حرکت بهینه ربات میله‌ای

محمّد علی آبام   محمد قدسی

دانشکدهٔ مهندسی کامپیوتر
دانشگاه صنعتی شریف

چکیده

مساله d_1-optimal به این صورت بیان می‌شود: یک ربات میله‌ای با قابلیت جابه‌جایی و چرخش در یک محیط دوبعدی ایستا شامل موانع چندضلعی به‌همراه دو موقعیت ابتدایی و انتهایی داده شده است، حرکتی بدون برخورد با موانع بین این دو موقعیت پیدا کنید به‌طوری که طول اقلیدسی مسیر طی‌شده توسط نقطه ثابت اما دل‌خواه F روی ربات مینیمم شود. ما در این مقاله به حالت خاصی از این مساله که در آن ربات میله‌ای قابلیت جابه‌جایی و چرخش تحت زاویه‌های مشخص حول نقطه F را دار است پرداخته‌ایم. این مساله یک مساله جدید در این حوزه می‌باشد. کار انجام شده در این پروژه شامل توصیف حرکت d_1-optimal برای این حالت خاص و ارائه یک الگوریتم تقریبی با ضریب تقریب $1 + \epsilon$ می‌باشد.

کلید واژه‌ها: ربات میله‌ای   مسیر بهینه   الگوریتم تقریبی

۱ مقدمه

پیدا کردن مسیر بهینه ربات میله‌ای با قابلیت جابه‌جایی و چرخش تحت زاویه‌های مشخص حول یک نقطه ثابت اما دل‌خواه F در یک محیط دوبعدی ایستا با موانع چندضلعی مساله جدیدی است که در این مقاله به بررسی آن می‌پردازیم.

می‌توان تابع‌های هدف مختلفی برای اندازه‌گیری طول مسیر طی‌شده توسط ربات میله‌ای در نظر گرفت، اما طبیعی‌ترین تابع هدف، میانگین طول اقلیدسی مسیر طی‌شده n نقطه توزیع شده به‌طور یکنواخت روی ربات می‌باشد. به این تابع، d_n می‌گویند. به حرکت m از بین همه حرکت‌های موجود بین دو موقعیت ابتدایی و نهایی که دارای $d_n(m)$ مینیمم است، حرکت d_n-optimal می‌گویند. هدف در مساله d_n-optimal پیدا کردن حرکت d_n-optimal می‌باشد و به‌طور خاص در مساله d_1-optimal هدف پیدا کردن حرکتی است که طول اقلیدسی مسیر نقطه ثابت اما دل‌خواه F روی ربات مینیمم شود.

مساله d_2-optimal بدون حضور مانع که به مساله Ulam معروف است به‌طور کامل توسط Icking [۳] حل شده است اما با حضور مانع هم‌چنان به‌عنوان یک مساله باز مطرح می‌باشد. تاکنون الگوریتمی برای حل مساله d_n-optimal برای $n > 2$ ارائه نشده است و تنها O'Rourke [۴] مساله d_∞-optimal را برای رباتی که قابلیت جابه‌جایی و چرخش تحت زاویه ۹۰ درجه را دارد مورد مطالعه قرار داده و برای آن یک الگوریتم چندجمله‌ای برحسب تعداد اضلاع موانع ارائه کرده است. در [۱] ثابت شده مساله d_1-optimal زمانی که نقطه F داخل ربات واقع باشد، یک مساله NP-hard است و اگر بر یکی از دو سر ربات واقع باشد پیچیدگی آن مشخص نیست و

همچنان به‌عنوان یک مساله باز مطرح است. در [۱] همچنین یک الگوریتم تقریبی برحسب دو عدد مثبت دل‌خواه ϵ و ϵ' ارائه شده است. در [۵] حالت خاصی از این مساله که F واقع بر یکی از دو سر ربات و حرکت‌های آن محدود به حرکت روی پاره‌خط‌های مستقیمی است که رئوس موانع را به‌هم وصل می‌کند، مورد بررسی قرار گرفته و برای آن یک الگوریتم چندجمله‌ای ارائه شده است.

همان‌طور که گفته شد آنچه در این مقاله به بررسی آن خواهیم پرداخت حالت خاص شده‌ای از مساله $d_۱$-optimal می‌باشد. به‌طور دقیق‌تر فرض می‌کنیم ربات میله‌ای $\Re = \overrightarrow{AB}$ با طول ۱ دارای قابلیت جابه‌جایی و قرار گرفتن در جهت‌های $\{\alpha_۱, \cdots, \alpha_k\}$ باشد؛ به عبارت دیگر ربات می‌تواند برای تغییر جهت خود از α_i به α_j، چرخشی حول یک نقطه ثابت اما دل‌خواه F روی پاره‌خط $\overline{AB}$ تحت زاویه $\alpha_j - \alpha_i$ در جهت عکس عقربه ساعت انجام دهد. همچنین فرض می‌کنیم محیط کار شامل یک مجموعه $S = \{P_۱, \cdots, P_\ell\}$ از موانع چندضلعی محدب باشد به‌طوری که مانع P_i دارای p_iضلع است و $\sum_{i=۱}^{\ell} p_i = n$.

