

Optimal Path Planning for Mobile Robots Using a Multi-Objective Framework: Integrating Particle Swarm Optimization and Adaptive Bat Algorithm

Mohammad Asadi¹, Zahra Mohammadi², Hossein Nezamabadi-pour², Mohadeseh Soleimanpour-Moghadam^{1*}

¹Department of Mechatronics Engineering, Bam Higher Education Complex, Bam, Iran

²Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

Received: 12 July 2025, Revised: 16 March 2026, Accepted: 28 April 2026

Paper type: Research

Abstract

This study proposes a novel path planning framework based on multi-objective optimization, which simultaneously addresses three key criteria: minimizing path length, optimizing motion smoothness, and maximizing safe distance from obstacles. The multi-objective problem is first transformed into a single-objective formulation using the ϵ -constraint method and weighted sum approach. Given the NP-hard nature of the problem, an adaptive hybrid metaheuristic algorithm, termed PSO-ABA (Particle Swarm Optimization-Adaptive Bat Algorithm), is developed by integrating the global search mechanism of PSO with the frequency-tuning capabilities of the Bat Algorithm (BA). Subsequently, an adaptive hybrid metaheuristic algorithm, termed PSO-ABA, is developed, which leverages the integration of two complementary global search mechanisms based on Particle Swarm Optimization (PSO) and the Adaptive Bat Algorithm (ABA). In this algorithm, the task of updating the robot's position is assigned to the ABA, while the PSO is focused on tuning the appropriate parameters. The proposed algorithm addresses several key aspects. First, it aims to develop a trajectory in real space by modifying the existing relationships within the Bat Algorithm. This modification ensures that the robot's movement is directed as closely as possible toward the target path while minimizing abrupt turns. Second, by enhancing the fitness function, the algorithm prevents the robot's position from being updated when it is within obstacles, effectively eliminating invalid paths. Performance evaluation in simulated environments with complex obstacle arrangements demonstrates that the proposed framework outperforms conventional routing methods in achieving optimal paths for robots.

Keywords: Path planning, Multi-objective Optimization, Metaheuristic algorithms, Bat algorithm and PSO algorithm.

* Corresponding Author's email: mohadeseh.soleimanpour@yahoo.com

بهینه‌سازی مسیر ربات متحرک با رویکرد چندهدفه: ترکیب الگوریتم بهینه‌سازی ذرات و خفاش وفقی

محمد اسعدی^۱، زهرا محمدی^۲، حسین نظام آبادی پور^۲، محدثه سلیمانپور مقدم^{۱*}

^۱ مهندسی مکترونیک، مجتمع آموزش عالی بم، بم، ایران

^۲ مهندسی برق، دانشکده فنی و مهندسی، دانشگاه شهید باهنر کرمان، کرمان، ایران

تاریخ دریافت: ۱۴۰۴/۰۴/۲۱ تاریخ بازبینی: ۱۴۰۴/۱۲/۲۵ تاریخ پذیرش: ۱۴۰۵/۰۲/۰۸

نوع مقاله: پژوهشی

چکیده

این پژوهش یک چارچوب نوین برای برنامه‌ریزی مسیر مبتنی بر بهینه‌سازی چندهدفه ارائه می‌دهد که به طور همزمان سه معیار کلیدی شامل کمینه‌سازی طول مسیر، کاهش نرخ تغییرات زاویه‌ای و حداکثرسازی فاصله ایمن از موانع را مد نظر قرار می‌دهد. در مرحله نخست از ترکیب روش محدودیت اپسیلون و رویکرد جمع وزنی برای تبدیل مسئله بهینه‌سازی چندهدفه به یک مسئله تک‌هدفه استفاده شده است. به دلیل NP-hard بودن مسئله، یک الگوریتم فرا ابتکاری ترکیبی تطبیقی با عنوان PSO-ABA توسعه یافته است که از ادغام دو مکانیزم مکمل جستجوی سراسری مبتنی بر الگوریتم بهینه‌سازی ذرات (PSO) و الگوریتم خفاش انطباقی (ABA) بهره می‌برد. در این الگوریتم، وظیفه به‌روزرسانی موقعیت ربات بر عهده الگوریتم ABA است، در حالی که الگوریتم PSO به دنبال تنظیم پارامترهای مناسب می‌باشد. این مقاله به چند نکته کلیدی می‌پردازد: نخست، الگوریتم مذکور با اصلاح روابط موجود در الگوریتم خفاش، طراحی مسیری در فضای واقعی را هدف قرار می‌دهد به‌طوری‌که حرکت ربات تا حد امکان به سمت مسیر هدف هدایت شود و از چرخش‌های ناگهانی ربات جلوگیری گردد. دوم، با اصلاح تابع شایستگی، این الگوریتم مانع از به‌روزرسانی موقعیت ربات در داخل موانع می‌شود و مسیرهای نامعتبر را حذف می‌کند. ارزیابی عملکرد در محیط‌های شبیه‌سازی شده با چیدمان موانع پیچیده نشان می‌دهد که چارچوب پیشنهادی در مقایسه با روش‌های مرسوم مسیریابی، عملکرد بهتری در دستیابی به مسیر بهینه برای ربات دارد.

