

# مسیریابی سیستم چند رباتی در یک محیط سه بعدی با استفاده از الگوریتم بهینه‌سازی جمعیت ذرات بهبودیافته

آیت‌الله مهبودی<sup>۱</sup> حسین نظام‌آبادی پور<sup>۲</sup> محدثه سلیمانپورمقدم<sup>۳</sup>

۱- دانش آموخته کارشناسی ارشد- گروه مهندسی مکترونیک - مجتمع آموزش عالی بم - کرمان - ایران

[mehboodi@bam.ac.ir](mailto:mehboodi@bam.ac.ir)

۲- استاد- گروه مهندسی برق - دانشگاه شهید باهنر کرمان - کرمان - ایران

[nezam@uk.ac.ir](mailto:nezam@uk.ac.ir)

۳- استادیار- گروه مهندسی مکانیک - مجتمع آموزش عالی بم - کرمان - ایران

[m.soleimanpour@bam.ac.ir](mailto:m.soleimanpour@bam.ac.ir)

**چکیده:** یکی از بحث‌های اساسی و مهم در مورد ربات متحرک و توده ربات‌های متحرک مسئله برنامه‌ریزی مسیر است. مسئله برنامه‌ریزی، یافتن مسیری است که ربات را از نقطه شروع به نقطه هدف هدایت می‌کند. این حرکت باید به گونه‌ای باشد که ربات با موانع و دیگر ربات‌ها برخورد نداشته باشد، ضمن این که مسیر طی شده دو شرط ایمن بودن و بهینه بودن را نیز برآورده سازد و با کمترین هزینه به هدف خود برسد. تاکنون کارهای متعددی در زمینه مسیریابی بهینه توده رباتیکی در محیط‌های دو بعدی انجام شده است. اما در محیط واقعی، مسیر کاملاً مسطح نیست و محیط عملیاتی دارای ناهمواری‌هایی است که در حرکت ربات‌ها و یافتن مسیر بهینه بدون برخورد با موانع و دیگر ربات‌های گروه بسیار تأثیرگذار است. در این مقاله تلاش است تا به منظور دستیابی به عملکرد صحیح ربات، ناهمواری‌ها در محاسبات مسیریابی وارد شده و در محیط شبیه‌سازی در نظر گرفته شود. برای این منظور به کمک یک رویکرد ریاضی مسیر ناهموار طی شده به قطعات هموار شکسته و در محاسبات منظور می‌شود. در این مقاله، مسئله برنامه‌ریزی مسیر سه بعدی برای توده ربات‌ها به کمک روش فرا ابتکاری جمعیت ذرات بهبودیافته حل خواهد شد. روش پیشنهادی روی مسائلی با تعداد ربات‌های و تنوع محیطی متفاوت آزموده می‌شود. نتایج آزمایش، کارایی روش پیشنهادی را نسبت به الگوریتم جمعیت ذرات از دو جنبه طول مسیر و زمان سفر تأیید می‌کند.

**واژه‌های کلیدی:** توده رباتی، برنامه‌ریزی مسیر، الگوریتم بهینه‌سازی ذرات، محیط سه بعدی، موانع

**نوع مقاله:** پژوهشی

DOI: 10.52547/jiaeee.19.3.163

تاریخ ارسال مقاله: ۱۳۹۸/۱۲/۰۸

تاریخ پذیرش مشروط مقاله: ۱۴۰۰/۰۱/۲۱

تاریخ پذیرش مقاله: ۱۴۰۱/۰۲/۰۴

نام نویسنده‌ی مسئول: دکتر محدثه سلیمانپورمقدم

نشانی نویسنده‌ی مسئول: ایران - بم - مجتمع آموزش عالی بم - گروه مهندسی مکانیک

## ۱- مقدمه

صرفاً حسگر-مینا<sup>۵</sup> باشد و ربات بتواند با اتکا به حسگرهایش، مسیر مناسب تا رسیدن به هدف را (در صورت وجود) پیدا کرده و ببیند.

از طرف دیگر ناشناخته بودن محیط در دقت مکان‌یابی ربات نیز تأثیر مستقیم دارد. از این لحاظ که در محیط ناشناخته، ربات مشخصات مکان را از قبل نمی‌شناسد تا با مشاهده‌ی آن بتواند باور خود در مورد مکان فعلی‌اش را اصلاح کند. البته در صورتی که ربات توانائی مکان‌یابی و نقشه‌کشی هم‌زمان<sup>۶</sup> را داشته باشد، بعد از شناسایی و ترسیم نقشه‌ی محیط، مشخصات محیط را می‌داند [۳]. صرف‌نظر از نوع محیطی که ناوبری<sup>۷</sup> در آن انجام می‌شود، تمام روش‌های ناوبری باید توانائی اجتناب از موانع<sup>۸</sup> موجود در مسیر ربات به سمت هدف را داشته باشند. هر چه محیط بزرگ‌تر باشد پیچیدگی کار بیشتر بوده و پیچیدگی‌های طراحی مسیر با افزایش دامنه‌ی فضای وضعیت به صورت نمایی افزایش می‌یابد [۱].

در برنامه‌ریزی مسیر برای یک ربات، ربات باید ضمن حرکت به سمت هدف از برخورد با موانع متحرک و ثابت اجتناب کند که این کار ساده‌تر از طراحی برای توده ربات‌ها است. هر چه به تعداد ربات‌ها افزوده شود برنامه‌ریزی مسیر پیچیده‌تر می‌شود. اگر چه استفاده از سیستم‌های چندعاملی برای انجام کارهای مختلف رو به گسترش بوده، اما نحوه‌ی ایجاد هماهنگی و همکاری بین ربات‌ها در این سیستم‌ها، همواره از اهمیت ویژه‌ای برخوردار است. به بیان دیگر، مسیریابی برای یک سیستم چندعاملی مشکل‌تر از مسیریابی برای یک ربات است، چرا که با افزایش تعداد ربات‌ها، احتمال برخورد آن‌ها با یکدیگر در محیط بیشتر می‌شود که برای پیشگیری از این امر، هر ربات باید مسیر خودش و دیگر ربات‌ها را در نظر بگیرد.

از جهت دیگر، نوع ربات نیز در برنامه‌ریزی مسیر و محاسبات آن، تأثیر زیادی دارد و شکل، جرم، ابعاد، سرعت، نحوه‌ی حرکت و توانایی تغییر جهت، عواملی هستند که باید در محاسبات طراحی مسیر مورد توجه قرار گیرد. با توجه به آنچه ذکر شد، مسئله طراحی مسیر توده ربات‌ها درحالی‌که نوع و شکل ربات‌ها مشخص است در یک محیط که نوع محیط نیز معلوم است و با موانع مختلف و مقصد و هدف از پیش تعریف شده و با هدف مسیریابی ربات‌ها در حالی که از برخورد با یکدیگر و موانع اجتناب شود، بیان می‌شود. به منظور حل مسئله فوق باید از روش‌های برنامه‌ریزی مسیر استفاده کرد. روش‌های برنامه‌ریزی مسیر را می‌توان از زوایای مختلف مورد بررسی قرارداد که یکی از مهم‌ترین آن‌ها طراحی مسیر سراسری<sup>۹</sup> و طراحی مسیر محلی<sup>۱۰</sup> است. در روش طراحی مسیر سراسری که به طراحی حرکت ایستا<sup>۱۱</sup> نیز معروف است، قبل از شروع به حرکت کل مسیر، برنامه‌ریزی و مشخص می‌شود. این روش برای محیط‌های شناخته‌شده و محیط‌های ایستا، مناسب است. محیط ایستا و شناخته‌شده، حالت ایده‌آلی است که متأسفانه اتفاق نمی‌افتد و مسئله تحت تأثیر پارامترهای محیطی و نیز دیگر رخدادهای قرار می‌گیرد. طراحی مسیر محلی که حرکت پویا<sup>۱۲</sup> نیز نامیده می‌شود، اطلاعات مسیر حرکت ربات‌های متحرک همواره توسط

موضوع توده ربات‌ها، مبتنی بر هوش مصنوعی گروهی و همچنین مطالعات بیولوژیکی حشرات، مورچه‌ها و موجودات دیگر در طبیعت، که در آن‌ها رفتار گروهی وجود دارد، پدید آمده است. این قبیل موجودات هر کدام کار خود را انجام می‌دهند ولی رفتار پیچیده گروهی آن‌ها تابع رفتار همه آن‌ها است [۱]. گروه‌های جانوری دو رفتار متعادل شده دارند که یکی تمایل به ماندن در محدوده گروه و دیگری اجتناب از برخورد با موانع و دیگر اعضای گروه است و هرکدام از این دو خصوصیت فوایدی را برای آن‌ها در پی دارد. توده رباتیکی تأکید بر تعداد زیادی از ربات‌ها و ترویج مقیاس‌پذیری با استفاده از ارتباط محلی دارد. ربات‌های بسیار کوچک، قادر به استفاده در یک توده رباتی هستند که الهام گرفته از رفتارهای بروز داده شده در اجتماعات حشرات بوده که به هوش جمعی معروف است.

امروزه با ساخت ربات‌های مختلف، برنامه‌ریزی مسیر ربات، بسیار مورد توجه قرار گرفته است. به طور خلاصه، مسئله برنامه‌ریزی، یافتن مسیری است که ربات را از نقطه شروع به نقطه هدف هدایت می‌کند. در بسیاری از کاربردها برای اینکه ربات بتواند وظیفه خود را به درستی انجام دهد، نیاز به حرکت در محیط دارد. الزام حرکت در محیط موجب پرسیدن این سؤال می‌شود که ربات برای انجام وظیفه خود چه مسیری را طی کند که علاوه بر ایمن بودن، با کمترین هزینه به هدف خود برسد. با توجه به محدودیت انرژی، تسریع در انجام کارها و لزوم عدم برخورد ربات با موانع موجود در محیط، اهمیت این سؤال بیشتر درک می‌شود.

