

# مسیریابی از میان موانع جابه‌جاشونده

شکرانه خشخاشی مقدم<sup>۱\*</sup>، الپس مسیحی<sup>۲</sup>

دانشگاه تربیت مدرس

تاریخ دریافت مقاله: ۱۳۹۳/۰۸/۲۴

تاریخ پذیرش مقاله: ۱۳۹۳/۱۰/۰۸

## چکیده

مسئله برنامه‌ریزی حرکت با موانع جابه‌جاشونده، (NAMO)<sup>۳</sup>، عبارت از یافتن مسیرهایی بدون تصادم برای روبات است؛ این در حالی است که روبات برای یافتن یک مسیر، می‌تواند برخی موانع را جابه‌جا کند. NAMO یک مسئله NP-Complete و در زمره مسائلی از مسیریابی حرکت روبات قرار می‌گیرد که دارای محیط‌های متغیر هستند. در این حوزه یک برنامه بهینه برای روبات می‌تواند با توجه به فاکتورهای مختلف هم‌چون طول مسیره‌های انتقال و جابه‌جایی، تعداد اجسام جابه‌جاشونده، تعداد دفعات جابه‌جایی اجسام و زمان تعیین شود. در این مقاله با استفاده از مفاهیمی هم‌چون گراف دید نگار<sup>۴</sup>، و عمق نفوذ<sup>۵</sup>، الگوریتم بازگشتی ارائه شده قادر است مسائل مختلف NAMO را در زمان معقولی حل کند. هم‌چنین به‌کارگیری الگوریتم پیشنهادی برای حل برخی مسائل موجود در ادبیات، موجب کاهش چشمگیر تعداد اجسام جابه‌جا شده و تعداد دفعات جابه‌جایی اجسام جابه‌جاشونده شده است.

**واژه‌های کلیدی:** مسیریابی حرکت روبات، موانع جابه‌جاشونده، گراف دیدنگار، عمق نفوذ، جمع مینکوفسکی.

## ۱- مقدمه

عبارت از یافتن مسیرهایی بدون تصادم برای روبات است به‌صورتی که روبات، بتواند برای یافتن یک مسیر، برخی موانع (اجسام جابه‌جاشونده) را جابه‌جا کند. راه‌حل نهایی باید به‌صورت برنامه‌ریزی حرکتی باشد که عملیات حرکت روبات، گرفتن و جابه‌جایی اجسام را به شکل مکرر انجام می‌دهد تا روبات به هدف خود دست پیدا کند.

یکی از انگیزه‌های موجود برای پرداختن به مسئله NAMO، امنیت جان انسان‌ها است. عملیات امداد و نجاتی را تصور کنید که به‌منظور نجات آسیب‌دیدگان یک فاجعه طبیعی همانند طوفان یا زلزله انجام می‌گیرد. در چنین شرایطی احتمال وقوع حوادثی هم‌چون ریزش ساختمان‌های متزلزل و یا انتشار گازهای سمی و انفجار، جان امدادگران را به خطر انداخته و لذا استفاده از روبات به‌جای انسان، اهمیت

به‌طور مرسوم، برنامه‌ریزی حرکت روبات از یک نقطه شروع تا یک نقطه پایان، به دنبال یافتن مسیری عاری از مانع است. این در حالی است که بسیاری از مسائل موجود در دنیای واقعی، نیازمند توانایی روبات در جابه‌جایی اجسام هستند. مسئله برنامه‌ریزی حرکت با اجسام جابه‌جاشونده

\*۱- کارشناسی ارشد مهندسی صنایع، دانشگاه تربیت مدرس، نویسنده پاسخگو، پست‌الکترونیکی: sh.k.moghaddam@gmail.com، نشانی: تهران، میدان صنعت، خیابان هرمزان، هرمزان چهارم، پلاک ۱۸، کدپستی: ۱۴۶۶۶۱۳۶۴۱.

۲- استادیار بخش مهندسی صنایع، دانشگاه تربیت مدرس، پست‌الکترونیکی: masehian@modares.ac.ir

3- Navigation Among Movable Obstacles

4- Visibility Graph, VG

5- Penetration Depth, PD

فرمول‌بندی مسائل  $LP_1$  به شکل گرافی از اجزای ناهم‌بند فضای آزاد، یک راه‌حل ابتکاری کامل از نظر تفکیک‌پذیری<sup>۳</sup> به‌دست آمده است. برنامه‌ریز ارائه شده در [۵] می‌توانست مسائلی را که برنامه‌ریز CH قادر به حل آنها نبود حل کند و با موفقیت بر روبات انسان‌نمای HRP-2 به‌کار گرفته شد [۶]. تحقیقات انجام شده پس از آن، حرکات مرتبط چندین جسم را به‌صورت هم‌زمان مدنظر قرار داد [۷، ۸] (حل مسائل  $LP_k$ ) که در نهایت منجر به ارائه یک الگوریتم کامل احتمالاتی<sup>۴</sup> برای حل مسائل NAMO گردید [۹].

در [۱۰] برای اولین بار مسائل NAMO از محیط شناخته شده به محیط‌های ناشناخته تعمیم داده شدند. محیطی که الگوریتم ارائه شده در این مقاله در آن پیاده‌سازی شده است فقط شامل اجسام مستطیل شکل بوده و روبات نیز فقط می‌توانست اجسام را با هل دادن جابه‌جا کند. با وجود اینکه الگوریتم ارائه شده از نظر عملکرد بهینه بود، با در دست داشتن اطلاعات موجود، مسیریابی بهینه را تضمین نمی‌کرد. در [۱۱] نیز مسئله NAMO در محیط‌های ناشناخته بررسی شده است، با این تفاوت که در الگوریتم ارائه شده در این مقاله محدودیتی برای شکل اجسام و عملی که بر یک جسم انجام می‌شود وجود ندارد. در [۱۱] جسمی که یک‌بار جابه‌جا شده، امکان جابجایی مجدد آن وجود ندارد و همین امر سبب می‌شود تا برنامه‌ریز نتواند با توجه به اطلاعات محدود در دسترس از محیط اطراف، به جواب بهینه دست پیدا کند.