نتایج ما در این رساله شامل توصیف مسیر بهینه ربات میله‌ای با مشخصات ذکر شده و بیان یک الگوریتم تقریبی کارا با ضریب تقریب $۱ + \epsilon$ می‌باشد؛ به عبارتی دیگر اگر m^* حرکت بهینه و m حرکت حاصل از این الگوریتم بین دو موقعیت ابتدایی و نهایی داده شده باشد، داریم:

$$d_۱(m) \leq (۱ + \epsilon) d_۱(m^*)$$

زمان اجرای این الگوریتم وابسته به n مجموع تعداد اضلاع موانع، ϵ یک عدد مثبت دل‌خواه و L ماکزیمم تعداد بیت‌های هر عدد صحیح می‌باشد.

ادامه مقاله به این شکل تنظیم شده است. بخش ۲ شامل اصطلاحات و تعاریف پایه به‌همراه قضیه‌های مورد استفاده است. بخش ۳ به توصیف مسیر بهینه برای ربات میله‌ای می‌پردازد. در بخش ۴ یک الگوریتم تقریبی با ضریب تقریب $۱ + \epsilon$ ارائه شده است و در نهایت در بخش ۵ خلاصه‌ای از مطالب این مقاله به همراه نتیجه‌گیری آورده شده است.

۲ اصطلاحات و تعریف‌های پایه

در ابتدا به چند تعریف، مستقل از نوع ربات می‌پردازیم و در انتها چند اصطلاح برای ربات میله‌ای بیان می‌شود.

محیطی شامل یک مجموعه $S = \{P_۱, \cdots, P_\ell\}$ از موانع که در آن ربات قادر به حرکت است فضای کار گفته می‌شود. موقعیت ربات $\Re$ با قابلیت جابه‌جایی در یک محیط کار دوبعدی با مشخص کردن مختصات نقطه ثابتی از آن که از آن با نام نقطه مرجع یاد خواهیم کرد قابل بیان است. در حقیقت $\Re(x, y)$ به این معنی است که ربات در موقعیتی قرار گرفته که نقطه مرجع آن در نقطه (x, y) واقع شده است. اگر ربات علاوه بر جابه‌جایی، قابلیت چرخش نیز داشته باشد هر موقعیت ربات را می‌توان به‌وسیله سه پارامتر (x, y, α) نمایش داد که $x, y \in R^۲, \alpha \in [۰, ۲\pi)$. در حالت کلی، یک موقعیت ربات به‌وسیله تعدادی پارامتر که متناظر با تعداد درجه آزادی ربات می‌باشد قابل نمایش است.

به فضای پارامترهای ربات اصطلاحاً فضای پیکربندی گفته و با $C(\Re)$ نمایش داده می‌شود. هر نقطه p در فضای پیکربندی متناظر با یک موقعیت $\Re(p)$ در فضای کار است. برای ربات مسطح با قابلیت جابه‌جایی و چرخش، فضای پیکربندی سه‌بعدی $(R^۲ * [۰, ۲\pi))$ است. به زیرمجموعه‌ای از $C(\Re)$ که به ازای هر نقطه p از آن $\Re(p)$ با موانع برخورد دارد فضای ممنوعه گفته می‌شود و با$C_{forb}(\Re)$ نمایش داده می‌شود و به $C_{free}(\Re) = C(\Re) - C_{forb}(\Re)$ فضای آزاد گفته می‌شود. از آن‌جا که بین موقعیت‌های ربات در فضای کار و نقاط فضای پیکربندی تناظر یک‌به‌یک وجود دارد، می‌توان مسیر حرکت ربات در محیط کار را به یک منحنی در $C(\Re)$ نگاشت کرد و برعکس هر منحنی در فضای پیکربندی متناظر با یک مسیر حرکت برای ربات در فضای کار می‌باشد. البته هر مسیری برای ربات مناسب نمی‌باشد و در حقیقت ربات بدنبال یک مسیر بدون برخورد می‌باشد یعنی مسیری که منحنی متناظرش در $C(\Re)$ زیرمجموعه $C_{free}(\Re)$ باشد. مانع P به مجموعه نقاطی مثل p در فضای پیکربندی نگاشت می‌شود که $\Re(p)$ با مانع P برخورد داشته باشد و به این مجموعه، مانع فضای پیکربندی متناظر با P گفته و با $C_{obstacle}(P)$ نمایش داده می‌شود.