به منظور پاسخگویی به این پرسش که پرسش اساسی در زمینه تحقیقاتی مهمی تحت عنوان برنامه‌ریزی مسیر است، تاکنون پروژه‌های متعددی بر روی ناوبری خودمختار<sup>۱</sup> ربات‌های متحرک<sup>۲</sup> انجام شده است. برنامه‌ریزی مسیر به عواملی چون نوع محیط و موانع، نوع و تعداد ربات‌ها (تک ربات یا سیستم چند رباتی و توده ربات‌ها)، نحوه کنترل ربات‌ها (متمرکز یا گسترده) و چگونگی ارتباط آن‌ها و تبادل اطلاعات با همدیگر و دریافت اطلاعات از محیط بستگی دارد [۲] که در ادامه به اختصار تشریح خواهند شد.

با توجه به اطلاعات موجود از محیط، محیط‌های عملیاتی به دو نوع شناخته‌شده<sup>۳</sup> و ناشناخته<sup>۴</sup> تقسیم می‌شوند. در محیط شناخته‌شده، اطلاعات لازم از کل محیط در اختیار است. در صورتی که محیط شناخته‌شده باشد، ربات می‌تواند با توجه به نقشه‌ی محیط، مسیری بهینه و خالی از موانع را به سمت هدف محاسبه کند و با بهره‌گرفتن از یک روش مکان‌یابی مناسب، به هدف برسد. اما باید به این نکته اشاره داشت که در اختیار داشتن نقشه‌ی محیط همیشه امکان‌پذیر نیست. در محیط ناشناخته مسیر باید به گونه‌ای طراحی شود که ربات بدون نیاز به نقشه‌ی محیط و بدون آگاهی از مشخصات مکانی بتواند به نقطه‌ی هدف برسد. به عبارت دیگر، روشی مطلوب خواهد بود که

ریزی مبتنی بر حس گر در مرجع [۱۴] بحث شده‌اند. در مرجع [۱۵] روش‌های بر پایه اصول احتمالی برای حل مسئله برنامه‌ریزی مسیر بررسی شده است. مرجع [۹] را می‌توان به عنوان یکی از کتاب‌های مطرح در مسیریابی وسایل نقلیه به عنوان گروه ویژه‌ای از مسائل برنامه‌ریزی مسیر دانست.

همان‌گونه که در بخش قبلی اشاره شد، مسائل برنامه‌ریزی مسیر چند رباتی را می‌توان به روش‌های مختلفی مانند روش‌های برنامه‌ریزی به صورت لحظه‌ای یا غیر لحظه‌ای، موانع ایستا یا متحرک و برنامه‌ریزی به صورت متمرکز یا توزیع شده طبقه‌بندی کرد. در [۱۷، ۱۶] روش‌های متمرکز، با تابع هزینه‌های مختلف و محدودیت‌های محاسبه مسیر برای همه ربات‌ها با هم در نظر گرفته شده‌اند. در مقابل، در [۱۸] برنامه‌ریزی توزیع شده، با حرکت مستقل به سمت هدف و بدون برخورد با موانع استاتیک برای هر ربات مورد بررسی قرار گرفته است.

از منظر حل مسئله، الگوریتم‌های برنامه‌ریزی مسیر را می‌توان در دو گروه اصلی برنامه‌ریزی ریاضی و روش‌های فرا ابتکاری طبقه‌بندی کرد. روش‌های مختلف برنامه‌ریزی ریاضی مانند برنامه‌ریزی خطی<sup>۱۶</sup> [۱۹]، برنامه‌ریزی صحیح<sup>۱۷</sup> [۲۰]، برنامه‌ریزی پویا<sup>۱۸</sup> [۲۱] و برنامه‌ریزی غیرخطی<sup>۱۹</sup> [۲۲] برای حل مسائل برنامه‌ریزی مسیر استفاده شده است. این روش‌ها برای جستجوی فضای عملی مسئله در نزدیکی نقطه شروع اولیه از اطلاعات تابع هدف و مشتق آن استفاده می‌کنند. علاوه بر این، انتخاب یک نقطه شروع مناسب نقش به‌سزایی در عملکرد این گروه از الگوریتم‌ها دارد. از طرفی، مسائل برنامه‌ریزی مسیر به عنوان یک مسئله بهینه‌سازی غیرمحدب با روش‌های برنامه‌ریزی ریاضی با مشکلات و پیچیدگی‌های محاسباتی روبه‌رو می‌شود. از این رو الگوریتم‌های فرا ابتکاری برای حل مسئله بهینه‌سازی برنامه‌ریزی مسیر مورد توجه قرار گرفته‌اند.

روش‌های فرا ابتکاری مختلفی برای دستیابی به مسیر مطلوب در مسائل مسیریابی ربات‌ها مورد استفاده قرار گرفته‌اند. مرجع [۲۳] با ارائه الگوریتم بهینه‌سازی ذرات<sup>۲۰</sup> (PSO) با عملکرد جهش به حل مسئله مسیریابی پرداخته است. در [۲۴] الگوریتم PSO به منظور حل مسئله بهینه‌سازی چندهدفه استفاده شده است که برای یک مسئله برنامه‌ریزی مسیر چند رباتی با شرط جلوگیری از مانع در یک محیط پویا طراحی شده است. مرجع [۲۵] الگوریتم‌های مبتنی بر PSO و منفی آن (PSO-NPSO) را برای برنامه‌ریزی مسیر ربات پیشنهاد می‌کند. در مرجع [۲۶] از دو مدل برنامه‌ریزی مسیر چند هدفه برای یافتن یک مسیر ایمن با به حداقل رساندن مصرف انرژی استفاده شد. نویسندگان در [۲۷] از PSO باینری برای یافتن مسیر بهینه استفاده کردند. الگوریتم جستجوی گرانشی<sup>۲۱</sup> (GSA) و الگوریتم‌های PSO در مرجع [۲۸] برای چندین ربات متحرک چرخ‌دار هولونومیک اعمال می‌شوند. در تحقیق دیگری، الگوریتم بهینه‌سازی جمعیت ذرات بهبودیافته (IPSO) با الگوریتم جستجوی گرانشی بهبودیافته (IGSA)

حس‌گرهایی که بر روی ربات هستند، دریافت می‌شود و با توجه به این اطلاعات، مسیر ربات مشخص می‌شود. این روش طراحی حرکت، انعطاف‌پذیرتر و مطمئن‌تر از روش قبلی است و مناسب برای تغییر شکل و موقعیت موانع در محیط‌های دینامیک است. طراحی مسیر محلی برای محیط‌های ناشناخته استفاده می‌شود. [۴، ۲].

این مقاله در تلاش است تا مسئله طراحی مسیر محلی چند ربات در یک محیط دینامیک سه‌بعدی را مدل‌سازی کند. با توجه به آنچه ذکر شد، موانع ثابت هستند و ربات‌های دیگر در حکم موانع متغیر برای ربات هستند. ربات به صورت نقطه‌ای تعریف شده است و به هر طرف می‌تواند حرکت کند. ملاک بهینه‌سازی در این مسئله کوتاه‌ترین مسیر است و زمان سفر نیز بررسی می‌گردد. روش طراحی بکار گرفته شده، طراحی اولویت‌بندی شده از روش‌های برنامه‌ریزی مسیر مستقل است که کنترل ربات‌ها هم می‌تواند به صورت متمرکز باشد یعنی ربات‌ها اطلاعات را به کنترل‌کننده مرکزی داده و هماهنگی حرکت ربات‌ها توسط سیستم کنترل مرکزی انجام شود و هم می‌تواند به صورت توزیع شده انجام گیرد. طراحی اولویت‌بندی شده به این صورت که ربات‌ها شماره‌گذاری شده و اگر دو ربات با هم به یک نقطه برسند، ربات با شماره بیشتر متوقف شده تا ربات با شماره کمتر عبور کند.

در ادامه، در بخش دوم مروری بر کارهای انجام شده ارائه می‌شود، در بخش سوم الگوریتم بهینه‌سازی توده ذرات به عنوان اصول اولیه یک روش ابتکاری که مسئله برنامه‌ریزی مسیر را با آن حل کرده‌ایم، مورد بررسی قرار می‌گیرد. در بخش چهارم روش پیشنهادی را مطرح و در بخش پنجم چند مسئله نمونه را با آن روش حل می‌کنیم. در خاتمه در بخش ششم جمع‌بندی ارائه می‌شود.

## ۲- مروری بر کارهای گذشته

در یک مسئله برنامه‌ریزی مسیر سیستم‌های چند رباتی، هر ربات در یک فضای جستجوی دارای مکان اولیه و مقصد نهایی مشخص در تلاش است تا مسیر خود را بدون برخورد به هیچ‌یک از موانع موجود یا سایر ربات‌ها در محیط تعریف شده برنامه‌ریزی کند. ربات‌هایی مانند ربات‌های زیرآبی<sup>۱۳</sup> [۵]، ربات صخره‌نورد<sup>۱۴</sup> [۶]، ربات‌های هوایی کوچک<sup>۱۵</sup> [۷] و دیگر ربات‌های مشابه [۸] قبلاً با روش‌های مختلفی مورد آزمایش قرار گرفته‌اند [۹].