کلیدواژگان: برنامه‌ریزی مسیر، بهینه‌سازی چند هدفه، الگوریتم‌های فراابتکاری، الگوریتم خفاش و الگوریتم بهینه‌سازی ذرات.

* رایانامه نویسنده مسؤول: mohadese.soleimanpour@yahoo.com

۱- مقدمه

در سال‌های اخیر، پیشرفت‌های قابل توجهی در زمینه مسیریابی ربات‌ها رخ داده است. این پیشرفت‌ها نه تنها ناشی از توسعه فناوری‌های سخت‌افزاری است، بلکه به دلیل نوآوری در روش‌های نرم‌افزاری و الگوریتم‌هایی هستند که توانایی ربات‌ها را برای حرکت کارآمد در محیط‌های مختلف افزایش می‌دهند. طراحی الگوریتم مسیریابی ربات از اهمیت ویژه‌ای برخوردار است، زیرا شامل یافتن مسیرهای بهینه بین نقاط مشخص با حداقل کردن پارامترهایی مانند زمان، انرژی مصرفی یا پیچیدگی مسیر می‌شود. علاوه بر این، کاربردهای متنوع مسیریابی در حوزه‌هایی نظیر برنامه‌ریزی مسیر برای خودروهای زمینی بدون سرنشین [۱]، جمع‌آوری داده‌ها در شبکه‌های اینترنت اشیا [۲] و ربات‌های نظافتی در محیط‌های پویا با هدف اجتناب از برخورد با موانع اهمیت این موضوع را بیش از پیش نمایان ساخته است. این کاربردها نشان‌دهنده چالش‌ها و فرصت‌های جدیدی هستند که نیازمند توسعه روش‌های نوین مسیریابی می‌باشند. رویکردهای موجود برای برنامه‌ریزی مسیر ربات معمولاً به دو دسته اصلی تقسیم می‌شوند: رویکردهای کلاسیک که بر اساس اصول ریاضی و منطق صریح عمل می‌کنند و رویکردهای مبتنی بر فراابتکاری که از الهام‌گیری از پدیده‌های طبیعی برای یافتن راه حل استفاده می‌کنند [۳].

معروف‌ترین رویکرد کلاسیک در مسیریابی ربات، نمودار دیداری است [۴]. در این روش، از چندضلعی‌ها برای نمایش موانع استفاده می‌شود و ربات متحرک می‌تواند در امتداد لبه‌های این چندضلعی‌ها حرکت کرده و با جستجوی مجموعه خطوط، یک مسیر بهینه بین نقطه شروع و پایان را انتخاب کند. یکی دیگر از رویکردهای کلاسیک، نمودار ورونوی است [۵]. نمودار ورونوی گرافی است که یال‌های آن فضای آزاد بین موانع را به دو قسمت تقسیم می‌کنند. بگونه‌ای که هر یال از هر مانع اطراف خود به یک فاصله است. هدف این رویکرد افزایش فاصله ایمن از موانع است. مزیت این روش، سرعت محاسبه سریع و عیب آن جهش زیاد در طول حرکت است [۵]. هم‌چنین، رویکرد تجزیه سلولی در مسیریابی ربات، شامل تقسیم فضای عملی به مناطق ساده‌ای به نام سلول است [۶]. این روش با تبدیل فضا به یک شبکه متصل از سلول‌ها، امکان جستجوی مسیر بین نقطه شروع و هدف را فراهم می‌کند. با وجود این که این رویکرد در محیط‌های پیچیده موثر واقع می‌شود، اما ممکن است باعث ایجاد مسیرهای غیربهینه گردد. علاوه بر این، تکرار فرآیند