## ۲- مفروضات و مفاهیم

وجود اجسام جابه‌جا شونده می‌تواند اثر قابل‌توجهی بر برنامه‌ریزی حرکت روبات داشته باشد. حتی زمانی که هیچ مسیر ممکن برای روبات در بین دو پیکربندی موجود در یک چیدمان خاص از فضای کاری، وجود ندارد، روبات می‌تواند یک مسیر را با انتقال یک یا چند جسم جابه‌جا شونده به سمت هدف خود به وجود آورد و با وجود انسداد اولیه مسیر، راهی به سمت هدف بیابد.

پیدا می‌کند. این در حالی است که در جریان این‌چنین بلایای طبیعی، ساختمان‌های مخروبه، خرده‌سنگ و آوار، مسیرهایی را که پیش از این قابل تردد بوده‌اند، مسدود می‌کنند. با حل مسئله NAMO روبات‌ها محیط اطراف خود را مورد بررسی قرار داده و تصمیم می‌گیرند که کدام اجسام بایستی جابه‌جا شوند و مسیر موجود را برای پیشروی بیشتر باز می‌کنند.

در [۱] اثبات شده است که مسئله مسیریابی از میان اجسام جابه‌جا شونده یک مسئله NP-Complete است. پس از آن در [۲] نشان داده شد که نوع ساده‌شده این مسئله که در آن اجسام به شکل مربع‌هایی به طول ضلع یک واحد هستند نیز مسئله‌ای NP-Complete به شمار می‌رود. در [۳] اولین برنامه‌ریزی که چندین جسم جابه‌جا شونده و یک هدف را به‌صورت هم‌زمان در نظر می‌گرفت، ابداع شده که به برنامه‌ریز CH معروف است. ابتدا مجموعه‌ای از زیر اهداف را تولید کرده و پس از آن برای رسیدن به هر زیر هدف تلاش می‌کند. با این حال، این برنامه‌ریز نمی‌توانست مسائلی را که در آنها ترتیب جابه‌جایی اجسام در به‌دست آوردن جواب نهایی مهم باشد را حل کند.

در [۴] برنامه‌ریزی برای روبات انسان‌نما ارائه شد که قادر بود مسیری را از یک نقطه آغازین مشخص تا یک نقطه پایانی بیابد و در حین مسیریابی، اجسام را نیز جابه‌جا کند. برای ایجاد راه‌حل نهایی از سه برنامه‌ریز وظایف عمل‌وری محیط، حرکات مسیریابی<sup>۱</sup> و حرکات عمل‌وری<sup>۲</sup> استفاده شده بود. این برنامه‌ریز به هیچ عنوان بهینه بودن مسیر نهایی را تضمین نمی‌کرد. هم‌چنین جابه‌جایی اجسامی که به شکل غیرمستقیم در مسیر روبات واقع شده بودند نیز مدنظر قرار نگرفته بود.

برنامه‌ریز ارائه شده در [۵] قادر به حل زیرگروهی از مسائل NAMO با نام مسائل  $LP_1$  بود. در مسائل  $LP_1$  اجزای غیر هم‌بند فضای آزاد، می‌توانند فقط با جابه‌جایی یک جسم به یکدیگر متصل شوند. رویکرد موجود با در نظر گرفتن پیچیدگی وظیفه مسیریابی نسبت به ابعاد فضای محیط، فضای جستجوی مسائل NAMO را کاهش داد. با

3- Resolution Complete  
4- Probabilistically Complete

1- Navigation Motion Planner  
2- Manipulation Motion Planner

## ۲-۱- مفاهیم و اصطلاحات مربوط به تعریف فضای

### مسئله

موانع ثابت<sup>۱</sup> ( $B_i$ ): به هیچ عنوان امکان جابه‌جایی نداشته و به نوعی جزئی از محیط ایستا محسوب می‌شوند.

اجسام جابجا شونده ( $M_j$ ): خودبه‌خود قابلیت حرکت و جابه‌جایی ندارند و در صورتی می‌توانند حرکت کنند که روبات، آنها را گرفته و با خود جابه‌جا کند. در غیر این صورت ساکن می‌مانند.

فضای آزاد ( $C_{free}$ ): فضای آزاد در  $C$  طبق رابطه (۱) عبارت است از مجموعه پیکربندی‌هایی که در آنها روبات با هیچ یک از موانع تداخل ندارد.

$$C_{free} = C \setminus \left( \bigcup_i CB_i \right) \quad (1)$$

فضای تماسی  $xl(C_{free})$  مجموعه پیکربندی‌هایی است که در آنها روبات بدون هیچ نفوذی به داخل اجسام با یک یا چند جسم تماس پیدا می‌کند.

فضای پیکربندی مانع ( $C-obstacle$ ): این فضا در حقیقت متمم  $C_{free} \cup xl(C_{free})$  است. اگر یک پیکربندی در  $C-obstacle$  واقع شده باشد، روبات با یک یا چند جسم تقاطع داشته و در حقیقت در آن اجسام نفوذ می‌کند. (رابطه ۲)

$$CB = \{q | A(q) \cap B \neq \emptyset\} \quad (2)$$

در این مقاله فضای کاری (فضای پیکربندی اجسام ثابت و جابه‌جاشونده) و وضعیت پیکربندی‌های ابتدایی و انتهایی روبات از قبل مشخص بوده و به‌صورت دو بعدی و پیوسته است. روبات قادر است اجسام را صرفاً با حرکات انتقالی جابه‌جا کند و قابلیت چرخش اجسام را ندارد. جسمی که یک‌بار جابه‌جا شده می‌تواند مجدداً جابه‌جا شود و در نهایت روبات در هر لحظه فقط توان جابه‌جایی یک جسم را دارد.