تحقیقات متفاوتی در زمینه برنامه‌ریزی مسئله مسیریابی وجود دارد. مرجع [۱۰] الگوریتمی برای حل مسئله یافتن کوتاه‌ترین مسیر ارائه کرده است. توضیحات مفصل بررسی جامعی از روش‌های پیشنهادی برای مسئله مذکور تا سال ۱۹۹۰ در مرجع [۱۱] مطرح شده است که یکی از بهترین منابع اطلاعاتی در مورد الگوریتم‌ها برای حل مشکلات برنامه‌ریزی مسیر با حضور موانع چندضلعی در فضای عملی مسئله است. مسائلی با محیط پویا در مرجع [۱۲] بررسی شده است. تعداد چشمگیری از مقالات مرتبط در زمینه مسیریابی ربات‌ها در مرجع [۱۳] بررسی و دسته‌بندی است. مقالاتی با موضوع برنامه-

در این روابط  $C_1$  و  $C_2$  مقادیر ثابت و مثبت و بطور تجربی ثابت شده است که مجموع آن‌ها برابر ۴ است.  $\emptyset_1$  و  $\emptyset_2$  اعداد تصادفی هستند که به صورت نرمال در بازه مورد نظر، تولید می‌شوند. باید به این نکته اشاره کرد که، جمعیت اولیه در الگوریتم (ذرات) به صورت تصادفی در سرتاسر فضای جستجو مقداردهی می‌شوند که این موقعیت‌های اولیه به عنوان بهترین تجربه شخصی ذرات نیز شناخته می‌شوند ( $Pbest_i$ ) و در گام بعد بهترین ذره از میان ذرات موجود انتخاب و به عنوان بهترین پاسخ شناخته می‌شود ( $Gbest$ ). علاوه بر آن، وزن اینرسی است که تاثیر سرعت ذرات در گام قبل را بر سرعت فعلی تعیین می‌نماید. به این ترتیب که با مقادیر بزرگی از وزن اینرسی، قابلیت جستجوی عمومی (کاوش) الگوریتم بهبود یافته و فضای بیشتری مورد بررسی قرار می‌گیرد. حال آن‌که با مقادیر کوچک وزن اینرسی، فضای مورد بررسی محدود شده و جستجو در فضای محدود شده (بهره‌گیری) صورت می‌گیرد. در الگوریتم‌های برپایه جمعیت تلاش است تا با ایجاد مصالحه بین کاوش و بهره‌گیری، جواب بهینه در زمان قابل قبولی بدست آید. به همین دلیل، معمولاً الگوریتم بهینه‌سازی ذرات با مقدار بزرگی از وزن اینرسی شروع به حرکت می‌کند که سبب جستجوی گسترده فضا و افزایش کاوش در ابتدای اجرای الگوریتم شده و این وزن به مرور در طول زمان کاهش می‌یابد که سبب تمرکز جستجو در فضای کوچک و افزایش بهره‌گیری در گام‌های پایانی می‌شود. در مرجع [۲] به منظور برقراری تعادل میان کاوش و بهره‌گیری در طول زمان برای محاسبه ضریب وزن اینرسی از رابطه زیر استفاده شده است.

$$w_i = w_{min} + (w_{max} - w_{min}) \left( \frac{dist_i}{max\_dist} \right) \quad (4)$$

در این رابطه کمترین و بیشترین مقدار وزن اینرسی به ترتیب ۰/۴ و ۰/۹ است.  $dist_i$  فاصله ذره  $i$ ام تا بهترین ذره و  $max\_dist$  فاصله دورترین ذره تا بهترین ذره است و از رابطه‌های زیر به دست می‌آیند.

$$dist_i = \|Gbest - X_i\|_2 \quad (5)$$

$$max\_dist = \max(dist_i) \quad (6)$$

همچنین ضرایب  $C_1$  و  $C_2$  نیز متغیر است و از روابط زیر محاسبه می‌شوند [۲].

$$C_1 = c_{1i} - \left( \frac{c_{1i} - c_{1f}}{max\_iter} \right) t \quad (7)$$

$$C_2 = c_{2i} + \left( \frac{c_{2f} - c_{2i}}{max\_iter} \right) t \quad (8)$$

در این روابط  $c_{1i}$ ،  $c_{1f}$ ،  $c_{2i}$  و  $c_{2f}$  به ترتیب مقادیر ابتدایی و انتهای  $C_1$  و  $C_2$ ،  $t$  تکرار فعلی و  $max\_iter$  بیشترین تکرار است. ادامه به معرفی روش پیشنهادی برای حل مسئله مسیریابی در فضای دوبعدی و سه‌بعدی می‌پردازیم.

#### ۴- روش پیشنهادی

روش پیشنهادی در واقع یک نسخه حریصانه از الگوریتم جمعیت ذرات است. در این روش، مانند الگوریتم بهینه‌سازی جمعیت ذرات، موقعیت

ترکیب شده (که با نام IPSO-IGSA نامیده می‌شود) و به حل مسئله در فضای دو بعدی پرداخته می‌شود [۲].

با بررسی آخرین تحقیقات انجام شده می‌توان دریافت که بیشتر مطالعات در محیط دو بعدی انجام شده‌اند، اما در مسایل عملی اغلب محیط دارای ناهمواری است که این رخداد نقش به‌سزایی در نحوه حل مساله و میزان پیچیدگی آن دارد.

در این مقاله، با توجه به اینکه کاربردهای واقعی توده ربات‌ها اغلب در فضاهای سه بعدی رخ می‌دهد با این مسئله روبرو می‌شویم که محیط واقعی مسیر کاملاً مسطحی نیست و محیط عملیاتی دارای ناهمواری‌هایی است که در حرکت ربات‌ها و یافتن مسیر بهینه بدون برخورد با موانع و دیگر ربات‌های گروه بسیار تاثیرگذار است. برای عملکرد صحیح ربات باید این ناهمواری‌ها را در محاسبات وارد کرد و در محیط شبیه‌سازی در نظر گرفته شود.

حل مسئله بصورت سه‌بعدی نیازمند شبیه‌سازی مسئله فضای سه‌بعدی و حل آن است. در مقاله، مسئله برنامه‌ریزی مسیر برای سیستم چند رباتی در یک فضای سه‌بعدی و به کمک الگوریتم بهینه‌سازی توده ذرات بهبود یافته به عنوان یکی از روش‌های ابتکاری حل خواهد شد. با توجه به نتایج به دست آمده الگوریتم بهینه‌سازی توده ذرات در مقایسه با روش قبلی کارایی بهتری از خود نشان داده است.

#### ۳- معرفی الگوریتم بهینه‌سازی ذرات

الگوریتم بهینه‌سازی ذرات یک الگوریتم جستجوی اجتماعی است که از روی رفتار اجتماعی دسته‌های پرندگان مدل شده است. مانند سایر الگوریتم‌های جمعیتی، الگوریتم PSO از مجموعه‌ای از پاسخ‌های ممکن استفاده می‌کند که این پاسخ‌ها تا زمانی که یک پاسخ بهینه یافت شود یا شرایط خاتمه الگوریتم محقق شود به حرکت خود ادامه می‌دهد. در شبیه‌سازی این الگوریتم، رفتار هر ذره تحت تاثیر بهترین اطلاعات فردی<sup>۲۲</sup> و بهترین ذره عمومی<sup>۲۳</sup> (بهترین ذره در میان ذرات) باشد.

$$Pbest_i = \{Pbest_i^1, Pbest_i^2, \dots, Pbest_i^p\} \quad (1)$$

در رابطه فوق  $Pbest_i^d$  نشان دهنده بعد  $d$ ام برای بهترین موقعیت دیده شده توسط ذره  $i$ ام است.

در الگوریتم بهینه‌سازی ذرات  $X_i(t)$  وضعیت فعلی است و  $X_i(t+1)$  وضعیت جدید (آینده) ذره است. با توجه به رابطه (۲) اگر به وضعیت فعلی مقدار سرعت جدید اضافه شود به وضعیت جدید (آینده) دست می‌یابیم. براساس الگوریتم بهینه‌سازی ذرات،  $V_i(t+1)$  به عنوان سرعت آینده از میزان تصادفی سه مولفه توسط رابطه (۳) بدست می‌آید [۲].

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

$$V_i(t+1) = w * V_i(t) + \quad (3)$$

$$C_1 * \emptyset_1 * (Pbest_i(t) - X_i(t)) +$$

$$C_2 * \emptyset_2 * (Gbest(t) - X_i(t))$$

برای جلوگیری از برخورد ربات‌ها به همدیگر از روش الویت‌بندی استفاده شده است. در صورتی که  $Gbest$  دو ربات، برابر باشند،  $Gbest$  ربات با شماره کمتر برورسانی می‌شود و  $Gbest$  ربات با شماره بالاتر در این مرحله برورسانی نمی‌شود و در واقع ربات در این مرحله سر جای خود می‌ماند و در تکرار دیگر حرکت می‌کند.