جمع مینکوسکی دو مجموعه $S_۱, S_۲ \subset R^۲$ که با $S_۱ \oplus S_۲$ نمایش داده می‌شود به‌صورت زیر تعریف می‌شود:

$$S_۱ \oplus S_۲ = \{p + q : p \in S_۱, q \in S_۲\}$$

که $p+q$ جمع برداری، بردارهای p و q می‌باشد. از آن‌جا که چندضلعی‌ها مجموعه‌های مسطح می‌باشند جمع مینکوسکی را می‌توان برای آن‌ها بکار برد. قضیه زیر رابطه $C_{obstacle}(P)$ با جمع مینکوسکی را بیان می‌کند:

قضیه ۱. [۲] اگر $\Re$ یک ربات مسطح با قابلیت جابه‌جایی و P یک مانع باشد، در این صورت $C_{obstacle}(P) = P \oplus (-\Re(0,0))$

برای ربات میله‌ای $\Re = \overrightarrow{AB}$ به طول ۱ نقطه ثابت اما دل‌خواه F روی آن را به‌عنوان نقطه مرجع در نظر می‌گیریم. بنابراین هر موقعیت ربات بوسیله یک جفت $Z = (p, \alpha) \in R^2 * [0, 2\pi)$ قابل نمایش است. برای هر $q \in R^2$، $q[Z]$ به معنی نقطه‌ای در R^2 است که از دوران q حول مبدا مختصات تحت زاویه α و بعد انتقال تحت بردار $\overrightarrow{p}$ بدست آمده است. هر حرکت را می‌توان به‌وسیله یک تابع $m : [0, 1] \rightarrow C(\Re)$ بیان کرد. و با توجه به این تابع برای هر نقطه $X \in R^2$ تابع $X_m : [0, 1] \rightarrow R^2$ را به‌صورت $X_m(x) = X[m(x)]$ تعریف می‌کنیم که در حقیقت X_m مسیر حرکت نقطه X تحت حرکت m را نشان می‌دهد.

برای ربات میله‌ای با قابلیت جابه‌جایی و قرار گرفتن در جهت‌های $\{\alpha_1, \cdots, \alpha_k\}$، فضای پیکربندی $(C(\Re) : R^2 * \{\alpha_1, \cdots, \alpha_k\})$ سه‌بعدی است. بطور دقیق‌تر حرکت ربات را می‌توان به شکل زیر تعریف کرد.

تعریف ۱. حرکت m از ربات از موقعیت $P = (s, \alpha_s)$ به موقعیت $Q = (t, \alpha_t)$ $(s, t \in R^2, \alpha_s, \alpha_t \in \{\alpha_1, \cdots, \alpha_k\})$ یک تابع به‌صورت $m : [0, 1] \rightarrow C(\Re)$ می‌باشد بطوری که $m(0) = P, m(1) = Q$ و تابع F_m، تابع پیوسته با درازای متناهی است.

طول حرکت m را به‌صورت $d_1(m) = sup_{(s_0=0<s_1<\cdots<s_r=1)} \sum_{i=0}^{r-1} ||(F(m(s_{i+1})), F(m(s_i)))||$ تعریف می‌کنیم($||.||$ بیانگر فاصله اقلیدسی می‌باشد) که در اینجا کوچکترین کران بالا[1] روی همه زیرتقسیم‌های $s_0 < s_1 < \cdots < s_r$ از بازه $[0, 1]$ گرفته می‌شود. تابع فاصله را به صورت $d_1(P, Q) = inf_{P \overset{m}{\rightarrow} Q} d_1(m)$ تعریف می‌کنیم که بزرگترین کران پایین[2] روی همه حرکت m بدون برخورد با موانع گرفته می‌شود و به حرکت بدون برخورد با موانع m که $d_1(P, Q) = d_1(m)$ باشد حرکت d_1-optimal می‌گوییم.

۳ ساختار حرکت بهینه

قبل ار بیان ویژگی‌های مسیر بهینه، به ساختار و نحوه محاسبه فضای ممنوعه می‌پردازیم

۱.۳ فضای ممنوعه

فضای پیکربندی $C(\Re) : R^2 * \{\alpha_1, \cdots, \alpha_k\}$ سه‌بعدی است. به‌طور هندسی این فضا از k صفحه $\{S_1, \cdots, S_k\}$ تشکیل شده که صفحه S_i متناظر با زاویه α_i می‌باشد و همه نقاط واقع بر این صفحه موقعیتی از ربات را بیان می‌کنند که جهت ربات برابر α_i است و از نماد $\Re_{\alpha_i}$ برای بیان این موقعیت ربات استفاده می‌کنیم. وقتی ربات داخل یکی از صفحات حرکت می‌کند در حقیقت تنها حرکت انتقالی انجام می‌دهد و وقتی از صفحه‌ای به صفحه دیگر می‌رود حرکت چرخشی انجام می‌دهد. نکته‌ای که باید به آن توجه کرد این است که ربات در حین چرخش نباید با موانع تماس داشته باشد. بنابر آنچه گفته شد در صفحه S_i دو نوع فضای ممنوعه داریم:

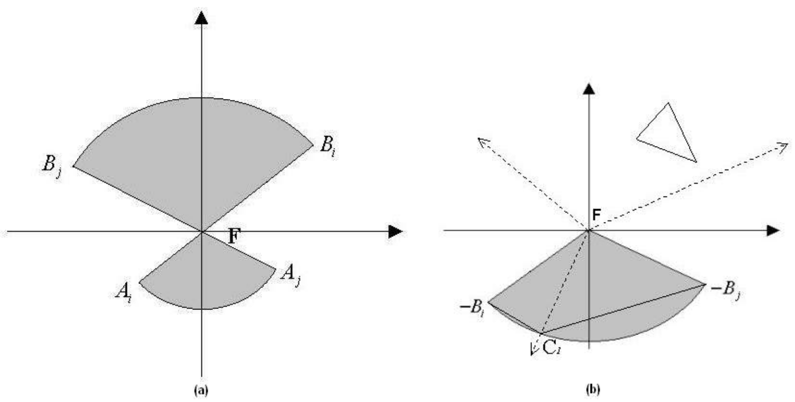
۱) فضای ممنوعه جابه‌جایی: این فضا را که با TFS_i[3] و مرز آن را با $BTFS_i$[4] نمایش می‌دهیم شامل نقاطی از صفحه S_i مثل (x, y) است که

$$\Re_{\alpha_i}(x, y) \cap (\cup_{j=1}^{l} P_j) \neq \emptyset$$

۲) فضای ممنوعه چرخشی: برای هر $\alpha_j (j \neq i)$ این فضا را که با RFS_i^j[5] و مرز آن را با $BRFS_i^j$[6] نمایش می‌دهیم شامل نقاطی از صفحه S_i مثل (x, y) است که

$$\exists \alpha \in [\alpha_i, \alpha_j] : \Re(x, y, \alpha) \cap (\cup_{j=1}^{l} P_j) \neq \emptyset$$

[1] Supremum
[2] Infimum
[3] Translating Forbidden Space
[4] Boundary Translating Forbidden Space
[5] Rotating Forbidden Space
[6] Boundary Rotating Forbidden Space



شکل ۱: (a) مساحت جاروب شده توسط ربات در حین چرخش (b) محاسبه جمع مینکوسکی قسمتی از یک دیسک با یک چندضلعی محدب

بنابراین در هر صفحه k فضای ممنوعه وجود دارد؛ یک فضای ممنوعه جابه‌جایی و $k-1$ فضای ممنوعه چرخشی که در ادامه به ساختار هر یک از این فضاها خواهیم پرداخت.

همان‌طور که در بخش ۲ بیان شد $TFS_i = -\Re_{\alpha_i}(\circ,\circ) \oplus (\cup_{j=1}^{l} P_j)$ اجتماع چندضلعی‌های محدب می‌باشد که به طور خلاصه می‌توان قضیه زیر را در مورد مرز این ناحیه بیان کرد:

قضیه ۲. پیچیدگی $BTFS_i$، $O(n)$ است و در زمان $O(n\log^2 n)$ قابل محاسبه است.

ساختار، نحوه محاسبه و زمان محاسبه $BTFS_i$ مشخص است؛ از این به بعد روی ساختار و نحوه محاسبه $BRFS_i^j$ تمرکز می‌کنیم. ربات برای رفتن از صفحه S_i به صفحه S_j باید حول F تحت زاویه $\alpha_j - \alpha_i$ در جهت عکس عقربه ساعت بچرخد و برعکس برای انتقال از صفحه S_j به S_i باید حول F تحت زاویه $\alpha_i - \alpha_j$ در جهت عکس عقربه ساعت بچرخد؛ در هر دو حالت ربات ناحیه‌ای مطابق شکل ۱(a) را جاروب می‌کند که این ناحیه را به نام $disk_{ij}$، قسمت بالایی آن که قسمتی از یک دیسک به شعاع $||FB||$ است را به نام $udisk_{ij}$ و قسمت پایینی آن که قسمتی از یک دیسک به شعاع $||FA||$ است را به نام $ldisk_{ij}$ می‌شناسیم (در شکل ۱(a) $(A_i = A[(\circ,\circ,\alpha_i)], A_j = A[(\circ,\circ,\alpha_j)], B_i = B[(\circ,\circ,\alpha_i)], B_j = B[(\circ,\circ,\alpha_j)]$). بنابراین می‌توان گفت:

$$RFS_i^j = -disk_{ij} \oplus \cup_{h=1}^{l} P_i = (-udisk_{ij} \oplus \cup_{h=1}^{l} P_i) \cup (-ldisk_{ij} \oplus \cup_{h=1}^{l} P_i)$$

بنابراین برای محاسبه RFS_i^j باید ابتدا هریک از $-ldisk_{ij} \oplus P_h$ و $-udisk_{ij} \oplus P_h$ را محاسبه کنیم که به دلیل تشابه در ادامه تنها به محاسبه $-udisk_{ij} \oplus P_h$ خواهیم پرداخت. فرض کنید $-udisk_{ij}$ محدب باشد (حالتی که غیر محدب باشد در ادامه بحث خواهد شد)، الگوریتم زیر نحوه محاسبه این جمع مینکوسکی را بیان می‌کند:

الگوریتم ۱

۱) قرینه $udisk_{ij}$ را محاسبه می‌کنیم که این کار را با قرینه کردن نقاط B_i و B_j که با $-B_i$ و $-B_j$ نمایش می‌دهیم به راحتی قابل انجام است.