تقسیم سلول‌ها تا رسیدن به مرزهای راه‌حل می‌تواند زمان‌بر باشد و نیاز به محاسبات زیاد داشته باشد. روش توپولوژی [۷] یک رویکرد اساسی برای کاهش ابعاد است که مسئله مسیریابی در فضاهای هندسی با ابعاد بالا را به مسائل تفکیک‌شده با ابعاد کم تبدیل می‌کند. این روش با ایجاد شبکه توپولوژی، مسیر ربات را از نقطه شروع تا نقطه هدف مشخص می‌کند. این رویکرد، در مقایسه با روش تجزیه سلولی، نیاز به زمان ساخت مدل و فضای ذخیره‌سازی کمتری دارد و برای محیط‌هایی با مشخصات واضح و موانع پراکنده مناسب است. اما در شرایطی که تعداد موانع افزایش یا کاهش یابد، اصلاح شبکه توپولوژی بسیار چالش‌برانگیز است.

در کنار رویکردهای کلاسیک ریاضی که دقت بالایی داشته و در محیط‌های پیچیده با محدودیت‌هایی مواجه هستند، روش‌های فرا-ابتکاری توانسته‌اند جایگاه مهمی در مسیریابی ربات به دست آورند. در الگوریتم‌های فراابتکاری، تمام راه‌حل‌های ممکن یک مسئله، جمعیت اولیه الگوریتم را تشکیل می‌دهند. با توجه به طراحی هر الگوریتم، عملیات بروزرسانی جمعیت باعث حرکت عامل‌ها در جهت نقطه بهینه یا شبه‌بهینه می‌شود. این الگوریتم‌ها توانایی یافتن جواب‌های شبه‌بهینه را در زمان قابل قبول دارند؛ اما تضمینی برای یافتن جواب بهینه ندارند. بسیاری از الگوریتم‌های بهینه‌سازی مبتنی بر فراابتکار از رفتار موجودات طبیعی اقتباس شده‌اند و این ویژگی آن‌ها را برای حل مسائل پیچیده و پویا مناسب می‌کند [۸].

بهینه‌سازی جمعیت زنبور عسل^۱ (ABC) یکی از الگوریتم‌های فرا-ابتکاری مبتنی بر جمعیت است که رفتار جستجوی غذا در جمعیت‌های زنبور عسل را شبیه‌سازی می‌کند. این الگوریتم در سال ۲۰۰۵ برای مسائل بهینه‌سازی با پارامترهای واقعی ارائه شد [۱۰]. به دلیل سادگی، انعطاف‌پذیری و سهولت پیاده‌سازی، این الگوریتم توجه زیادی را جلب کرده و در حل بسیاری از مسائل بهینه‌سازی عملی مورد استفاده قرار گرفته است [۱۱].

در [۱۲] الگوریتم جمعیت زنبور عسل هدایت‌شده^۲ (DABC)، که نسخه پیشرفته‌تری از الگوریتم ABC می‌باشد، برای برنامه‌ریزی مسیر ربات پیشنهاد شده است. طبق این الگوریتم، زنبورهای پیش‌تری باید به منابع با کیفیت بالا و زنبورهای کم‌تری باید به منابع با کیفیت پایین فرستاده شوند یا حتی رها شوند. با پیروی از قوانین طبیعت، این زنبورها شهد گل را می‌گیرند و به منطقه خاصی در کندو می‌روند تا اطلاعات خود را با زنبورهای دیگر به اشتراک بگذارند [۱۲]. برنامه‌ریزی مسیر حرکت ربات متحرک با استفاده از الگوریتم بهینه‌سازی جمعیت مورچه^۳ (ACO) نیز در مرجع [۱۳] ارائه شده است. در روش ACO، مورچه‌های مصنوعی به وسیله حرکت بر روی محیط مسئله و با باقی‌گذاشتن نشانه‌هایی از

³ Ant Colony Optimization (ACO)

¹ Artificial Bee Colony

² Directed Artificial Bee Colony (DABC)

می‌شود [۲۰].