## ۳- الگوریتم پیشنهادی برای حل مسائل NAMO

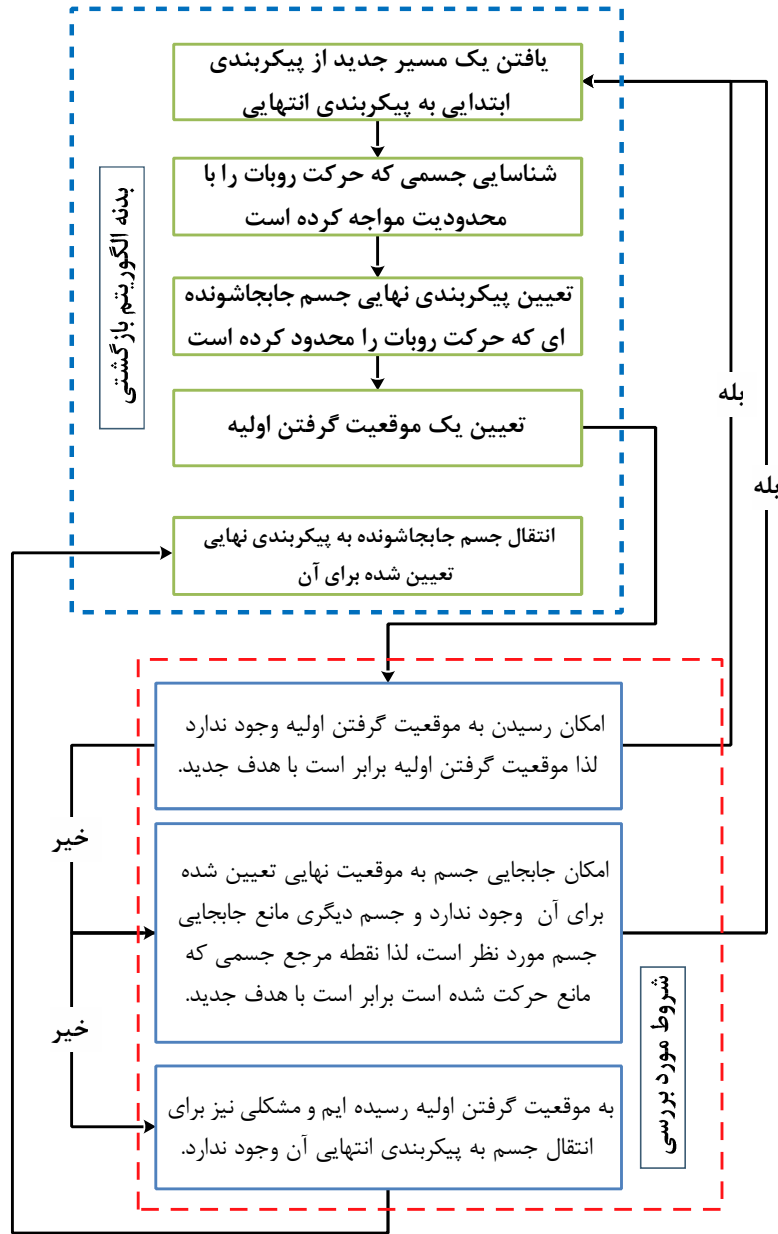
یکی از مشکلات موجود در حوزه NAMO که در حقیقت یکی از اهداف اصلی این مقاله نیز محسوب می‌شود، یافتن یک روش ابتکاری مناسب برای کاهش تعداد مکان‌های کاندید اجسام جابه‌جاشونده از میان فضای بسیار بزرگ جستجو است. در مقاله پیش رو، با الهام از برخی تحقیقات

انجام شده، هر مسئله NAMO به زیر مسائل مختلفی تقسیم شده و توانایی برنامه‌ریز در حل هر یک از این زیر مسائل، مورد بررسی قرار گرفته است. از این‌رو در راستای حل این مسئله یک الگوریتم بازگشتی طراحی شده است که با استفاده از رویکرد تقسیم و حل مسئله اصلی را به زیر مسائل کوچک‌تر تجزیه کرده و پس از حل هر یک از زیر مسائل موجود راه‌حل‌های ایجاد شده را برای رسیدن به جواب با یکدیگر ترکیب می‌کند.

همان‌گونه که در شکل (۱) نشان داده شده است، در هر مرتبه فراخوانی این الگوریتم بازگشتی، با به‌روزرسانی موقعیت اولیه و انتهایی روبات، فقط برای تعیین مکان نهایی یک جسم برنامه‌ریزی می‌شود. به این ترتیب که برنامه‌ریز ابتدا مسیری به سمت هدف خود می‌یابد و سپس در صورت تداخل مسیر انتخاب شده با جسم مورد نظر، تلاش می‌کند تا مسیر حرکتی خود را آزاد سازد. در الگوریتم بازگشتی طراحی شده برای حل مسئله مسیریابی از میان موانع جابه‌جاشونده گام‌های زیر مرتباً تکرار می‌شوند:

۱. تعیین پیکربندی ابتدایی و انتهایی روبات.
۲. یافتن مسیر از پیکربندی آغازین به سمت پیکربندی هدف.
۳. شناسایی جسمی که حرکت روبات را با محدودیت مواجه کرده است.
۴. تعیین پیکربندی نهایی جسم جابه‌جاشونده‌ای که حرکت روبات را محدود کرده است.
۵. تعیین یک موقعیت گرفتن اولیه.
۶. انتقال جسم جابه‌جاشونده به پیکربندی نهایی تعیین شده برای آن، توسط روبات.

