدست آورد.

قضیه ۳. می‌توان الگوریتم جاروب را طوری تغییر داد که اگر چه هدف(دیوار) بصورت عمودی نباشد امّا الگوریتم، دارای ضریب رقابتی $O(\sqrt{n})$ باشد.

اثبات. الگوریتم به زاویه θ که t با محور y هامی‌سازد بستگی دارد. فرض کنید که $0 \le \theta < \frac{\pi}{4}$ و جهت دیوار از جنوب غربی به شمال شرقی باشد (سه حالت دیگر نیز مشابه این حالت می‌باشد).دراینجا n طول کوتاهترین فاصله اقلیدسی ما بین s و t می‌باشد. دو حالت زیر را خواهیم داشت:

- حالت ۱: $\sin\theta \le \frac{1}{\sqrt{n}}$. در این حالت، ربات با استفاده از مسیر حریصانهِ شرق–جنوب (مسیری که در صورت امکان بطرف شرق پیموده می‌شود در غیراینصورت به طرف جنوب پیموده می‌شود) از نقطه شروع s بطرف هدف t حرکت می‌کند. مشاهده می‌شود که طول این مسیر حریصانه، مسافت L_1 مابین s و نقطه‌ای از t که مسیر با آن برخورد می‌کند می‌باشد. طول مختصات x از این مسافت L_1 بزرگتر از $\sqrt{n}$ نمی‌باشد. چون $\sin\theta \le \frac{1}{\sqrt{n}}$ پس طول مختصات y نیز حداکثر $O(n^{1.5})$ خواهد بود که در نتیجه خواهیم داشت $\rho(R,n) = O(\sqrt{n})$.

- حالت ۲: $\frac{1}{\sqrt{n}} \le \sin\theta$. در این حالت، الگوریتم جاروب را تا موقعی اجرا می‌کنیم که ربات به نقطه (x_0, y_0) برسد بطوریکه نقطه $(x_0, -W/2)$ روی یا زیر t باشد که W پهنای پنجره فعلی می‌باشد. سپس ربات با استفاده از مسیر حریصانه شرق–جنوب بطرف t حرکت می‌کند. با توجه به قضیه ۱، مسافت پیموده شده توسط ربات تا رسیدن به نقطه (x_0, y_0) برابر $O(\sqrt{n})$ می‌باشد. واضح است که طول مسیر حریصانه از آن نقطه تا t برابر$O(n+W)$ می‌باشد. با توجه به برهان قضیه ۲، کوتاهترین مسیر از s تا خط عمودی $x = x_0$ دارای طول $O(W)$ می‌باشد. بنابراین با توجه به شرط



شکل ۴: بدترین حالت الگوریتمِ جاروب

روی θ، کوتاهترین مسیر از s تا t دارای طول $O(W)$ خواهدبود.

۱.۴ پیمایش نقطه به نقطه

می‌خواهیم الگوریتمهای برای مسأله دیوار و خانه را باهم ترکیب کرده و الگوریتمی برای پیمایش در محیطهایی که t یک نقطه در (n,n) و موانع بصورت دایره باشند ارائه کنیم. ربات از نقطه شروع s توسط یک مسیر حریصانه[۲۴] $<+x,+y>$ به نقطه s' با مختصات x یا y مساوی با t می‌رسد. فرض کنیم s' و t دارای مختصات y یکسانی باشند. حال ربات در نقطه s' الگوریتم جاروب برای مسأله دیوار را اجرا می‌کند تا به نقطه‌ای مانند $(n,y_۰)$ برسد. بدون کاهش عمومیت، فرض می‌کنیم $y_۰ \geq n$. توجه داشته باشید که مسیر از s' تا $(n,y_۰)$ توسط ربات در الگوریتم جاروب، هرگز در جهت محور x ها، کاهش نمی‌یابد و $y_۰ - n$ حداکثر W_f خواهد بود. این مسیر تضمین می‌کند که مسیر حریصانه[۲۵] $<-y,-x>$ از $(n,y_۰)$ به یک نقطه‌ای از $(x_۰,n)$ با $x_۰ \geq ۰$ خواهد رسید (در حقیقت، $x_۰$ حداقل باندازه مختصاتِ x از s' خواهد بود). حال با توجه به قضیه ۳.۲ در [۴] می‌توانیم برای مسأله دیوار را فراخوانی کنیم و به نقطه t برسیم.