الگوریتم بهینه‌سازی خفاش^۴ (BA) [۲۱] از رفتار خفاش‌ها در یافتن طعمه الهام گرفته شده است. بدین صورت که خفاش‌ها پالس‌های صوتی منتشر می‌کنند و به صدای انعکاس‌یافته از موانع هنگام پرواز گوش می‌دهند. با استفاده از اختلاف زمان بین گوش‌هایشان، بلند بودن پاسخ و زمان تاخیر، خفاش‌ها می‌توانند سرعت، شکل و اندازه طعمه و موانع را دریابند. یک خفاش همچنین توانایی تغییر نحوه کار سونار خود را دارد؛ بدین صورت که اگر پالس‌های صوتی را با سرعت بالایی ارسال کند، می‌تواند با کسب جزئیات دقیق در مورد محیط اطراف، برای مدت زمان کم‌تری پرواز کند. در حالی که بلندی صدا معمولاً هنگامی که خفاش طعمه خود را پیدا می‌کند کاهش می‌یابد. با ادامه تکرارها، بلندی صدا و میزان انتشار پالس باید بروز شود. در سال ۲۰۱۸، تحقیق [۲۲] روشی نوین برای برنامه‌ریزی مسیر ربات‌های متحرک با ترکیب دو الگوریتم فراابتکاری الهام‌گرفته از طبیعت، یعنی جستجوی فاخته و BA، ارائه داده است. این روش که از مزایای هر دو الگوریتم بهره می‌برد، در مقایسه با استفاده از هر یک از آن‌ها به صورت جداگانه، زمان کمتری را برای یافتن هدف صرف می‌کند. یک نسخه بهبود یافته از الگوریتم BA به نام الگوریتم جمعیت خفاش با فرکانس اصلاح شده^۵ (MFB) در [۲۳] پیشنهاد شده است. در الگوریتم MFB، برای ایجاد تعادل مناسب میان اکتشاف و بهره‌برداری، از مکانیزم اصلاح فرکانس بهره گرفته شده است. در این روش، الگوریتم با فرکانس پایین آغاز می‌شود تا توانایی جستجوی سراسری افزایش یابد. سپس، با پیشرفت تکرارها، فرکانس به صورت تدریجی افزایش پیدا می‌کند تا تمرکز بیشتری بر بهره‌برداری و بهینه‌سازی محلی داشته باشد. شایان ذکر است که در [۲۳] تنها کمینه کردن فاصله طی شده ربات به عنوان هدف اصلی طراحی در نظر گرفته شده است. در مرجع [۲۴] برنامه‌ریزی مسیر ربات در محیط‌های ثابت با استفاده از الگوریتم ترکیبی PSO و MFB با عنوان PSO-MFB مطرح شده است. از مزایای این روش می‌توان به یافتن مسیر کوتاه و همچنین عدم برخورد با موانع اشاره کرد. اما عیب این روش، چرخش ناگهانی ربات ناشی از شکستگی خط مسیر است. علاوه بر این، نویسندگان در [۲۴]، پس از هر بار به‌روزرسانی موقعیت ربات، از یک الگوریتم اصلاحی استفاده کرده‌اند تا در صورت ورود ربات به داخل مانع، آن را به خارج از محدوده مانع هدایت کنند. این رویکرد نه تنها منجر به غیر بهینه شدن مسیر برنامه‌ریزی شده ربات می‌شود، بلکه پیچیدگی فرآیند را نیز افزایش

خود باعث می‌شوند که مورچه‌های مصنوعی بعدی بتوانند راه‌حل‌های بهتری را برای مسئله فراهم نمایند.

استفاده از الگوریتم‌های جمعیت باکتری‌ها^۱ (BC) به عنوان یک رویکرد نوین در جستجوی پاسخ بهینه، توجه برخی از محققان را به خود جلب کرده است [۱۴، ۱۵]. این الگوریتم‌ها با شبیه‌سازی رفتار جستجوی غذا در باکتری‌ها، به دنبال شناسایی مسیرهای بهینه یا نزدیک به بهینه در محیط‌های پیچیده هستند. برای بهبود کارایی الگوریتم BC در برنامه‌ریزی مسیر، نویسندگان در [۱۶] به تعریف موانع مجازی پرداخته‌اند تا از به دام افتادن در جواب‌های بهینه محلی جلوگیری کنند. این رویکرد اگرچه باعث افزایش دقت نتایج می‌شود، اما افزایش پیچیدگی پیاده‌سازی الگوریتم را به دنبال دارد. نکته قابل توجه این است که در اکثر مطالعات مورد بررسی، طول مسیر به عنوان هدف اصلی طراحی مسیر در نظر گرفته شده و به منظور تعریف تابع شایستگی مورد استفاده قرار گرفته است.