از میان گام‌های فوق، گام‌های ۲ و ۴ (یافتن مسیر و تعیین پیکربندی نهایی جسم جابه‌جاشونده) از اهمیت بیشتری برخوردار هستند. با توجه به پیوسته در نظر گرفتن فضای کاری روبات در این مقاله، از گراف دیدنگار به‌عنوان نقشه مسیر<sup>۲</sup> استفاده شده است. سپس با توجه به مسیر انتخاب شده به سمت پیکربندی هدف، مفهوم عمق نفوذ به ما کمک می‌کند تا موقعیت نهایی اجسام جابه‌جاشونده‌ای که مسیر روبات به سمت هدف را مسدود می‌کنند تعیین گردد.



شکل (۱): مدل کلی الگوریتم بازگشتی

## ۳-۱- گراف دیدنگار

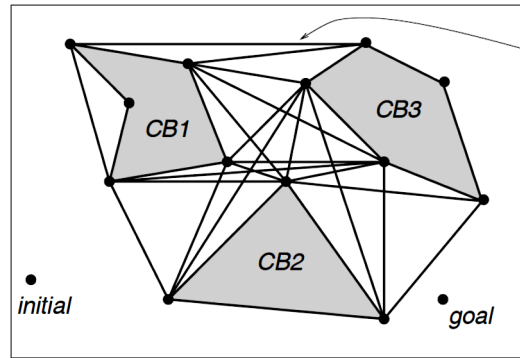
گراف دیدنگار یک نقشه مسیر است. مشخصه‌های یک گراف دیدنگار این است که گره‌های آن در صورتی که در راستای دید یکدیگر باشند با یک یال به هم متصل می‌شوند و همچنین تمام نقاط موجود در فضای آزاد پیکربندی روبات در دیدرس حداقل یک گره از گره‌های گراف دیدنگار هستند. گراف دیدنگار استاندارد در یک فضای پیکربندی دو بعدی شامل چندضلعی‌های متعدد تعریف می‌شود. گره‌های گراف دیدنگار،  $v_i$  شامل پیکربندی ابتدایی روبات، پیکربندی نهایی

روبات و تمام رئوس  $C$ -obstacle‌ها است. طبق رابطه (۳) یال‌های این گراف،  $e_{ij}$ ، پاره‌خط‌های مستقیمی هستند که دو گره در دیدرس یکدیگر،  $v_i$  و  $v_j$ ، را به هم متصل می‌کنند [۱۲].

$$e_{ij} \neq 0 \Leftrightarrow [sv_i + (1-s)v_j] \in cl(C_{free}), \forall s \in [0,1] \quad (3)$$

به عبارت دیگر پاره‌خطی که این دو گره را به یکدیگر متصل می‌کند به صورت کامل در فضای آزاد پیکربندی روبات

واقع شده است. بایستی توجه داشت که اضلاع چندضلعی‌ها نیز به‌عنوان یال‌های گراف دیدنگار در نظر گرفته می‌شوند.



شکل (۲): نمایی از یک گراف دید نگار [۱۲].

ساخت گراف دیدنگار می‌تواند در  $O(n^2 \log(n))$  انجام شود. پس از ساخت گراف دیدنگار می‌توان کوتاه‌ترین مسیر موجود میان پیکربندی‌های ابتدایی و انتهایی روبات را با استفاده از روش‌های جستجوی هم‌چون  $A^*$  و یا دایکسترا یافت. در شکل (۲) یک گراف دیدنگار نمایش داده شده است.

در مسئله مسیریابی از میان اجسام جابه‌جاشونده، اجسام موجود در فضای کاری روبات شامل اجسام جابه‌جاشونده و اجسام ثابت هستند. هم‌چنین نحوه قرارگیری اجسام جابه‌جاشونده در این فضا در ابتدای امر به شکلی است که هیچ مسیر آزادی برای رسیدن به موقعیت هدف وجود ندارد، به این معنی که اگر رئوس  $C$ -obstacle تمام اجسام موجود (اعم از ثابت و جابه‌جاشونده) برای ساخت گراف دیدنگار مدنظر قرار گیرد، گراف ساخته‌شده هم‌بند نخواهد بود و لذا نمی‌توان مسیری عاری از تداخل برای روبات پیدا کرد.

در روش به‌کار گرفته‌شده در این مقاله، گره‌های گراف دیدنگار، شامل پیکربندی ابتدایی روبات، پیکربندی نهایی روبات و رئوس تمام  $C$ -obstacle های موجود، اعم از اجسام ثابت و جابه‌جاشونده است. در راستای اتصال گره‌ها به یکدیگر، فرض می‌شود که اجسام جابه‌جاشونده در فضای کاری روبات وجود ندارند. در نظر نگرفتن اجسام جابه‌جاشونده در محیط، باعث می‌شود که یال‌های عبوری از داخل اجسام جابه‌جاشونده نیز در گراف دیدنگار مدنظر قرار گیرند.

شکل (۳) نمایی از یک گراف دیدنگار را قبل و بعد از در نظر نگرفتن جسم جابه‌جاشونده در محیط، نشان می‌دهد. همان‌طور که در قسمت (الف) شکل مشخص است نمی‌توان هیچ مسیری از پیکربندی ابتدایی تا پیکربندی انتهایی یافت.

### ۳-۲- عمق نفوذ

عمق نفوذ (PD) یک معیار، فاصله برای تعیین میزان تقاطع بین اجسام متقاطع است. با در اختیار داشتن میزان تقاطع روبات با موانع موجود در مسیر، عمق نفوذ این امکان را در اختیار ما می‌گذارد تا فاصله هر پیکربندی متداخل (پیکربندی از روبات که در داخل  $CB$  واقع شده است) را تا نزدیک‌ترین پیکربندی آزاد (پیکربندی از روبات که در  $C_{free}$  واقع شده است) محاسبه کند.