۲) بردارهای عمود بر اضلاع مانع P_h به طرف خارج مانع را از مبدا رسم کرده و تقاطع آن را با کمان $\widehat{(-B_i)(-B_j)}$ بدست می آوریم. فرض کنید نقاط تقاطع $C_1, \cdots, C_r$ باشد (شکل ۱(b)). چون ممکن است بعضی از بردارها کمان را قطع نکنند، داریم $r \le p_h$. حاصل این مرحله از الگوریتم چندضلعی محدب $C = F(-B_i)C_1 \cdots C_r(-B_j)$ می‌باشد.

۳) مجموع $C \oplus P_h$ را با روش ارائه شده در [۲] محاسبه می‌کنیم.

۴) بر روی اضلاع $C \oplus P_h$ که متناظر با اضلاع $C_r(-B_j), \cdots, C_1C_2, (-B_i)C_1$ (که در حقیقت مساوی و موازی این اضلاع می‌باشند) هستند کمانی به شعاع $||FB||$ رسم می‌کنیم.

اگر $udisk_{ij}-$ غیر محدب باشد می‌توان آن را به یک نیم دایره و یک تکه محدب از دیسک افراز کرد؛ جمع هریک با P_h را با استفاده از الگوریتم فوق محاسبه کرده و در نهایت اجتماع آن‌ها را محاسبه کرد.

لم ۱ . مجموع مینکوسکی $udisk_{ij} \oplus P_h-$ دارای پیچیدگی خطی $O(p_h)$ می‌باشد و در زمان $O(p_h)$ قابل محاسبه است.

بعد از محاسبه هر یک از $udisk_{ij} \oplus P_h-$ و به‌طور مشابه $ldisk_{ij} \oplus P_h-$ با استفاده از الگوریتم ارائه شده می‌توان با استفاده از روش تقسیم و حل $RFS_i^j = -disk_{ij} \oplus \cup_{h=1}^{l} P_i$ را محاسبه کرد.

قضیه ۳ . $BRFS_i^j$ (مرز RFS_i^j) در زمان $O(n^2 \log^2 n)$ قابل محاسبه است

همان‌طور که مشاهده کردید $BTFS_i$ از مجموعه‌ای از پاره‌خط‌های راست تشکیل شده و $BRFS_i^j$ از پاره‌خط‌های ر است و کمان‌هایی به شعاع $||FA||$ یا $||FB||$. دو نکته ساده و جالب دیگر در مورد $BTFS_i$ و $BRFS_i^j$ قابل ذکر است:

۱) از آن‌جا که $\Re_{\alpha_i} \subset disk_{ij}$ می‌باشد بنابراین $-\Re_{\alpha_i} \oplus P_h \subset -disk_{ij} \oplus P_h$ خواهد بود که به این معنی است که $TFS_i \subset RFS_i^j$ ($BRFS_i^j$، $BTFS_i$ را در برگرفته است) است یا به عبارتی محدوده‌ای که ربات می‌تواند بچرخد زیر مجموعه محدوده‌ای است که ربات می‌تواند انتقال یابد.

۲) از آن‌جا که $disk_{ij} = disk_{ji}$ است، $BRFS_i^j = BRFS_j^i$.

ربات برای تغییر جهت خود از α_i به α_j می‌تواند به اندازه $\alpha_j - \alpha_i$ در جهت عکس عقربه ساعت یا به اندازه $2\pi + \alpha_i - \alpha_j$ در جهت عقربه ساعت بچرخد که همان‌طور که ملاحظه می‌کنید ما فقط به چرخش نوع اول پرداختیم اما این موضوع از کلیت مساله نمی‌کاهد چرا که اگر هر دو چرخش را در نظر بگیریم تنها باید بعد از محاسبه مرز متناظر با هر چرخش اشتراک آن‌ها را محاسبه کرد که این موضوع تغییری در پیچیدگی‌های بدست آمده تا کنون ایجاد نمی‌کند.