حال مسافت پیموده شده توسط ربات را تحلیل می‌کنیم. مسافت پیموده شده با استفاده از الگوریتم برای مسأله دیوار حداکثر $O(W_f\sqrt{n})$ خواهد بود که W_f اندازه آخرین پنجره ملاحظه شده می‌باشد. اندازه خانه (بصورت مستطیلی) ایجاد شده حداکثر $n \times W_f$ خواهد بود. بنابراین با استفاده از قضیه ۳.۲ در [۴] مسافت پیموده شده برای رسیدن به t، $O(W_f\sqrt{n})$ خواهد بود. با استفاده از قضیه ۲، طول کوتاهترین مسیر از s' به t حداقل cW_f برای بعضی c های ثابت خواهد بود. حال اگر $W_f \leq \frac{۴n}{c}$ باشد پس نرخ $\rho(R,n) = O(\sqrt{n})$ خواهد بود چون $d(S) \geq n$ می‌باشد. اگر $W_f > \frac{۴n}{c}$ باشد چون طول کوتاهترین مسیر از s' تا s حداکثر $۲n$ می‌باشد پس خواهیم داشت $d(S) \geq cW_f - ۲n \geq \frac{c}{۲}W_f$ و در نتیجه ضریب $O(\sqrt{n})$ بدست خواهد آمد. بنابراین قضیه زیر را خواهیم داشت.

قضیه ۳. برای ناحیه دوبعدی S که در آن s و t نقاط دلخواهی در آن می‌باشند و موانع بصورت دایره می‌باشند الگوریتم ارائه شده در بالا دارای ضریب رقابتی $\rho(R,n) = O(\sqrt{n})$ خواهد بود.

۵ نتیجه گیری

در این مقاله ابتدا بیان کردیم که ممکن است در حل بعضی مسائل، همه اطلاعات و داده‌های مسأله را نداشته باشیم و مجبور بشویم آنرا مرحله به مرحله حل کنیم. بطور مثال، مسأله حرکت ربات در محیطهای شامل موانع را ذکر کردیم که ربات نقشه محیط را ندارد، در واقع اطلاعات ربات شامل نقشه‌های نمایان محلی می‌باشد، همچنان که ربات حرکت می‌کند می‌تواند اطلاعات جزئی از محیط را بدست آورده و آنها را پردازش کند یعنی حرکت ربات باید بصورت برخط باشد.

سپس مسأله حرکت ربات را در محیطهایی که شامل موانعی با اضلاع موازی با کران محیط باشند را بررسی کردیم

[۲۴] مسیری است که ربات در صورت امکان در جهت محور x ها حرکت می‌کند در غیر اینصورت در جهت محور y ها حرکت می‌کند.

[۲۵] مسیری است که ربات در صورت امکان در جهت منفی محور y ها حرکت می‌کند در غیر اینصورت در جهت منفی محور x ها حرکت می‌کند.

و خلاصهِ نتایج کارهایی که در این زمینه انجام شده بود را بیان کردیم. بعد از روشی که Blum برای حل حالت خاصی از مسأله(معروف به مسأله دیوار) ارائه کرده بود استفاده کرده و روشی برای حل مسأله دیوار موقعی که موانع بصورت دایره باشند ارائه کردیم.

مراجع

[1] S. Albers, *Competitive Online Algorithms,* SIAM J. Comput., 26(1):110-137,Feb. 1998

[2] C. H. Papadimitriou and Yannakakis, *Shortest paths without a map,* Proc. 16th ICALP, pp. 610-620,July 1989

[3] P.Eades, X.Lin , and N.C.Wormald, *Performance guarantees for motion planing with temporal uncertainy,* Australian Computer Journal , 25(1):21-28,1993

[4] A. Blum, P. Raghavan, and Schieber, *Navigating in unfamiliar geometric terrain,* SIAM J. Comput., 26(1):110-137,Feb. 1997

[5] E.BAr-Eli , P.Berman , A.Fiat, and P.Yan, *Online Navigation in a room,* J, Algorithms, 17:319-341.1994

[6] E. G. Coffman and E. N. Gilbert, *Paths through a maze of rectangles,* Networks, vol. 22,no. 4,pp. 349-367, July 1992

[7] V. J. Lumelsky and A. A. Stepanov, *Dynamic path planning for a mobile automaton with limited information on the environment,* IEEE Transactions on Automatic Control, AC-31:1058-1063, 1986

[8] P.Berman , A.Blum , A.Fiat , H.Karloff, A.Rosenn, and M.Saks, *Randomized Robot Navigation Algorithms,* In Proc , 7th ACM. SIAM Sympos, Discrete Algorithms , pages 74-84, 1996

[9] P.Berman and M.Karpiniski *Randomized navigation to a wall through convex obstacles,* Technical Report 85118-CS, Bonn University.1994