در صورت در اختیار داشتن دو چندضلعی محدب  $P$  و  $Q$  و یک جهت  $d$  آنگاه  $\sigma_d(P, Q)$ ، نشان‌دهنده کوتاه‌ترین مسافتی است که در آن  $Q$  بایستی در راستای  $d$  جابه‌جا شود تا مناطقی از  $P$  و  $Q$  که به داخل یکدیگر نفوذ کرده‌اند از یکدیگر جدا شوند.  $\sigma_d(P, Q)$  عمق نفوذ بین دو چندضلعی  $P$  و  $Q$  در راستای  $d$  نامیده می‌شود. حداقل عمق نفوذ،  $\sigma_d(P, Q)$ ، بین  $P$  و  $Q$  در حقیقت حداقل  $\sigma_d(P, Q)$  با در نظر گرفتن تمام جهات ممکن،  $d$ ، است.  $\sigma(P, Q)$  با نام عمق نفوذ انتقالی<sup>۱</sup> نیز شناخته می‌شود [۱۳].

بیشتر فعالیت‌های انجام‌گرفته در حوزه PD بر عمق نفوذ انتقالی ( $PD^t$ ) تمرکز دارند. برای دو جسم متقاطع  $A$  و  $B$ ،  $PD^t$  با توجه به رابطه (۴) به شکل زیر تعریف می‌شود: کمترین انتقال اعمال‌شده بر یکی از اجسام ( $A$ ) برای جداسازی آن، از جسم دیگر  $(B)$  [۱۴].

$$PD^t(A, B) = \min\{\|d\| \mid \text{interior}(A+d) \cap B = \emptyset\} \quad (4)$$

در راستای اندازه‌گیری  $PD^t$ ، الگوریتم‌های بسیار خوبی بین اجسام محدب و غیر محدب موجود است. بیشتر این الگوریتم‌ها بر پایه فرمول‌بندی Minkowski Sum پایه‌ریزی شده‌اند. در حقیقت محاسبه  $PD^t$  برابر است با یافتن نزدیک‌ترین نقطه بر سطح Minkowski Sum تا نقطه مرجع [۱۳]. اگر مجموعه‌ای از بردارهای  $A$  و  $B$  موجود

1- Translational penetration depth

یعنی Minkowski Sum پاره خط  $L1$  و قرینه فضای پیکربندی روبات حول  $M1$  نسبت به نقطه مرجع  $M1$  نشان داده شده است. در قسمت (ج) با توجه به Minkowski Sum حاصل شده کمترین فاصله تا مرز Minkowski Sum که همان حداقل عمق نفوذ ( $d$ ) محاسبه شده است. قسمت (د) نیز نشان‌دهنده  $d \oplus M1$  یا همان Minkowski Sum جسم  $M1$  و جهت  $d$  است.

#### ۴- نتایج محاسباتی

در این مقاله نمونه مسائل برگرفته از ادبیات، توسط نرم‌افزار MATLAB شبیه‌سازی شده‌اند. فقط فاکتورهایی که امکان مقایسه آنها با مقادیر مشابه موجود در ادبیات مسئله NAMO وجود دارد، مانند تعداد اجسام جابه‌جا شده، تعداد کل دفعات جابه‌جایی اجسام و زمان حل مسئله هستند. همان‌طور که مشاهده می‌شود با استفاده از روش حل ارائه شده در این مقاله، حداقل یکی از فاکتورهای مذکور نسبت به فاکتور متناظر موجود در ادبیات، بهبود یافته است.

باشند، Minkowski Sum آنها طبق رابطه مجموعه برابر است با:

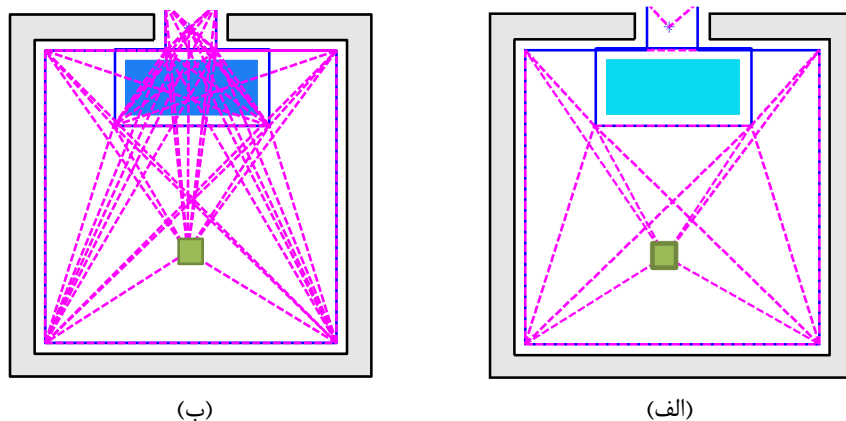
$$A \oplus B = \{a+b | a \in A, b \in B\} \quad (5)$$

همان‌طور که در شکل (۴) نشان داده شده است دو مفهوم فضای پیکربندی مانع و Minkowski Sum در حقیقت خروجی یکسانی دارند.

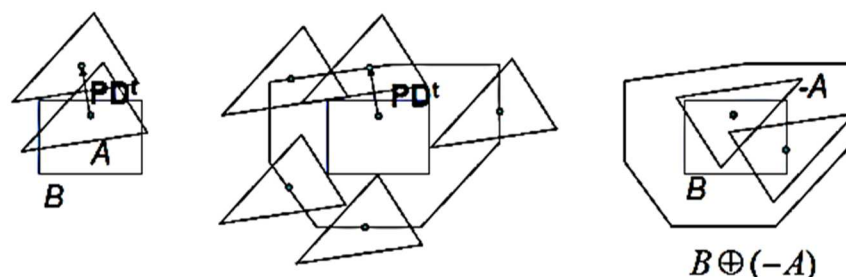
#### ۳-۳- استفاده از گراف دیدنگار و عمق نفوذ در راستای حل مسائل NAMO

با توجه به توضیحات ارائه شده بخش‌های قبیل، در ارتباط با عمق نفوذ و Minkowski Sum، مشخص است که در صورت تقاطع قسمتی از مسیر انتخاب‌شده به سمت پیکربندی نهایی با یک یا چند جسم جابه‌جا شونده، می‌توان این تقاطع را با کمترین جابه‌جایی، برطرف کرد.