#### ۴-۱- برنامه ریزی مسیر در یک فضای سه بعدی

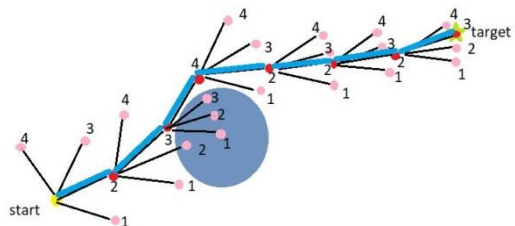
از جمله چالش‌ها و اموری که در مسیریابی ربات‌ها کمتر به آن توجه شده است، طراحی مسیر در فضای سه‌بعدی است. در هنگام استفاده از ربات‌ها با این مسئله روبرو می‌شویم که محیط واقعی مسیر، مسطح نبوده و محیط عملیاتی دارای ناهمواری‌هایی است که در حرکت ربات‌ها و یافتن مسیر بهینه بدون برخورد با موانع و دیگر ربات‌های گروه بسیار تاثیرگذار است. برای عملکرد صحیح ربات باید این ناهمواری‌ها در محاسبات وارد شده و در محیط شبیه‌سازی در نظر گرفت. به منظور بررسی دقیق‌تر موضوع ابتدا به بیان نحوه محاسبه طول منحنی می‌پردازیم.

برای محاسبه طول یک مسیر خمیده در یک صفحه (محیط دو بعدی)، مسیر را به تکه‌های کوچک تقسیم، دو سر هر تکه را به هم وصل کرد و طول پاره‌خط‌هایی را که دو سر آنها روی خم است را جمع می‌کنیم. این برآورد همواره حدی از دقت دارد و به تعداد پاره‌خط‌ها بستگی دارد. با کوچک لحاظ کردن پاره‌خط‌ها، خط شکسته‌ای که از پاره‌خط‌ها پدید می‌آید، هر چه بیشتر بر مسیر منحنی منطبق می‌شود. با انجام چنین کاری طول‌های این خط‌های شکسته به عددی میل می‌کنند که می‌توان آن را با یک انتگرال محاسبه کرد. در یک محیط سه‌بعدی برای بدست آوردن طول یک مسیر نیز می‌توان طول مسیر را به تکه‌های کوچکتری تقسیم کنیم (شکل ۲).

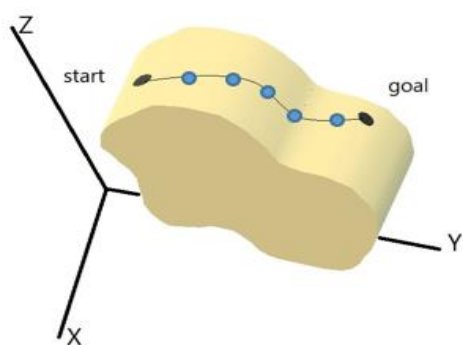
اولیه تمام ذرات در نقطه شروع تنظیم می‌شود و سرعت اولیه ذرات تصادفی انتخاب می‌گردد. این سرعت ذرات را به موقعیت جدید می‌رساند. پس از اعتبارسنجی ذرات و مشخص نمودن بهترین ذره، پس از برورسانی سرعت ذرات با استفاده از رابطه (۳)، موقعیت جدید ذرات از جمع سرعت هر ذره با موقعیت بهترین ذره از رابطه (۹) محاسبه می‌گردد و تا حصول شرط توقف الگوریتم تکرار می‌شود.

$$X_i(t+1) = Gbest(t) + V_i(t+1) \quad (9)$$

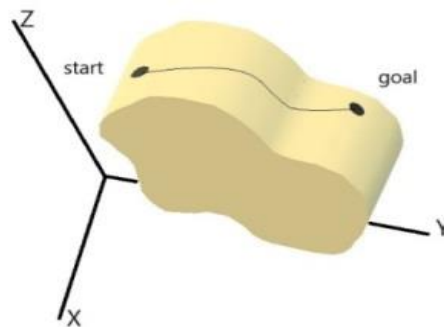
در این رابطه،  $X_i(t+1)$  موقعیت جدید ذره  $i$ ام،  $V_i(t+1)$  سرعت جدید ذره  $i$ ام و  $Gbest(t)$  موقعیت فعلی بهترین ذره در تکرار فعلی است. تفاوت اصلی روش پیشنهادی با الگوریتم اصلی بهینه‌سازی توده ذرات در برورسانی موقعیت ذرات است و دیگر مراحل و روابط مانند الگوریتم بهینه‌سازی توده ذرات است. چگونگی عملکرد الگوریتم در شکل (۱) نشان داده شده است. همان‌طور که در شکل مشخص است، هنگامی که در شروع حرکت از بین جمعیت ذرات، ذره دوم بهترین موقعیت را دارد، موقعیت ذره دوم مکان شروع برای حرکت تمام ذرات در تکرار بعد است و این حرکت تا رسیدن به شرط توقف ادامه می‌یابد. در هر تکرار، موقعیت بهترین ذره مکانی است که ربات به آنجا باید برود. در شکل (۱) موقعیت بهترین ذره در هر تکرار، با نقاط قرمز رنگ و مسیر ربات به رنگ آبی نشان داده شده است.



شکل (۱): مسیر حرکت ذرات، بهترین ذره و مسیر حرکت ربات



(ب) تقسیم مسیر به تکه‌های کوچک



(الف) مسیر بین دو نقطه

شکل (۲): محاسبه طول مسیر در یک مسئله سه بعدی

$$\Delta L = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2} \quad (10)$$

به بیان ساده، طول مسیر برابر با مجموع این تکه‌ها است و اندازه هر کدام از این تکه‌ها برابر است با:

در حل مسئله برنامه‌ریزی مسیر در یک محیط سه‌بعدی با استفاده از الگوریتم بهینه‌سازی ذرات، فرضیات ذیل لحاظ گردیده است. الف) محیط مسئله سه‌بعدی است. بدین معنی که زمین ناهموار است و ربات برای رسیدن به مقصد باید این مسیر ناهموار را طی کند. ب) ربات مسیر را زمینی طی می‌کند و قادر به پرواز کردن نیست. ج) محیط کاملاً شناخته‌شده است. بدین معنی که نقاط شروع و پایان و محل موانع کاملاً مشخص است و ضمن اینکه مختصات هر نقطه از صفحه را که انتخاب کنیم در دسترس است. د) موانع کروی و ثابت هستند و ربات‌های دیگر در حکم مانع متحرک برای ربات هستند.

برای اینکه ربات به نقطه هدف برسد، از نقطه مبدأ شروع به حرکت می‌کند. رویکرد حرکت به این صورت است که ربات مستقیم به سوی هدف حرکت کند به‌جز این‌که به یک مانع برخورد کند که در این صورت مانع را دور می‌زند و دوباره به مسیر خود ادامه می‌دهد. یک ربات با بردارهای سرعت مختلف و در می‌تواند حرکت کند و این بردارهای سرعت هستند که ربات را به مقصد جدید سوق می‌دهند و با توجه به بردارهای سرعت موقعیت‌ها و حالت‌های متفاوتی می‌تواند برای ربات به وجود آید که مجموعه این حالت‌ها، مجموعه جواب ممکن یا همان ذرات در الگوریتم هستند. نزدیک‌ترین ذره به هدف، در صورتی که در مرز مانع نباشد بهترین ذره است و به‌این ترتیب ربات از بین مجموعه جواب‌ها بهترین را انتخاب کرده و یک گام به هدف نزدیک می‌شود. بردارهای سرعت داری سه مؤلفه  $(x, y, z)$  هستند. انتخاب این مؤلفه‌ها به این صورت باید انجام می‌گیرد که بردار سرعت در جهات مختلف باید در رویه‌ای قرار گیرد که مسیر ربات از آن عبور می‌کند. اگر سطحی که ربات قرار است در آن حرکت کند با رابطه  $Z = f(x, y)$  بیان شود بردار سرعت، باید به صورت  $\vec{v} = (Vx, Vy, f(Vx, Vy))$  باشد تا در رویه قرار گیرد.

موقعیت اولیه همه ذرات در هر ربات نقطه شروع آن ربات است. نقطه شروع و هدف هر ربات در یک فضای سه‌بعدی به صورت  $start(m) = (Xs(m), Ys(m), Zs(m))$  و نقطه هدف هر ربات به صورت  $target(m) = (Xt(m), Yt(m), Zt(m))$  تعریف می‌شود. که با توجه به رابطه  $Z = f(x, y)$  این روابط به صورت زیر می‌توان نوشت.

$$start(m) = (Xs(m), Ys(m), f(Xs(m), Ys(m))) \quad (16)$$

$$target(m) = (Xt(m), Yt(m), f(Xt(m), Yt(m))) \quad (17)$$

سرعت اولیه یک مقدار تصادفی است که با رابطه (۱۸) بیان می‌شود.

$$Velocity_i(m) = (Vx_i(m), Vy_i(m), Vz_i(m)) \quad (18)$$

در این روابط  $m$  شماره ربات،  $i$  شماره ذره،  $Vx_i$  و  $Vy_i$  و  $Vz_i$  به ترتیب سرعت در راستای  $x$  و  $y$  و  $z$  و  $start$  و  $target$  به ترتیب نقاط شروع و هدف است. سرعت اولیه در دو مؤلفه  $x$  و  $y$  به صورت تصادفی در محدوده مجاز سرعت، تولید می‌شود و بعد از آن سرعت در راستای  $z$  با استفاده از رابطه (۱۹) به دست می‌آید.