از سوی دیگر، در سال ۲۰۱۵، روشی مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات^۲ (PSO) برای ربات‌های متحرک معرفی گردید [۱۷]. این روش بر اساس رفتار اجتماعی پرندگان و ماهی‌ها طراحی شده است. به گونه‌ای که هر پرند در جمعیت دارای موقعیت و سرعت خاص خود است و بهترین موقعیت هر پرند بر اساس تابع هدف یا شایستگی تعیین می‌شود. همچنین، برنامه‌ریزی مسیر برای ربات‌های متحرک مستقل با استفاده از روش بهینه‌سازی PSO در [۱۸] پیشنهاد شده است که در مقایسه با الگوریتم مورچه، تحت شرایط محیطی مشابه، عملکرد بهتری را نشان می‌دهد. علاوه بر این، در [۱۹] نسخه دیگری از الگوریتم PSO برای برنامه‌ریزی مسیر ربات ارائه گردیده است. در این نسخه، در هر تکرار، موقعیت بهترین ذره به عنوان مکانی است که ربات باید به سمت آن حرکت کند. همچنین، موقعیت بهترین ذره در هر تکرار به عنوان نقطه شروع برای حرکت تمامی ذرات در تکرار بعدی در نظر گرفته می‌شود و این فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. با این حال، این رویکرد حریصانه می‌تواند منجر به طراحی مسیرهای غیر بهینه برای ربات شود. همچنین، الگوریتم PSO استاندارد به سبب نداشتن سازوکار تنظیم پویای پارامترها، معمولاً با مشکل همگرایی زودهنگام در بهینه‌های محلی مواجه است. در مقابل، الگوریتم PSO بهبود یافته^۳ (IPSO) با معرفی وزن‌های اینرسی تطبیقی، تعادل مطلوبی میان اکتشاف و بهره‌برداری برقرار کرده و موجب بهبود مسیر برنامه‌ریزی شده ربات در فضاها، جستجوی غیرمحدب و پیچیده

⁴ Bat Algorithm (BA)

⁵ Modified Frequency Bat (MFB)

¹ Bacteria Colony (BC)

² Particle Swarm Optimization (PSO)

³ Improved PSO

می‌دهد.

در حوزه مسیریابی رباتیک، الگوریتم جستجوی گرانشی (GSA)^۱ به عنوان یک روش مبتنی بر شبیه‌سازی قوانین فیزیکی نیوتنی و گرانش، برای برنامه‌ریزی مسیر ربات پیشنهاد شده است. در این چارچوب، ذرات (موقعیت‌های کاندید مسیر) تحت تأثیر نیروهای گرانشی متقابل قرار می‌گیرند که در نهایت منجر به همگرایی به سمت بهترین موقعیت می‌شود. با این حال، GSA مرسوم اغلب با چالش‌هایی از قبیل نرخ همگرایی پایین و تمایل به بهینه‌های محلی مواجه است که کارایی آن را در حل مسائل پیچیده مسیریابی تضعیف می‌کند [۲۵، ۲۶].

به منظور غلبه بر محدودیت GSA، گونه دیگری از آن، از جمله GSA بهبودیافته (IGSA)، توسعه یافته است که هدف اصلی آن بهبود سینماتیک الگوریتم و افزایش سرعت همگرایی برای کمینه‌سازی طول کل مسیر است [۲۵، ۲۶]. علاوه بر این، جهت ارتقای عملکرد و بهره‌گیری از مزایای الگوریتم‌های مختلف، در [۲۰] الگوریتم PSO بهبودیافته (IPSO) با IGSA ترکیب شده و الگوریتم IPSO-IGSA معرفی شده است که به طور خاص برای به حداقل رساندن طول مسیر طراحی شده است.

با این وجود، یک نقص اساسی در اکثر مطالعات مذکور، تمرکز صرف بر کوتاه‌ترین طول مسیر است، بدون آنکه همواری مسیر را در نظر بگیرند از این‌رو در تعداد بالا عملکرد نه چندان مناسبی از خود بروز می‌دهند. در مرجع [۲۷] با استفاده از دو معیار کلیدی کاهش طول مسیر و افزایش حاشیه ایمنی نسبت به موانع (کلیرنس)^۲ نسبت به موانع، به برنامه‌ریزی مسیر ربات پرداخته است. اگرچه این رویکرد فاصله ربات از موانع را به خوبی مدیریت می‌کند، اما چالش مربوط به چرخش‌های زاویه‌ای و دینامیک حرکت ربات کماکان بدون راه‌حل باقی مانده است.