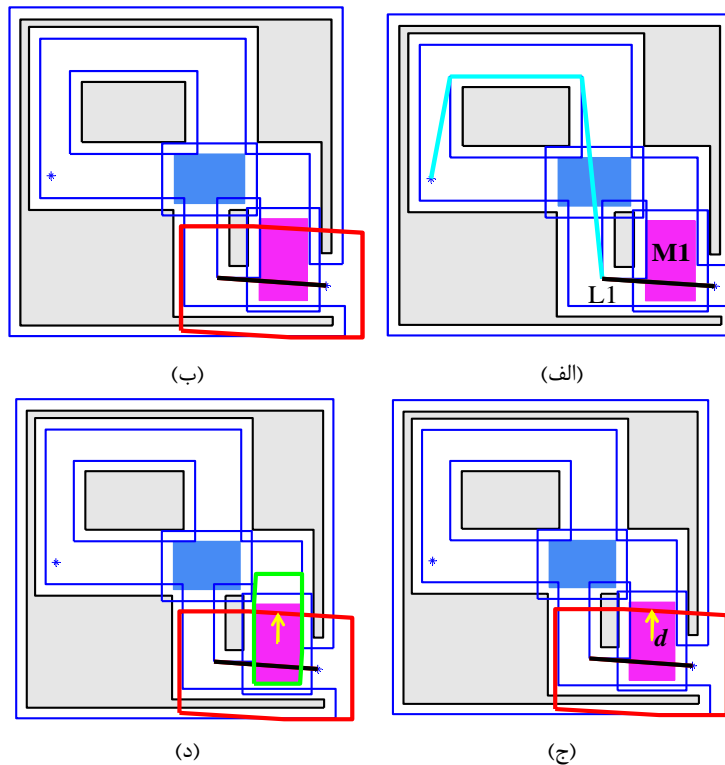
در شکل (۵) قسمت (الف) یک فضای کاری با دو جسم جابه‌جا شونده نمایش داده شده است و پاره خط  $L1$  از مسیر انتخاب‌شده با جسم  $M1$  که آخرین جسم موجود بر مسیر انتخابی است، تقاطع دارد. در قسمت (ب)  $L1 \oplus (-CB_{M1})$



شکل (۳): (الف) نمایی از یک گراف دیدنگار ناهم‌بند. (ب) نمایی از یک گراف دیدنگار هم‌بند.



شکل (۴): نمایی از تقاطع دو جسم  $A$  و  $B$ ، فضای پیکربندی مانع زمانی که روبات جسم  $A$  فرض شود و Minkowski Sum قرینه جسم  $A$  حول  $B$  [۱۴].

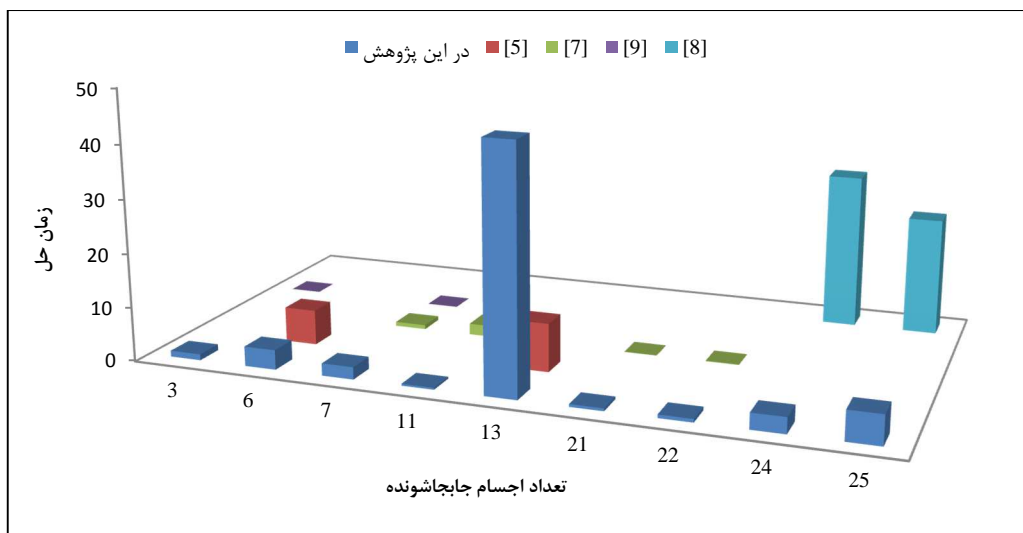


شکل (۵): پاره خط  $L1$  با جسم جابه‌جاشونده  $M1$  تقاطع دارد. (ب).  $(-CB_{M1}) \cdot L1$ . (ج). کمترین میزان جابه‌جایی برای جداسازی  $M1$  و  $L1$  با  $d$  نمایش داده شده است. (د).  $d \oplus (M1)$ .

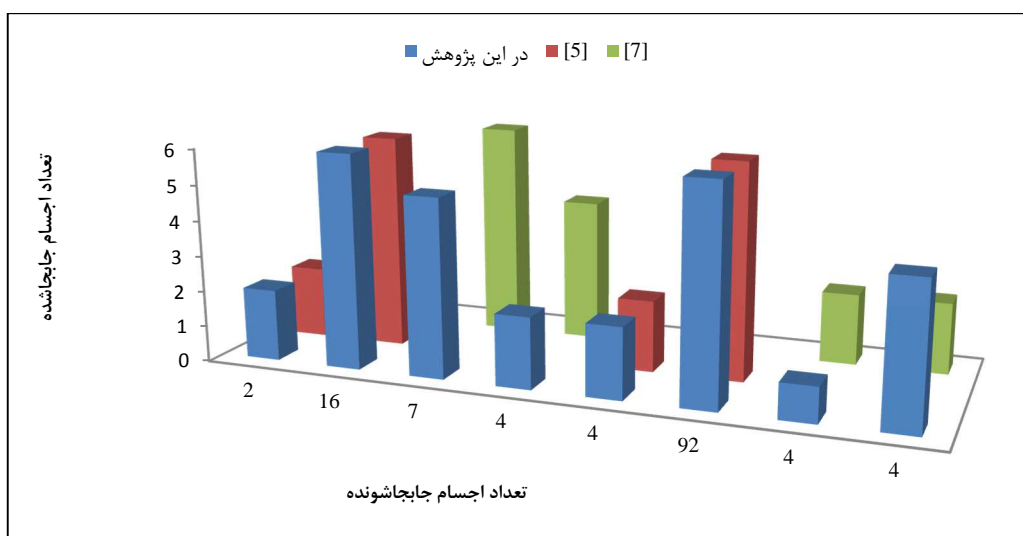
جدول (۱): نتایج حل مسائل نمونه موجود در مقالات