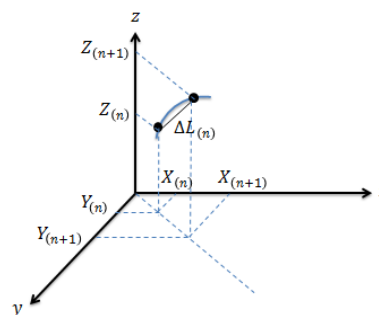
اگر مسیر ما به  $(N)$  قسمت، تقسیم شود یعنی نقطه شروع حرکت اولین نقطه (نقطه شماره ۱) باشد و نقطه هدف آخرین نقطه (نقطه شماره  $N + 1$ ) باشد، برای تعیین اندازه قسمت  $m$  از مسیر داریم:

$$\Delta X_n = (X_{n+1} - X_n) \quad (11)$$

$$\Delta Y_n = (Y_{n+1} - Y_n) \quad (12)$$

$$\Delta Z_n = (Z_{n+1} - Z_n) \quad (13)$$

$$\Delta l_n = \sqrt{\Delta x_n^2 + \Delta y_n^2 + \Delta z_n^2} \quad (14)$$

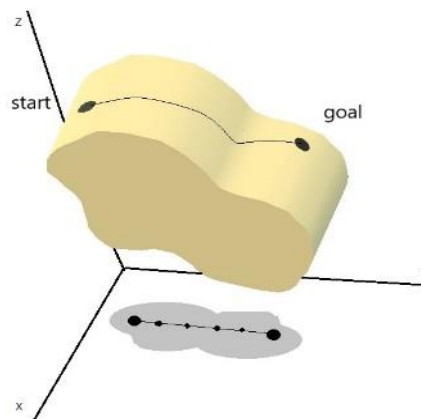


شکل (۳): تقریب یک منحنی با یک خط مستقیم

یعنی کافی است که ما در بین نقطه فعلی و هدف در یک توزیع یکنواخت تعداد  $(N + 1)$  نقطه انتخاب کنیم و فاصله بین این نقاط را به دست آوریم و سپس با جمع این فاصله‌ها، طول مسیر با تقریب خوبی به دست می‌آید. بنابراین طول مسیر ذره  $\Delta m$  تا مقصد برابر است با:

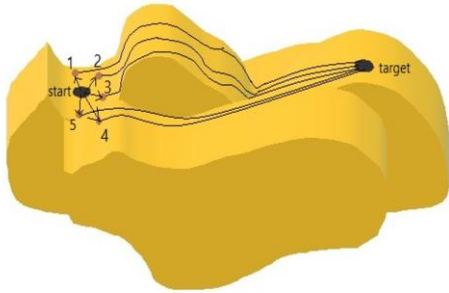
$$l_i = |\vec{v}_i| + \sum_{n=1}^N \sqrt{(X_{n+1} - X_n)^2 + (Y_{n+1} - Y_n)^2 + (Z_{n+1} - Z_n)^2} \quad (15)$$

به‌منظور انتخاب نقاط در طول مسیر، روش پیشنهادی این است که فاصله میان ذره تا مقصد بدون در نظر گرفتن ارتفاع (یک مسیر مستقیم در فضای دوبعدی (صفحه  $x - y$ )) را به  $N$  فاصله تقسیم شود و با در دست داشتن طول و عرض هر نقطه  $(x, y)$  هر نقطه، ارتفاع نقطه از رابطه  $Z = f(x, y)$  تعیین و سپس با استفاده از رابطه (۱۵) طول مسیر ذره تا مقصد محاسبه گردد (شکل ۴).



شکل (۴): روش تقسیم مسیر به چند قسمت

در شکل (۵) بردارهای سرعت  $v_1$  تا  $v_5$  می‌توانند ربات را در موقعیت‌های ۱ تا ۵ قرار دهند. بهترین نقطه که ربات می‌تواند به آن برود همان است که ربات را به هدف نزدیک‌تر سازد. با محاسبه طول مسیرها و مقایسه آن‌ها با یکدیگر، بهترین نقطه انتخاب می‌شود و ربات به آن نقطه می‌رود. اگر در شکل (۵) آن‌طور که از ظاهر شکل پیدا است مسیر ۴ کوتاه‌ترین مسیر باشد ربات به موقعیت ۴ می‌رود، سرعت‌ها به‌روزرسانی می‌شوند و از موقعیت جدید (نقطه ۴) مراحل قبل تکرار می‌شوند.



شکل (۵): انتخاب مسیر با کمترین طول توسط روش پیشنهادی (از بین موقعیت ذرات آنکه طول مسیر کمتری دارد به‌عنوان موقعیت بعدی ربات تعیین می‌گردد)

در بخش بعد به بررسی نتایج شبیه‌سازی در فضای دوبعدی و سه-بعدی می‌پردازیم.

## ۵- شبیه‌سازی و نتایج

شبیه‌سازی برنامه‌ریزی مسیر، این امکان را به ما می‌دهد که طریقه حرکت و رفتار ربات‌ها را بدون نیاز به پیاده‌سازی عملی مورد مطالعه قرار دهیم و میزان موفقیت و مؤثر بودن روش خود را بررسی کنیم که این امر باعث کاهش هزینه‌های اجرایی می‌گردد. در این فصل شبیه‌سازی مسیریابی چند ربات در یک فضای سه‌بعدی با استفاده از نرم‌افزار متلب انجام شده است. پارامترهای الگوریتم بهینه‌سازی توده ذرات پیشنهادی در جدول (۱) آمده است.

جدول (۱): پارامترهای الگوریتم پیشنهادی

$c_1 = 0.5$	تعداد تکرار = ۴۰
$c_2 = 3.5$	تعداد ذرات = ۵۰
$w = 1$	

### ۵-۱- شبیه‌سازی و نتایج در محیط دوبعدی

در ادامه شبیه‌سازی در محیط متلب انجام و روش پیشنهادی با الگوریتم بهینه‌سازی توده ذرات بهبودیافته برای یک ربات در فضای دوبعدی مقایسه شده است و در ادامه مقایسه‌های بیشتر با تعداد مختلف ربات ارائه شده است. در حالت اول، به حل مسئله مسیریابی برای یک ربات در حضور موانع پرداخته می‌شود. نقطه شروع ربات (۶-۸)، مکان همه ذرات در نقطه شروع است. سرعت اولیه ذرات

$$z_i(m) = f(Vx_i(m), Vy_i(m)) \quad (19)$$

سرعت باعث حرکت می‌شود و در مرحله اول هر ربات می‌تواند به موقعیت‌های مختلفی دست یابد که این مجموعه جواب، همان موقعیت ذرات PSO هستند:

$$X_i(m) = X_s(m) + Vx_i(m) \quad (20)$$

$$Y_i(m) = Y_s(m) + Vy_i(m) \quad (21)$$

$$Z_i(m) = Z_s(m) + Vz_i(m) \quad (22)$$

$$position_i(m) = (X_i(m), Y_i(m), Z_i(m)) \quad (23)$$

بنابراین موقعیت جدید ربات از بین مجموعه جواب‌های ممکن انتخاب می‌شود به این صورت که فاصله بین تمام موقعیت‌ها با مقصد محاسبه و نزدیک‌ترین نقطه به مقصد به‌عنوان موقعیت جدید انتخاب خواهد شد و ربات به موقعیت جدید حرکت کرده و این پایان مرحله اول است. پس از مرحله اول سرعت طبق روابط حاکم بر الگوریتم بهینه‌سازی توده ذرات بروز رسانی شده و مرحله دیگر الگوریتم شروع می‌شود و الگوریتم تا رسیدن ربات به هدف ادامه می‌یابد. برای محاسبه فاصله هر موقعیت تا هدف، فاصله را همان‌طور که شرح داده شد به تکه‌هایی تقسیم کرده و طول تکه‌ها را محاسبه و باهم جمع می‌کنیم.

برای جلوگیری از برخورد ربات با مانع یک تابع جریمه را برای ذراتی که به محدوده مؤثر مانع وارد می‌شوند تعریف و با مقدار  $l$  آن ذره جمع می‌شود. اگر مانع  $j$ ام به مرکز  $(x_o(j), y_o(j), z_o(j))$  باشد، فاصله مرکز مانع  $j$ ام تا موقعیت ذره  $i$ ام را با  $D_{ij}$  نشان می‌دهیم و از رابطه (۲۴) به دست می‌آوریم.

$$D_{ij}(m) = \sqrt{(x_o(j) - X_i(m))^2 + (y_o(j) - Y_i(m))^2 + (z_o(j) - Z_i(m))^2} \quad (24)$$

این مقدار باید از مقدار باید از شعاع مانع بیشتر باشد یعنی:  $D_{ij} > R_j$  یا  $D_{ij} - R_j > 0$ . مقدار جریمه و طول مسیر پس از اعمال جریمه نیز از رابطه (۲۵) بدست می‌آید [۲].

$$\begin{cases} Penalti = \max((R_j - D_{ij}), 0) \\ l = l + \lambda * Penalti \end{cases} \quad (25)$$

$R$  و  $j$  به ترتیب شماره و شعاع مانع،  $D$  فاصله مانع تا ذره و  $\lambda$  یک عدد مثبت است.

برای محاسبه طول مسیر، فاصله هر نقطه تا نقطه شروع نیز باید به طول مسیر اضافه شود. بنابراین طول مسیر ذره  $i$ ام برابر است با:

$$l_i = |\vec{v}_i| + \sum_{n=1}^N \sqrt{(X_{n+1} - X_n)^2 + (Y_{n+1} - Y_n)^2 + (Z_{n+1} - Z_n)^2} \quad (26)$$

در این رابطه  $v_i$  سرعت ذره  $i$ ام است. طول مسیر ذره  $i$ ام تا مقصد به  $N$  قسمت تقسیم شده است که  $n$  شماره هر قسمت است. همان‌طور که قبلاً هم اشاره شد هر چه تعداد این تکه‌ها بیشتر و تراکم نقطه‌های انتخابی بیشتر باشد تقریب به مقدار واقعی نزدیک‌تر می‌شود.