بر این اساس، در این مقاله یک رویکرد نوین برای برنامه‌ریزی مسیر ربات ارائه شده است که شامل فرمول‌بندی مسئله برنامه‌ریزی مسیر به یک چارچوب بهینه‌سازی چندهدفه است. برای این منظور، سه تابع هدف اصلی شامل کمینه‌سازی طول مسیر، کاهش چرخش‌های زاویه‌ای ناگهانی و حداکثر کردن فاصله ایمن از موانع مورد توجه قرار گرفته‌اند. برای حل این مسئله، ابتدا از ترکیب روش‌های اپسیلون محدودیت^۳ و روش مجموع وزنی (WSM)^۴ استفاده شده است تا مسئله چندهدفه به یک مسئله تک هدفه تبدیل شود. با توجه به NP-hard بودن این مسئله، از ترکیب اصلاح‌شده دو

الگوریتم فراابتکاری، یعنی الگوریتم بهینه‌سازی ذرات PSO و الگوریتم خفاش، با عنوان الگوریتم بهینه‌سازی ذرات - خفاش وقتی^۵ (PSO-ABA) برای حل آن استفاده شده است. در این رویکرد، الگوریتم خفاش مسئول بروزرسانی مسیر ربات است. برای این منظور، به نکات زیر توجه شده است: نخست، روابط الگوریتم خفاش به گونه‌ای اصلاح شده‌اند که حرکت هر ذره تا حد ممکن در مسیر مستقیم به سمت هدف باشد. دوم، طول هر گام برای جلوگیری از چرخش‌های ناگهانی ربات (که ناشی از جلوگیری از برخورد با موانع در گام‌های بزرگ ربات است) کاهش یافته است.

علاوه بر این، الگوریتم PSO به منظور بهینه‌سازی پارامترهای کلیدی الگوریتم خفاش، به ویژه نرخ ارسال پالس و میزان بلندی صدا به کار گرفته می‌شود. در واقع، جمعیت اولیه PSO توسط مکان بروزرسانی شده هر ذره در الگوریتم خفاش ارزیابی می‌شود و در نهایت بهترین ذره به عنوان نتیجه نهایی در نظر گرفته می‌شود. همچنین، در این مقاله پیشنهاد شده است که تابع شایستگی به‌گونه‌ای تعریف شود که مانع از ورود ذرات به محدوده موانع گردد و بدین ترتیب، بهینه‌سازی مسیر با کارایی بیشتری امکان‌پذیر گردد. نتایج حاصل از این تحقیق نشان‌دهنده کارایی بالای چارچوب پیشنهادی در یافتن مسیر مطلوب برای ربات است.

در ادامه این تحقیق، در بخش دوم، مدل سیستم مورد استفاده برای برنامه‌ریزی مسیر تشریح شده و مسئله بهینه‌سازی چندهدفه فرمول‌بندی می‌شود. چالش‌های موجود در این زمینه، از جمله پیچیدگی محیط و نیاز به بهینه‌سازی چندهدفه، مورد بحث قرار می‌گیرد. در بخش سوم، چارچوب پیشنهادی برای برنامه‌ریزی مسیر ربات بر اساس الگوریتم ترکیبی PSO-ABA ارائه می‌شود. این چارچوب با استفاده از بهینه‌سازی چندهدفه، معیارهای طول مسیر، نرمی حرکت و فاصله از موانع را به‌طور همزمان بهینه می‌کند. در بخش چهارم، نتایج شبیه‌سازی برای ارزیابی کارایی چارچوب پیشنهادی ارائه شده و سناریوهای مختلف شبیه‌سازی می‌شوند. نتایج با روش‌های مرسوم مقایسه می‌شود تا کارایی آن مشخص شود. در نهایت، در بخش پنجم، یافته‌های اصلی مقاله خلاصه شده و مزایای چارچوب پیشنهادی نسبت به روش‌های موجود مورد تحلیل قرار می‌گیرد.