تعداد کل دفعات جابجایی				تعداد اجسام جابجا شده			زمان (ثانیه)					تعداد اجسام جابجا شونده
d	c	b	a	c	b	a	e	d	c	b	a	
N/G	N/G	۲	۲	N/G	۲	۲	N/G	N/G	N/G	N/G	۰,۴۰۹	۲
N/G	۲۱۴	N/G	۶	N/G	N/G	۳	N/G	۰,۰۱	N/G	N/G	۱,۱۲۲	۳
N/G	N/G	۶	۷	N/G	۶	۶	N/G	N/G	N/G	۶,۵	۳,۶۶۹	۱۶
۶	۲۱۰۰	N/G	۶	۶	N/G	۵	N/G	۰,۰۳	۰,۷۷	N/G	۲,۲۹۶	۷
N/G	N/G	N/G	۲	N/G	N/G	۲	N/G	N/G	N/G	N/G	۰,۴۰۰	۱۰
۴	N/G	N/G	۲	۴	N/G	۲	N/G	N/G	۲,۰۸	N/G	۰,۳۵۲	۴
N/G	N/G	۲	۲	N/G	۲	۲	N/G	N/G	N/G	N/G	۰,۸۰۹	۴
N/G	N/G	۶	۶	N/G	۶	۶	N/G	N/G	N/G	۹	۴۷,۸۴۴	۹۲
۲	N/G	N/G	۱	۲	N/G	۱	N/G	N/G	۰,۰۵	N/G	۰,۵۰۹	۴
۲	N/G	N/G	۴	۲	N/G	۴	N/G	N/G	۰,۱۰	N/G	۰,۶۴۰	۴
N/G	N/G	N/G	۲	N/G	N/G	۲	۲۹	N/G	N/G	N/G	۲,۹۷۸	۲۷
N/G	N/G	N/G	۶	N/G	N/G	۶	۲۲	N/G	N/G	N/G	۵,۴۶۷	۲۲

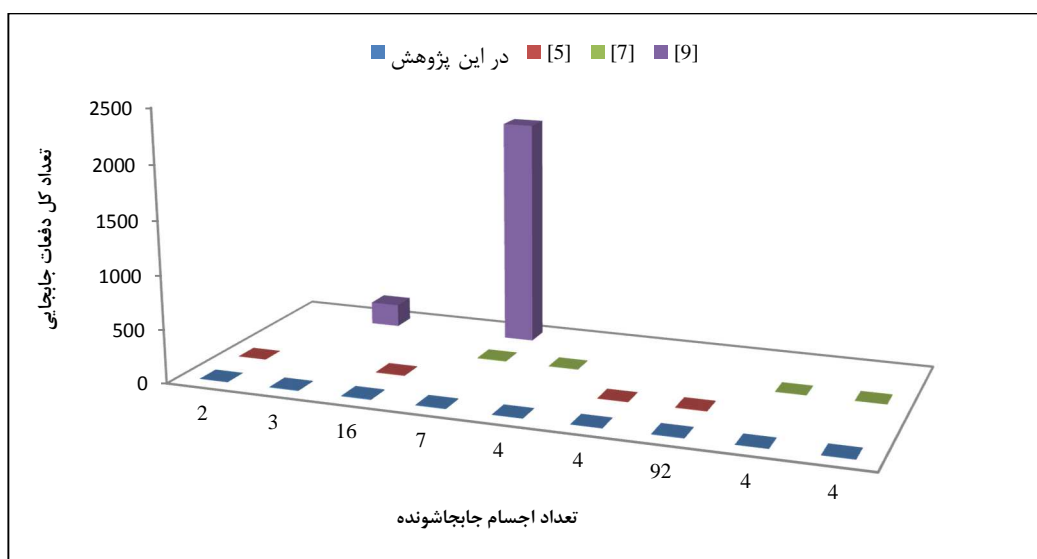
N/G : Not Given  
a: Our method (Intel Core i5 2.50GHz with 4 GB of memory)  
b: [5] Pentium 4, 3GHz  
c: [6] (Intel Pentium M, 1.6Ghz)  
d: [9] (1.66 GHz Intel T5500 processor with 1GByte of memory)  
e: [8] (Pentium 2.40GHz with 1GB of memory)



شکل (۶): مقایسه زمان حل مسائل موجود در ادبیات و زمان حل مسائل مشابه در این مقاله.



شکل (۷): مقایسه تعداد اجسام جابه‌جا شده در مسائل موجود در ادبیات و این مقاله.



شکل (۸): مقایسه تعداد کل دفعات جابه‌جایی اجسام در مسائل موجود در ادبیات و این مقاله.