ذرات اعتبار سنجی شده و ربات به موقعیت بهترین ذره حرکت می‌کند و تا رسیدن ربات به مقصد الگوریتم تکرار می‌گردد. جدول (۲) نتایج حاصل از اجرای ده مرتبه اجرای الگوریتم بهینه‌سازی توده ذرات و روش پیشنهادی است. همان‌طور که مشاهده می‌شود طول مسیر در روش پیشنهادی کمتر و سرعت رسیدن ربات به مقصد هم بیشتر از الگوریتم بهینه‌سازی توده ذرات است. تعداد شکل حرکت مسیر حرکت ربات به دو روش الگوریتم بهینه‌سازی توده ذرات و الگوریتم پیشنهادی را در یکی از اجراها نشان می‌دهد.

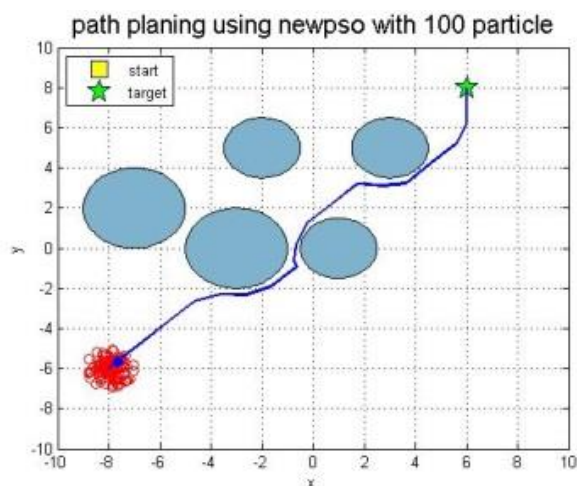
جدول (۲): میانگین طول مسیر و تعداد تکرار برای دستیابی به هدف یک ربات به روش پیشنهادی و روش PSO در یک محیط دوبعدی

روش پیشنهادی	PSO	
۲۲/۳۳	۳۰/۰۸	میانگین طول مسیر طی شده توسط ربات
۲۴/۳	۳۵/۱	میانگین تعداد تکرار برای رسیدن به هدف

باید به این مهم اشاره داشت که هر چه تعداد ذرات بیشتر باشد مسیر با طول کمتری پیدا خواهد شد اما با افزایش تعداد ذرات زمان اجرای الگوریتم افزایش می‌یابد. جدول (۳) مقایسه اجرای الگوریتم با تعداد ده ذره و صد ذره را نشان می‌دهد.

جدول (۳): تأثیر تعداد ذرات در طول مسیر

اجرای الگوریتم با تعداد ۱۰ ذره	اجرای الگوریتم با تعداد ۱۰۰ ذره	
۲۸/۷۴	۲۲/۳۳	میانگین طول مسیر در ۱۰ بار اجرای الگوریتم
۱/۰۲ ثانیه	۴/۹۲ ثانیه	متوسط زمان اجرا در ۱۰ بار اجرای الگوریتم (ثانیه)



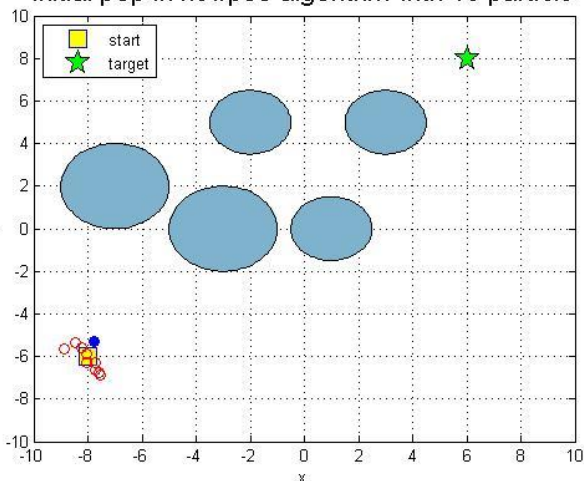
ب) اجرای الگوریتم پیشنهادی با ۱۰۰ عضو جمعیت

شکل (۸): مسیر حرکت ربات با تعداد مختلفی از اعضای جمعیت در الگوریتم پیشنهادی

در حالت بعدی به‌منظور بررسی دقیق‌تر روش پیشنهادی به مسیریابی چند ربات می‌پردازیم. با اضافه نمودن هر نقطه شروع و هدف به برنامه، یک ربات به شکل اضافه می‌گردد. اجرای برنامه برای ۵ ربات به روش پیشنهادی و الگوریتم ازدحام ذرات جدول (۴) را نتیجه می‌دهد.

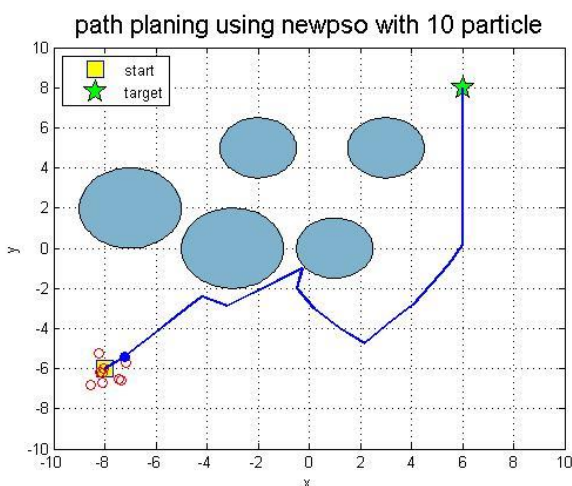
به‌طور تصادفی انتخاب می‌گردد. این سرعت ذرات را به موقعیت جدید می‌رساند. مسیر ذرات از نقطه شروع تا موقعیت جدید تا نقطه مقصد محاسبه شده و ذره دارای کمترین مسیر به‌عنوان بهترین ذره انتخاب شده و ربات به موقعیت بهترین ذره حرکت می‌کند. شکل (۶) موقعیت ده ذره را پس از اولین حرکت نشان می‌دهد. موقعیت بهترین ذره با رنگ آبی و بقیه ذرات به‌صورت دایره‌های توخالی قرمز نشان داده شده است.

initial pop in newpso algorithm with 10 particle



شکل (۶): تولید تصادفی موقعیت ده ذره بعد از اولین حرکت در الگوریتم پیشنهادی و موقعیت بهترین ذره که با رنگ آبی از دیگر ذرات متمایز است.

در تکرار بعد سرعت ذرات به‌روزرسانی شده و با موقعیت بهترین ذره جمع شده تا موقعیت جدید ذرات به دست آید. موقعیت جدید



الف) اجرای الگوریتم پیشنهادی با ۱۰ عضو جمعیت

در شکل (۷) مسیر حرکت ربات از نقطه شروع تا هدف و از بین موانع دیده می‌شود. همان‌گونه که در شکل می‌بینیم با افزایش جمعیت مسیر انتخابی بهتر و با طول کمتری خواهد بود. در شکل (۸) دایره‌های توخالی قرمز رنگ موقعیت ذرات در اولین حرکت و دایره تو پر آبی رنگ موقعیت بهترین ذره در اولین حرکت است.

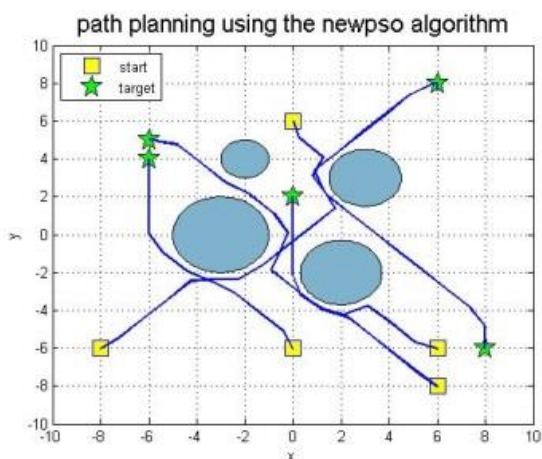


مسیریابی ۵ ربات

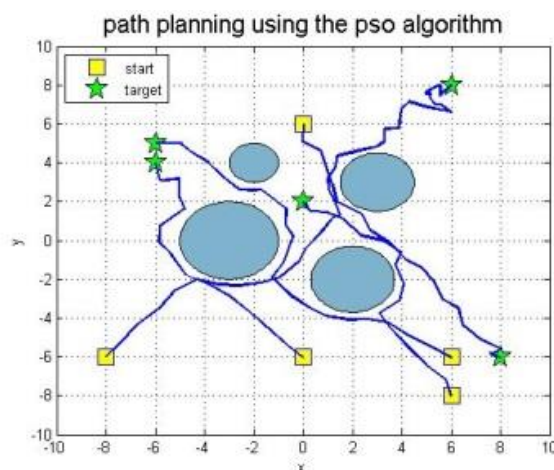
شماره ربات	میانگین تعداد تکرار برای رسیدن به هدف		میانگین مسیر طی شده توسط ربات	
	PSO	روش پیشنهادی	PSO	روش پیشنهادی
۱	۳۸/۴۱	۱۹/۷۳	۱۳/۴۶	۱۲/۵۹
۲	۳۸/۸۲	۱۹/۹۸	۱۳/۸۲	۱۲/۹۴
۳	۴۰/۸۴	۲۲/۹۲	۲۵/۴۱	۲۲/۳۵
۴	۳۹/۴۴	۲۰/۰۲	۱۶/۷۱	۱۵/۵۸
۵	۳۹/۹۲	۲۱/۳۱	۲۰/۶۱	۱۹/۴۶

همان‌طور که در جدول (۴) قابل مشاهده است میانگین طول مسیر در ده اجرا و مجموع طول مسیر برای همه ربات‌ها در روش پیشنهادی کمتر از PSO است. مهم‌تر این‌که در روش پیشنهادی ربات‌ها در تکرار کمتر الگوریتم به مقصد می‌رسند و تعداد تکرارها در روش پیشنهادی تقریباً نصف روش بهینه‌سازی توده ذرات است. تعداد تکرارها به سرعت و طول مسیر بستگی دارد. تعداد موانع که در مسیر ربات قرار می‌گردند و اندازه آن‌ها، فاصله بین شروع و مقصد و تعداد ربات‌ها باعث افزایش طول مسیر می‌شوند. شکل (۹) مسیر حرکت ربات‌ها را نشان می‌دهد.