۲.۳ ویژگی‌های مسیر بهینه

حال به بررسی ویژگی مسیر بهینه m بین (s, θ_s) و (t, θ_t) $(\theta_s, \theta_t \in \{\alpha_1, \cdots, \alpha_k\})$ می‌پردازیم. چون مسیر m بهینه است بنابراین هر زیر مسیر از آن نیز باید بهینه باشد در نتیجه برای هر صفحه S_i قسمتی از m که داخل این صفحه قرار دارد بهینه می‌باشد و از آن‌جا که در هر صفحه تنها حرکت انتقالی صورت می‌گیرد این زیر مسیر بهینه طبق قضیه ؟؟ از تعدادی پاره‌خط تشکیل شده و رئوس این پاره‌خط‌ها به غیر از نقاطی که در آن‌ها دوران صورت گرفته بر روی رئوس $BTFS_i$ها، s و t قرار دارد. با توجه به آنچه گفته شد مسیر $F_m = m_0 m_1 \cdots m_q$ خواهد بود که $m_r m_{r+1}$ یک پاره‌خط است و $m_0 = s, m_q = t$.

می‌توان به سادگی نشان داد که می‌توان فرض کرد رئوسی مثل m_r که در آن‌ها دوران صورت گرفته است روی $BRFS_i^j$ قرار دارد. به‌طور خلاصه می‌توان قضیه زیر را در مورد مسیر بهینه بیان کرد:

لم ۲ . مسیر بهینه m بین (s, θ_s) و (t, θ_t) $(\theta_s, \theta_t \in \{\alpha_1, \cdots, \alpha_k\})$ را می‌توان به مسیر بهینه m' تبدیل کرد به‌طوری که $F_{m'} = m_0 m_1 \cdots m_q$ خط شکسته‌ای می‌باشد که رئوس آن به غیر از رئوسی که در آن‌ها چرخش صورت می‌گیرد روی رئوس $BTFS_i$ها، s و t قرار دارد و رئوسی که در آن‌ها چرخش صورت می‌گیرد روی $BRTS_i^j$ها (روی یال و راس‌ها) قرار دارد.

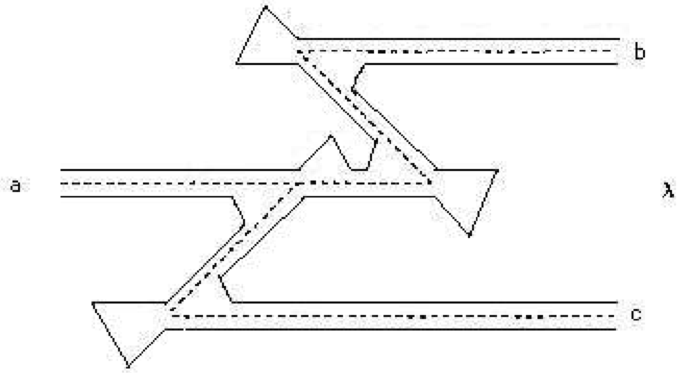
در قضیه زیر ویژگی مسیر بهینه در نقطه‌ای که چرخش صورت می‌گیرد بیان شده است.

قضیه ۴ . اگر m_r راسی از مسیر بهینه باشد که در آن ربات جهت خود را از α_i به α_j تغییر داده است ، در این صورت یکی از سه حالت زیر امکان پذیر است:

۱) m_r یکی از رئوس $BRFS_i^j$ است.

۲) $m_{r-1} m_r$ و $m_r m_{r+1}$ هم راستا هستند.

۳) $m_r m_{r+1}$ منعکس یافته $m_{r-1} m_r$ روی یک یال از $BRFS_i^j$ است.



شکل ۲: محیط پایه‌ای مورد استفاده در کاهش ۴SAT به d_1-optimal

۴ الگوریتم تقریبی

همان‌طور که در بخش ۱ به آن اشاره شد ثابت شده است مساله d_1-optimal در حالت کلی و وقتی F یک نقطه داخلی از ربات باشد یک مساله NP-hard است. در این اثبات برای کاهش ۴SAT به مساله d_1-optimal از محیط‌هایی استفاده شده است (مطابق شکل ۲) که در آن ربات تنها می‌تواند در دو جهت مشخص قرار بگیرد و حرکت انتقالی انجام دهد. هر چند در این محیط موانع غیر محدب هستند اما هر مانع غیر محدب را می‌توان به موانع محدب افراز کرد. بنابراین می‌توانیم بگوییم مساله خاص شده‌ای که در این مقاله مورد بررسی قرار گرفت اگر F یک نقطه داخلی از ربات باشد مساله NP-hard می‌باشد. اگر F بر یکی از دوسر ربات واقع باشد، وضعیت پیچیدگی مساله d_1-optimal در حالت کلی مشخص نیست و هنوز به عنوان یک مساله باز مطرح می‌باشد.

با توجه به این موضوع به بیان یک الگوریتم تقریبی برای حالتی که F بر دو سر ربات باشد خواهیم پرداخت و در ادامه چگونگی تغییر در الگوریتم برای استفاده برای هر نقطه دل‌خواه F روی ربات را بیان می‌کنیم. پارامترهایی که در این الگوریتم تقریبی دخیل هستند عبارتند از: n تعداد اضلاع موانع، k تعداد جهت‌های ربات، L ماکزیمم تعداد بیت‌های هر عدد صحیح، ϵ یک عدد مثبت دل‌خواه.