هرچند مدت زمان احتساب شده برای حل یک مسئله در این مقاله نسبت به سایر تحقیقات موجود نمی‌تواند معیار مناسبی برای قیاس الگوریتم‌های ارائه شده باشد، ولی الگوریتم بازگشتی طراحی شده قادر است در زمان‌هایی معقول مسائل را حل کرده و حتی در برخی موارد، زمان حل آنها را بهبود بخشد (شکل ۶). همان‌طور که در شکل (۷) و (۸) نشان داده شده است، در این مقاله با استفاده از رویکرد انتخابی تعداد اجسام جابه‌جا شده و تعداد دفعات جابه‌جایی اجسام یا تغییر نکرده‌اند و یا نسبت به مسائل مشابه موجود کاهش یافته‌اند.

### ۵- نتیجه‌گیری و جمع‌بندی

در این مقاله به بررسی مسئله مسیریابی از میان اجسام جابه‌جاشونده (NAMO) پرداخته شد. همان‌گونه که از نام مسئله مشخص است، در NAMO دو وظیفه مسیریابی و جابه‌جایی اجسام بایستی توأمان انجام شوند. در حقیقت در این مسئله از روبات و یا هر وسیله دیگری که قابلیت جابه‌جایی اجسام را داراست، انتظار می‌رود تا با استفاده هم‌زمان از فنون مسیریابی و عمل‌وری اجسام، مسیری را از یک نقطه آغازین تا یک هدف مشخص طی کند. کاربرد بارز این مسئله در محیط‌های شلوغ و درهم‌ریخته مشخص می‌شود. نمونه بارز محیط‌های این‌چنینی را می‌توان پس از وقوع برخی وقایع طبیعی هم‌چون سیل، زلزله و طوفان مشاهده کرد.

فضای بسیار زیاد جستجو که ناشی از نامعلوم بودن مکان نهایی اجسام جابه‌جاشونده است، یکی از مشکلات اصلی حوزه NAMO محسوب می‌شود. به همین دلیل پژوهشگران، فضای حرکت اجسام جابه‌جاشونده را محدود می‌کنند. محدود کردن فضای حرکت روبات منجر به محدود کردن فضای جواب<sup>۲</sup> نیز خواهد شد.

در هیچ یک از پژوهش‌های مرور شده در این مقاله که تمرکز بیشتری روی آنها صورت گرفته است، مکان نهایی یک جسم جابه‌جاشونده به یک‌باره مشخص نمی‌شود؛ بلکه با استفاده از یک روش جستجو، جسم به صورت گام به گام در یک جهت خاص جابه‌جا شده و سپس برقرار بودن یک شرط

2- Solution Space

(برای مثال اتصال دو زیر فضای آزاد یا قابلیت قرار گرفتن روبات و یا جسم در یک سلول که قبلاً مسدود بوده است) بررسی می‌شود.

در این مقاله فضای کاری روبات به صورت پیوسته در نظر گرفته شده است و با استفاده از دو مفهوم گراف دیدنگار و عمق نفوذ تلاش شده که پیکربندی نهایی اجسام جابه‌جاشونده‌ای که مانع حرکت روبات هستند در همین فضای پیوسته و با صرف زمان کمتر مشخص گردد. استفاده از مفهوم عمق نفوذ می‌تواند یافتن موقعیت نهایی یک جسم جابه‌جاشونده را به مراتب ساده‌تر کند؛ چرا که با استفاده از این مفهوم نیازی به جابه‌جایی گام به گام جسم و بررسی یک شرط خاص وجود ندارد.

### ۶- منابع

- [۱] Wilfong, G., "Motion Planning In The Presence Of Movable Obstacles". Annals of Mathematics and Artificial Intelligence,, 1991. 3: p. 131-150
- [۲] Demaine, E.D., M.L. Demaine, and J. O'Rourke, "pushpush and push1 are hard in 2d". in In Proceedings of the 12th Canadian Conference on Computational Geometry,, 2000: p. 211-219
- [۳] Chen, P.C. and Y.K. Hwang, "Practical Path Planning among movable obstacles". Proceedings of the IEEE International Conference on Robotics and Automation, 1991: p. 444-449.
- [۴] Okada, K., et al., "Environment manipulation planner for humanoid robots using task graph that generates action sequence" Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.
- [۵] Stilman, M. and J.J. Kuffner, "Navigation Among Movable Obstacles: Real-Time Reasoning In Complex Environments". International Journal of Humanoid Robotics, 2005.2(4): p. 479-504.
- [۶] Stilman, M., et al., "Planning and Executing Navigation Among Movable Obstacles". in IEEE/RSJ Int. Conf. On Intelligent Robots and Systems (IROS 06), 2006: p. 820-826.

- [۷] Stilman, M. and J.J. Kuffner, “*Planning Among Movable Obstacles with Artificial Constraints*”. In Proc. 6th Int. Workshop on the Algorithmic Foundations of Robotics, 2006: p. 1-20.
- [۸] Nieuwenhuisen, D., A.F. van der Stappen, and M. H. Overmars, “*An Effective Framework for Path Planning amidst Movable Obstacles*”. Algorithmic Foundation of Robotics 2008.VII: p. 87-102.
- [۹] Berg, J.v.d., et al., “*Path Planning among Movable Obstacles: a Probabilistically Complete Approach*”. Workshop on Algorithmic Foundation of Robotics (WAFR VIII), 2008: p. 599-614.
- [۱۰] Wu, H.N., M. Levihn, and M. Stilman, “*Navigation Among Movable Obstacles in Unknown Environments*”. in IEEE/RSJ Int. Conf. On Intelligent Robots and Systems (IROS 2010).
- [۱۱] Levihn, M., “*Navigation among Movable Obstacles in Unknown Environments*”, 2011, Georgia Institute of Technology. p. 87
- [۱۲] Choset, H., et al., “*Principles of Robot Motion-Theory, Algorithms, and Implementation*”, 2005, Cambridge, Massachusetts: The MIT Press.
- [۱۳] Dobkin, D., et al., “*Computing the intersection-depth of polyhedra*”. Algorithmica, 1993.9: p. 518-533.
- [۱۴] Zhang, L., et al., “*Generalized penetration depth computation*”. Computer-Aided Design, 2007.39(8): p. 625–638.