جدول (۴): بررسی میانگین طول مسیر هر ربات و تعداد تکرار برای



ب) الگوریتم پیشنهادی



الف) الگوریتم PSO

شکل (۹): مقایسه مسیر حرکت ۵ ربات برای الگوریتم PSO و الگوریتم پیشنهادی با نقاط شروع و پایان مختلف

جدول (۵): بررسی میانگین طول مسیر هر ربات و تعداد تکرار برای

مسیریابی ۷ ربات

شماره ربات	میانگین تعداد تکرار برای رسیدن به هدف		میانگین مسیر طی شده توسط ربات	
	PSO استاندارد	روش پیشنهادی	PSO استاندارد	روش پیشنهادی
۱	۲۶/۱۸	۱۷/۲۶	۱۸/۸۶	۱۸/۳۹
۲	۲۶/۱۳	۱۷/۱۲	۱۱/۷۱	۱۱/۳۱
۳	۲۵/۷۶	۱۶/۵۲	۷/۴۸	۷/۳۷
۴	۲۶/۰۴	۱۷/۰۱	۱۲/۶۹	۱۲/۲۳
۵	۲۶/۶۲	۱۷/۲۳	۱۶/۳۲	۱۵/۳۵
۶	۲۶/۱۰	۱۶/۸۹	۱۸/۵۶	۱۷/۶۸
۷	۲۶/۸۷	۱۷/۹۴	۱۲/۷۸	۲۴/۵۳

همان‌طور که مشاهده می‌گردد طول مسیر در الگوریتم پیشنهادی کوتاه‌تر است. علاوه بر آن روش پیشنهادی خیلی زودتر ربات را به مقصد می‌رساند. شکل (۱۰) مسیر حرکت ربات‌ها را در دو الگوریتم مورد مقایسه به تصویر کشیده است.

با توجه به شکل (۱۰) می‌توان دریافت که الگوریتم پیشنهادی به مسیری هموارتر و با طول کمتر برای مسئله مسیریابی دست می‌یابد که این نشان دهنده عملکرد مناسب الگوریتم پیشنهادی در حل مسئله سه‌بعدی است.

۵-۲- محیط سه‌بعدی

در این قسمت دو مسئله با شرایط محیطی مختلف و با تعداد متفاوت ربات‌ها تعریف و سپس به حل آن‌ها خواهیم پرداخت. مکان و اندازه موانع، نقاط شروع و هدف ربات و توزیع آن‌ها در محیط، بیشتر تصادفی و در محدود مواردی به‌منظور ایجاد شرایط بعضی از آزمایش‌های مشخص (مثلاً در دو طرف یک چاله) انتخاب شده است. در ایجاد روابط محیط سعی بر آن شده است که محیطی با شکل ناهمواری‌های مختلف ایجاد شود. به منظور بررسی عملکرد الگوریتم در مسئله مسیریابی در محیط سه‌بعدی، به مقایسه الگوریتم پیشنهادی و الگوریتم بهینه‌سازی جمعیت ذرات استاندارد می‌پردازیم.

محیط مسئله اول با رابطه  $z = f(x, y) = (0.25 * x) - (\frac{3}{1+(y^2)})$  تعریف شده است. تعداد ۱۲ مانع کروی در محیط قرار دارند. این محیط یک محیط نسبتاً هموار است که یک دره سراسری و همان‌طور که از معادله پیدا است مرکز دره در  $y = 0$  است. چون دره سراسری است بنابراین ربات ناچار به عبور از آن است. مسیریابی برای ۷ ربات در این محیط به دو روش بهینه‌سازی توده ذرات و روش پیشنهادی انجام شده است. جدول (۵) مقایسه میانگین طول مسیر و تعداد تکرار در محیط سه‌بعدی اول را نشان می‌دهد.

است. مقایسه حرکت ربات‌ها در دو الگوریتم حکایت از حرکت یکنواخت و کم‌نوسان الگوریتم پیشنهادی دارد.

### ۶- نتیجه‌گیری

استفاده از سیستم‌های چند رباتی به علت توانایی‌های مانند مقیاس‌پذیری، انعطاف‌پذیری و مقاومت، در حال افزایش است. یکی از بحث‌های اساسی و مهم در مورد ربات متحرک و سیستم‌های چند رباتی مسئله برنامه‌ریزی مسیر است. به‌طور خلاصه، مسئله برنامه‌ریزی، یافتن مسیری است که ربات را از نقطه شروع به نقطه هدف هدایت می‌کند. این حرکت باید به‌گونه‌ای باشد که ربات با موانع و دیگر ربات‌ها برخورد نداشته باشد. ضمن اینکه مسیر طی شده دو شرط ایمن بودن و بهینه بودن را نیز برآورده سازد و با کم‌ترین هزینه به هدف خود برسد. در این مقاله از الگوریتم بهینه‌سازی ذرات بهبودیافته برای مسیریابی یک سیستم چند رباتی به روش محلی در فضای دو-بعدی و سه‌بعدی استفاده شد. در این تحقیق، روش محاسبه طول مسیر ذرات تا مقصد، تقسیم مسیر به تکه‌های کوچک‌تر بود که بسیار ساده و کارا و بدون هیچ پیچیدگی ریاضی بود. لزوماً با افزایش تعداد تکه‌ها، طول تقریبی به اندازه واقعی نزدیک‌تر خواهد شد. در نهایت شبیه‌سازی مسئله مسیریابی برای محیط‌های دوبعدی و همچنین محیط‌های ناهموار با تعداد موانع متفاوت و برای تعداد مختلفی از ربات‌ها مورد آزمایش قرار گرفته است. براساس آزمایش‌های انجام‌شده، روش پیشنهادی توانایی قابل قبولی در مسیریابی ربات‌ها در محیط‌هایی با ناهمواری‌های مختلف و با تعداد ربات و موانع متفاوت را دارا است.

رابطه محیط دو به صورت  $z = f(x, y) = \sqrt{|x - y(\frac{2}{3})|} + \frac{10}{(1+(y+2)^2+(x-4)^2)}$  تعریف شده است که تعداد ۷ مانع کروی در محیط قرار دارند. این محیط یک رویه مقعر است که یک تپه در آن قرار دارد. مسئله دوم، مسیریابی برای ۶ ربات در این محیط تعریف شده است. نقاط شروع و هدف بعضی از ربات‌ها در دو طرف تپه انتخاب شده‌اند که رفتار ربات در برخورد با تپه را مشاهده کنیم. اطلاعات طول مسیر و تعداد تکرار در جدول (۶) آمده است. همان‌طور که مشاهده می‌شود، روش پیشنهادی مسیریابی را سریع‌تر و با طول مسیر کوتاه‌تر پیدا کرده است.

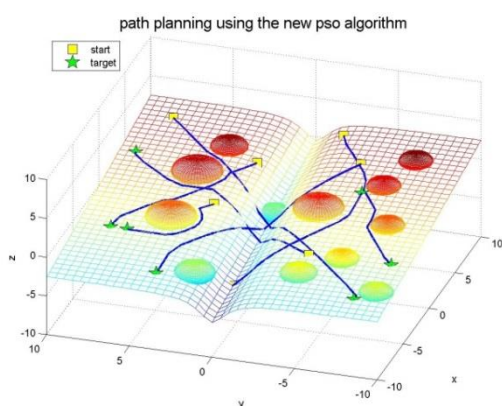
جدول (۶): بررسی میانگین طول مسیر هر ربات و تعداد تکرار برای

#### مسیریابی ۶ ربات برای محیط دوم

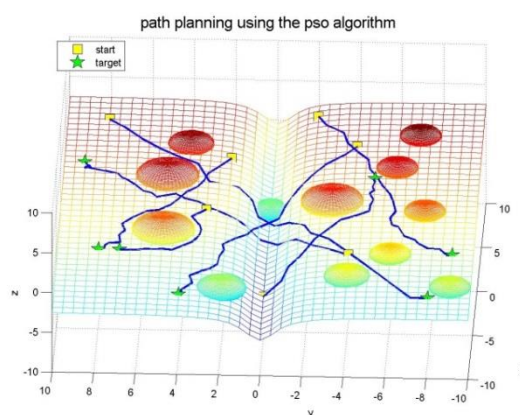
شماره ربات	میانگین تعداد تکرار برای رسیدن به هدف		میانگین مسیر طی شده توسط ربات	
	PSO استاندارد	روش پیشنهادی	PSO استاندارد	روش پیشنهادی
۱	۳۱/۱۳	۱۹/۷۳	۲۴/۱۴	۲۱/۱۴
۲	۳۰/۹۶	۱۸/۹۱	۱۲/۲۹	۱۲/۹۱
۳	۳۱/۷۹	۱۹/۲۵	۱۹/۳۶	۱۸/۰۸
۴	۳۱/۶۴	۱۹/۰۲	۱۷/۷۱	۱۶/۲۴
۵	۳۱/۹۲	۱۹/۵۱	۲۰/۲۱	۱۸/۹۶
۶	۳۲/۱۷	۱۹/۹۲	۱۹/۴۸	۱۸/۶۵

شکل (۱۱) مسیر حرکت ربات‌ها در دو الگوریتم را نشان داده است. مشابه آنچه در محیط اول رخ داد، الگوریتم پیشنهادی قابلیت دستیابی به مسیری هموار را دارا است.