قبل از بیان الگوریتم لم زیر را که نقش اساسی در الگوریتم دارا می‌باشد را بیان می‌کنیم.

لم ۳ . اگر F یکی از دو سر ربات و $\widehat{CD}$ یک کمان از $BRFS_i^j$ باشد، اشتراک قطاع حاصل از این کمان (به غیر از C و D) با $BTFS_i$ تهی می‌باشد.

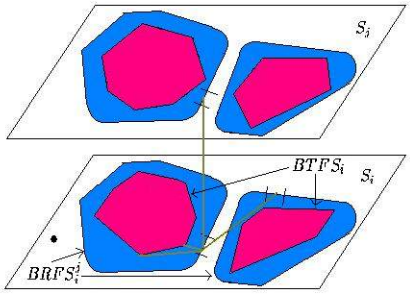
ایده اصلی الگوریتم تقسیم هر یال $BRFS_i^j$ به قطعه‌های مجزا که به جز نقاط ابتدایی و انتهایی اشتراکی ندارند، می‌باشد. الگوریتم شامل:

۱) $BTFS_i$ و $BRFS_i^j$ را برای $i, j = ۱, \cdots, k$ محاسبه می‌کنیم و تعداد کل یال‌های این مرزها را e می‌گیریم.

۲) هر یال E از $BRFS_i^j$ را با معرفی نقاط $b_۰, b_{\pm ۱}, \cdots$ به صورت زیر تقسیم می‌کنیم. ابتدا فرض کنید E یک پاره‌خط باشد:

- نزدیکترین نقطه به s روی E را بدست می‌آوریم و آن را با $b_۰$ نمایش می‌دهیم و این فاصله را با $D(s, E)$ نمایش می‌دهیم (این نقطه دو سر پاره‌خط E یا تقاطع خط عمود از s به E و E می‌باشد).
- $b_۱$ را روی E به فاصله $D(s, E)$ از $b_۰$ در نظر می‌گیریم (b_hها در یک طرف $b_۰$ می‌باشند و قرینه آن‌ها نسبت به $b_۰$ را با b_{-h} نمایش می‌دهیم).
- بازه $[b_۰, b_۱]$ را به بازه‌هایی به طول $D(s, E) * \epsilon_۱$ تقسیم می‌کنیم و نقاط حاصل را $q_۱, q_۲, \cdots$ می‌نامیم ($\epsilon_۱ = \frac{\epsilon}{۸e}$).
- نقطه b_h روی E را به فاصله $D(s, E) * (۱ + \epsilon_۱)^{h-۱}$ از $b_۰$ در نظر می‌گیریم.

اگر E یک کمان و نقطه s خارج دایره‌ای باشد که شامل کمان E است (شعاع این دایره $r = ۱$ می‌باشد)



شکل ۳: فضاهای ممنوعه و گراف رویت

- نزدیکترین نقطه به s روی E را بدست می‌آوریم و آن را با b_0 نمایش می‌دهیم و این فاصله را با $D(s,E)$ نمایش می‌دهیم (این نقطه دو سر کمان E یا تقاطع خط عمود از s بر دایره‌ای است که E را شامل شده).
- b_1 را روی E به فاصله $D(s,E)$ از b_0 در نظر می‌گیریم (b_hها در یک طرف b_0 می‌باشند و قرینه آن‌ها نسبت به خط عمود از s بر دایره را با b_{-h} نمایش می‌دهیم).
- بازه $[b_0, b_1]$ را به بازه‌هایی به طول $D(s,E) * \epsilon_1$ تقسیم می‌کنیم و نقاط حاصل را $q_1, q_2, \cdots$ می‌نامیم.
- نقطه b_h روی E را به فاصله $D(s,E) * (1+\epsilon_1)^{h-1}$ از b_0 در نظر می‌گیریم.
- کمان $b_h b_{h+1}$ را به کمان‌هایی به طول $||\widehat{b_h b_{h+1}}||\frac{r}{2}$ تقسیم می‌کنیم.

و اگر E یک کمان و نقطه s داخل دایره‌ای باشد که شامل کمان E است مانند حالت قبل عمل می‌کنیم و تنها در مرحله آخر آن کمان $b_h b_{h+1}$ را به کمان‌هایی به طول $||\widehat{b_h b_{h+2}}||\frac{r}{4}$ تقسیم می‌کنیم.

۳) گراف رویت G را می‌سازیم (شکل ۳). رئوس این گراف، رئوس $BTFS_i$ها، نقطه s در صفحه مبدا و نقطه t در صفحه مقصد و پاره‌خط‌هایی است که از مرحله قبل بدست آمده است. دو پاره‌خط σ_1 و σ_2 از یک صفحه یکدیگر را می‌بینند اگر $x_1 \in \sigma_1$ و $x_2 \in \sigma_2$ وجود داشته باشند که این دو یکدیگر را ببینند (رئوس $BTFS_i$ها، s و t را پاره‌خط‌هایی به طول صفر در نظر می‌گیریم).

۴) به هر یال گراف که رئوس متناظر با σ_1 و σ_2 را به‌هم وصل کرده فاصله بین وسط این دو پاره‌خط را به عنوان وزن آن نسبت می‌دهیم.

۵) پاره‌خط‌های متناظر در دو صفحه را به یکدیگر وصل می‌کنیم (رئوس متناظر با آن‌ها در گراف G) و به آن‌ها وزن صفر را نسبت می‌دهیم.

۶) در گراف G کوتاهترین مسیر بین مبدا و مقصد را با استفاده از الگوریتم دایکسترا بدست می‌آوریم.

۷) اگر خروجی الگوریتم $\sigma_1, ..., \sigma_q$ باشد مسیر مورد نظر به‌صورت زیر خواهد بود:

$$s, x_1, y_1, \cdots, x_q, y_q, t$$

که $x_h, y_h \in \sigma_h$ و (y_h, x_{h+1}) همدیگر را می‌بینند همین طور (s, x_1) و (y_q, t).

لم ۴.

۱) طول پاره‌خط σ که از تقسیم یال E بدست آمده در رابطه زیر صدق می‌کند: $l(\sigma) \leq \epsilon_1 D(s,\sigma)$

۲) تعداد نقاطی که از تقسیم یال E بدست می آید از $O(\frac{L}{\log(1+\epsilon_1)})$ می‌باشد و تعداد راس‌های گراف از $O(\frac{eL}{\log(1+\epsilon_1)})$ است.

۳) $e = O(k^2 n^2)$.

قضیه ۵. حرکت حاصل (m) از الگوریتم ارائه شده بین دو موقعیت (s, θ_s) و (t, θ_t) در رابطه زیر صدق می‌کند

$$d_۱(m) \leq (۱ + \epsilon) d_۱((s, \theta_s), (t, \theta_t))$$

و زمان اجرای این الگوریتم $O(\frac{n^۶ k^۳ L^۲}{\log^۲(۱+\epsilon_۱)})$ می‌باشد.

اگر F دو سر ربات نباشد می‌توان با اضافه کردن یک مرحله بین مرحله ۱ و ۲ الگوریتم فوق، این الگوریتم را برای این حالت نیز استفاده کرد. این مرحله شامل افراز هر کمان در مرز به کمان‌هایی است که قطاع هر کمان با $BTFS_i$ اشتراک داخلی نداشته باشد. به این منظور باید برای هر کمان رئوسی از $BTFS_i$ که داخل قطاع قرار دارد را پیدا کرده و از بین آن‌ها راسی که زاویه خط گذرنده از آن و ابتدای کمان کمترین مقدار یا بیشترین مقدار (بستگی به جهت دارد) را دارد را انتخاب کرده و این خط را رسم کرد تا کمان را در نقطه‌ای دیگر قطع کند حال یک کمان داریم که اشتراک داخلی آن با $BTFS_i$ تهی می‌باشد؛ قسمت باقی مانده کمان اصلی را نیز با همین روش و به‌طور بازگشتی افراز می‌کنیم (توجه کنید با این کار مقدار e افزایش پیدا خواهد کرد).

۵ نتیجه‌گیری

در این مقاله به بررسی حرکت بهینه ربات میله‌ای با قابلیت جابه‌جایی و جرخش تحت زاویه‌های مشخص حول یک نقطه ثابت اما دل‌خواه F روی ربات پرداخته شد. هدف در این مساله پیدا کردن حرکتی از موقعیت ابتدایی به موقعیت نهایی است به‌طوری که مسافت طی‌شده توسط F مینیمم شود. در این مقاله ضمن توصیف مسیر بهینه، یک الگوریتم تقریبی با ضریب تقریب $۱ + \epsilon$ برای حل این مساله ارائه شد.

مراجع

[1] T. Asano, D. Kirkaptric and C. Y. Yap, $d_۱$ *optimal motion for a rod*, In Proc. 12th Annu. ACM Sympos. Comput. Geom., 252-263, 1996.

[2] M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry : Algorithms and its Applications*, Springer, 1997.

[3] C. Icking, G. Rote, E. Welzl and C. Yap, *Shortest paths for line segment*, Algorithmica, 10: 182-200, 1993.

[4] J. O'Rourke, *Finding a shortest ladder path : a special case*, IMA Preprint Series 353, Institute for Mathematic and its Applications, University of Minnesota, 1987.

[5] C. H. Papadimitriou and E. B. Silverberg, *Optimal piecewise linear motion of an object among obstacles*, Algorithmica, 2:523-287, 1987.