روش حل مسئله در یک فضای سه‌بعدی به‌گونه‌ای است که ربات‌ها از همان آغاز حرکت سعی در مواجهه نشدن با تپه دارند که با توجه به ارتفاع تپه، این رفتار ربات‌ها برای کاهش طول مسیر، منطقی

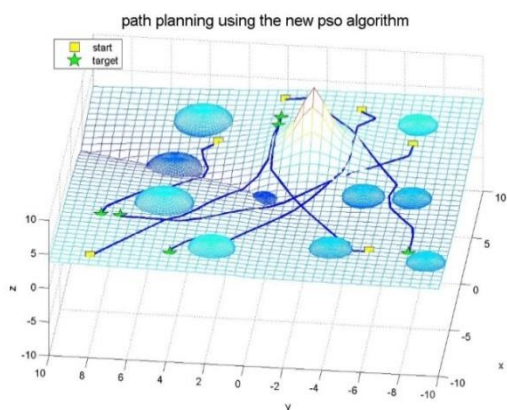


ب) الگوریتم پیشنهادی

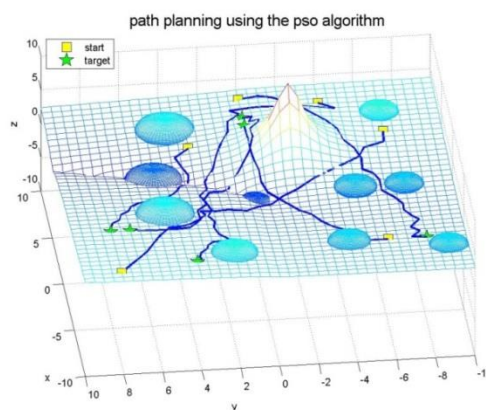


الف) الگوریتم جمعیت ذرات استاندارد

شکل (۱۰): مقایسه مسیر حرکت ۷ ربات برای الگوریتم پیشنهادی و الگوریتم جمعیت ذرات استاندارد در محیط سه‌بعدی اول



ب) الگوریتم پیشنهادی



الف) الگوریتم جمعیت ذرات استاندارد

شکل(۱۱): مقایسه مسیر حرکت ۶ ربات برای الگوریتم پیشنهادی و الگوریتم جمعیت ذرات استاندارد در محیط سه بعدی دوم

## مراجع

- [1] راضیه فرازکیش. ۱۳۹۹. مدل سازی قابلیت اطمینان در نانوربات های زیستی، نشریه مهندسی برق و الکترونیک ایران. 3. 11-16.
- [2] Das, P.K., Behera, H.S. and Panigrahi, B.K., 2016. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation*, 28, pp.14-28
- [3] Tang, B., Zhu, Z. and Luo, J., 2016. Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning. *International Journal of Advanced Robotic Systems*, 13(3), p.86.
- [4] Nguyen, H.T. and Le, H.X., 2016. Path planning and Obstacle avoidance approaches for Mobile robot. *arXiv preprint arXiv:1609.01935*.
- [5] Ahmadi-Mousavi, M., Moshiri, B., & Heshmati, Z. , 2018. Linear Time Varying MPC Based Path Planning of an Autonomous Vehicle via Convex Optimization. *Journal of Iranian Association of Electrical and Electronics Engineers*, 14(4), pp.79-88.
- [6] Yue, R., Xiao, J., Wang, S. and Joseph, S.L., 2009, December. Modeling and path planning of the city-climber robot part II: 3D path planning using mixed integer linear programming. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 2391-2396). IEEE.
- [7] Duan, H., Yu, Y., Zhang, X. and Shao, S., 2010. Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm. *Simulation Modelling Practice and Theory*, 18(8), pp.1104-1115
- [8] Schler, F., 2012. 3d path planning for autonomous aerial vehicles in constrained spaces [Ph. D. thesis]. Department of Electronic Systems, Faculty of Engineering and Science, Aalborg University, Aalborg, Denmark.
- [9] Das, P. K., & Jena, P. K., 2020, Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Applied Soft Computing*, 92, 106312.
- [10] Carlyle, W.M., Royset, J.O. and Kevin Wood, R., 2008. Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks: An International Journal*, 52(4), pp.256-270.
- [11] Laumond, J.P. ed., 1998. *Robot motion planning and control* (Vol. 229). Berlin: Springer.
- [12] Fiorini, P. and Shiller, Z., 1998. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7), pp.760-772.
- [13] Hwang, Y.K. and Ahuja, N., 1992. Gross motion planning—a survey. *ACM Computing Surveys (CSUR)*, 24(3), pp.219-291.
- [14] Tarabanis, K.A., Allen, P.K. and Tsai, R.Y., 1995. A survey of sensor planning in computer vision. *IEEE transactions on Robotics and Automation*, 11(1), pp.86-104.
- [15] Svestka, P. and Overmars, M.H., 1995, May. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation (Vol. 2, pp. 1631-1636)*. IEEE.
- [16] Nazarahari, M., Khanmirza, E., & Doostie, S., 2019. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, 115, pp. 106-120.
- [17] Zhang, H. Y., Lin, W. M., & Chen, A. X., 2018, Path planning for the mobile robot: A review. *Symmetry*, 10(10), 450.
- [18] Guo, Y. and Parker, L.E., 2002, May. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292) (Vol. 3, pp. 2612-2619)*. IEEE.
- [19] Banfi, J., Guzzi, J., Amigoni, F., Flushing, E.F., Giusti, A., Gambardella, L. and Di Caro, G.A., 2019. An integer linear programming model for fair multitarget tracking in cooperative multirobot systems. *Autonomous Robots*, 43(3), pp.665-680.
- [20] Yeh, J., Lewis, L.I., Alvarez, J. and Cannons, J., 2017. Robot Path Planning using Integer Linear Programming.
- [21] Van Den Berg, J., Patil, S. and Alterovitz, R., 2017. Motion planning under uncertainty using differential dynamic programming in belief space. In *Robotics Research* (pp. 473-490). Springer, Cham.
- [22] Pardo, D., Möller, L., Neunert, M., Winkler, A.W. and Buchli, J., 2016. Evaluating direct transcription and nonlinear optimization methods for robot motion planning. *IEEE Robotics and Automation Letters*, 1(2), pp.946-953.
- [23] Qin, Y.Q., Sun, D.B., Li, N. and Cen, Y.G., 2004, August. Path planning for mobile robot using the particle swarm optimization with mutation operator. In *Proceedings of 2004 international conference on*

- machine learning and cybernetics (IEEE Cat. No. 04EX826) (Vol. 4, pp. 2473-2478). IEEE.
- [24] Min, H.Q., Zhu, J.H. and Zheng, X.J., 2005, August. Obstacle avoidance with multi-objective optimization by PSO in dynamic environment. In 2005 International Conference on Machine Learning and Cybernetics (Vol. 5, pp. 2950-2956). IEEE.
- [25] Masehian, E. and Sedighzadeh, D., 2010. Multi-objective PSO-and NPSO-based algorithms for robot path planning. *Advances in electrical and computer engineering*, 10(4), pp.69-76.
- [26] Davoodi, M., Panahi, F., Mohades, A. and Hashemi, S.N., 2015. Clear and smooth path planning. *Applied Soft Computing*, 32, pp.568-579.
- [27] Qiaorong, Z. and Guochang, G., 2008, September. Path planning based on improved binary particle swarm optimization algorithm. In 2008 IEEE Conference on Robotics, Automation and Mechatronics (pp. 462-466). IEEE.
- [28] Purcaru, C., Precup, R.E., Iercan, D., Fedorovici, L.O., Petriu, E.M. and Voisan, E.I., 2013, July. Multi-robot GSA-and PSO-based optimal path planning in static environments. In 9th International Workshop on Robot Motion and Control (pp. 197-202). IEEE.

## زیر نویس ها

- <sup>1</sup> Autonomous navigation
- <sup>2</sup> Mobile robots
- <sup>3</sup> Known
- <sup>4</sup> Unknown
- <sup>5</sup> Sensor- based
- <sup>6</sup> Simultaneous Localization And Mapping(SLAM)
- <sup>7</sup> Navigation
- <sup>8</sup> Avoidance obstacles
- <sup>9</sup> Global path planning(GPP)
- <sup>10</sup> Local path planning(LPP)
- <sup>11</sup> Of line
- <sup>12</sup> Dynamic
- <sup>13</sup> Underwater robot
- <sup>14</sup> Wall-climbing robot
- <sup>15</sup> Micro air robot
- <sup>16</sup> Linear programming
- <sup>17</sup> Integer programming
- <sup>18</sup> Dynamic programming
- <sup>19</sup> Nonlinear programming
- <sup>20</sup> Particle Swarm Optimization (PSO)
- <sup>21</sup> Gravitational Search Algorithm (GSA)
- <sup>22</sup> Local Best
- <sup>23</sup> Global Best