۲- مدل سیستم و تعریف مسئله

در این مقاله، یک فضای کاری مستطیلی به نام $\mathcal{X}_c$ به عنوان محیط فعالیت ربات در نظر گرفته می‌شود. هر نقطه i با بردار مکان $p_i =$

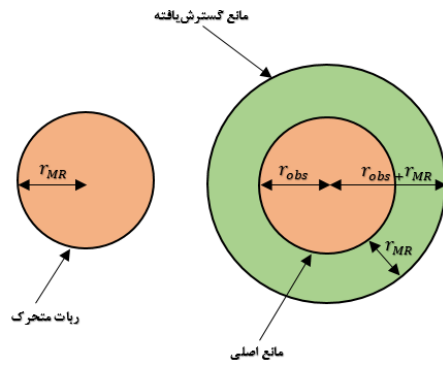
³ ϵ -Constraint

⁴ Weighted Sum Method (WSM)

⁵ PSO - Adaptive BA

¹ Gravitational Search Algorithm

² Clearance



شکل ۱. افزایش شعاع ربات به اندازه ابعاد ربات

بنابراین، مسئله مسیریابی بهینه‌ای که در این مقاله مورد بررسی قرار می‌گیرد، شامل سه تابع هدف طول مسیر، تغییر زاویه حرکت و فاصله ایمن از موانع است که در ادامه به تفصیل به تعریف هر یک از آن‌ها خواهیم پرداخت.

الف) طول مسیر

حداقل کردن طول مسیر یکی از مهم‌ترین اهداف در برنامه‌ریزی مسیر ربات‌ها است. این تابع، طول کل مسیر طی شده توسط ربات را محاسبه می‌کند و بصورت

$$J_L = \sum_{t=1}^T d_{t-1,t} \quad (1)$$

بیان می‌گردد که در آن $d_{t-1,t}$ طول گام t در مسیر است و بصورت

$$d_{t-1,t} = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}, t = 1, \dots, T \quad (2)$$

بدست می‌آید، که در آن مختصات دکارتی مکان ربات در i -امین گام ربات است. در این جا مقصود از (x_0, y_0) مختصات دکارتی نقطه شروع یا (x_s, y_s) است و $d_{0,1}$ طول اولین گام ربات است.

ب) تغییر زاویه حرکت

این تابع، مجموع تغییرات زاویه‌ای از مسیر مستقیم به سمت هدف در طول مسیر را اندازه‌گیری می‌کند و هدف آن کاهش این تغییرات برای داشتن یک مسیر هموارتر است. این تابع را نیز می‌توان به صورت زیر مدل کرد:

$$J_\theta = \sum_{t=1}^T |\theta_{(t,t-1)} - \theta_{(t-1,T)}| \quad (3)$$

که در آن

$$\theta_{(t,t-1)} = \tan^{-1} \frac{y_t - y_{t-1}}{x_t - x_{t-1}} \quad (4)$$

و

$$\theta_{(t-1,T)} = \tan^{-1} \frac{y_g - y_{t-1}}{x_g - x_{t-1}} \quad (5)$$

در این جا مقصود از $\theta_{(1,0)}$ زاویه حرکت اولین گام ربات است.

$[x_i, y_i]^T$ مشخص می‌شود که در آن (x_i, y_i) مختصات دکارتی نقطه i را نشان می‌دهد. هدف اصلی این تحقیق، برنامه‌ریزی مسیر بهینه برای ربات است به گونه‌ای که ربات از نقطه شروع با مختصات (x_s, y_s) به موقعیت هدف (x_g, y_g) برسد. در این راستا، T تعداد کل گام‌های ربات تا رسیدن به هدف در نظر گرفته می‌شود. فرض می‌شود که در فضای کاری، مجموعه‌ای از موانع ایستای دایروی با ابعاد مختلف وجود دارد که به صورت $O = \{O_1, \dots, O_M\}$ تعریف شده‌اند.

این موانع باید در فرآیند مسیریابی مد نظر قرار گیرند تا ربات بتواند به طور ایمن و کارآمد به مقصد خود برسد. به علاوه، فرضیات زیر برای طراحی الگوریتم مسیریابی در نظر گرفته می‌شوند:

- همان طور که در شکل ۱ نشان داده شده است، اندازه فیزیکی ربات از طریق افزایش شعاع موانع در مدل لحاظ شده است. این رویکرد که در بسیاری از مطالعات رباتیک مورد استفاده قرار می‌گیرد، امکان مدل‌سازی ربات به صورت جرم نقطه‌ای را در الگوریتم‌های برنامه‌ریزی مسیر فراهم می‌کند، بدون آنکه دقت مدل کاهش یابد [۲۴].
- در این پژوهش فرض شده است که ربات‌ها از نوع ربات‌های زمینی همه‌جهته^۱ هستند؛ به گونه‌ای که می‌توانند بدون نیاز به تغییر جهت، در هر راستایی حرکت کنند. این نوع ربات‌ها به دلیل مانورپذیری بالا، به‌طور گسترده در محیط‌های داخلی و سناریوهای شهری مبتنی بر اینترنت اشیا مورد استفاده قرار می‌گیرند و امکان ناوبری کارآمد در فضاهای محدود را فراهم می‌کنند [۲۸، ۲۹].
- در مدل پیشنهادی، موقعیت و ابعاد موانع از پیش مشخص در نظر گرفته شده است. این فرض با شرایط عملیاتی رایج در محیط‌های کنترل‌شده مانند انبارهای هوشمند و فضاهای داخلی شهری همخوانی دارد. در این محیط‌ها نقشه موانع پیش از استقرار ربات‌ها تهیه می‌شود تا امکان برنامه‌ریزی غیربرخط^۲ و قابل اطمینان مسیر ربات فراهم گردد [۳۰].
- همچنین فرض شده است که سنسورهای ربات قادرند موانع را تا شعاع ۰/۸ متر تشخیص دهند [۲۴].

همان‌طور که پیش‌تر اشاره شد، هدف این مقاله طراحی مسیری برای ربات است که به طور همزمان سه معیار اساسی حداقل طول مسیر، چرخش‌های زاویه‌ای کمینه و حداکثر فاصله ایمن از موانع را مد نظر قرار دهد.

² Offline

¹ Omnidirectional

ج) فاصله ایمن از موانع

همان‌گونه که می‌دانیم، میزان ایمنی مسیر برنامه‌ریزی شده برای ربات به کمینه فاصله هر یک از نقاط مسیر برنامه‌ریزی شده از هر یک از موانع دارد. چرا که به دلیل عدم قطعیت‌های ناشی از مکان موانع و حرکت ربات افزایش این فاصله به حفظ ایمنی ربات و محیط اطراف آن کمک می‌کند. این فاصله با عنوان کلیرنس شناخته می‌شود و به صورت زیر تعریف می‌شود:

$$J_C = \min_{t \in \{1,2,\dots,T\}} \min_{p \in O} (d_{t,p}) \quad (6)$$

که در آن $d_{t,p}$ فاصله بین مکان ربات در گام t و نقطه p متعلق به مجموعه موانع O است و عبارت است از

$$d_{t,p} = \sqrt{(x_t - x_p)^2 + (y_t - y_p)^2}, t = 1, \dots, T \quad (7)$$

بطوری که (x_p, y_p) مختصات دکارتی نقطه p است. لذا، به عنوان هدفی دیگر از برنامه‌ریزی مسیر ربات، بیشینه‌سازی این فاصله است. در مجموع، چنان‌چه $\mathbf{x} = [x_1, \dots, x_T]^T$ و $\mathbf{y} = [y_1, \dots, y_T]^T$ مسئله برنامه‌ریزی مسیر ربات به صورت مسئله بهینه‌سازی چند هدفه زیر قابل بیان است:

$$[\mathbf{x}^T, \mathbf{y}^T]^T = \{\min J_L, \min J_\theta, \max J_C\} \quad (8)$$

s.t.

$$(\bar{A}-8), (1), (3) \text{ و } (6) \quad (\bar{A}-8)$$

$$d_{(t,g)} < d_{(t-1,g)}, \forall t \quad (\text{ب}-8)$$

$$(x_t, y_t) \in \mathcal{X}_c, \forall t \quad (\text{ج}-8)$$

که در آن، $(\bar{A}-8)$ مجموع طول مسیر ربات، تغییر زاویه حرکت ربات و فاصله مسیر برنامه‌ریزی شده را از موانع اندازه‌گیری می‌کند. علاوه بر این، قید $(\text{ب}-8)$ بیانگر این است که با هر گام، ربات باید به هدف نزدیک‌تر شود. در نهایت، قید $(\text{ج}-8)$ تضمین می‌کند که هر گام ربات باید در داخل فضای کاری قرار گیرد. با توجه به اینکه هر یک از اهداف می‌توانند با یکدیگر در تعارض باشند، معمولاً یافتن یک نقطه بهینه واحد که تمام اهداف را ارضا کند، ممکن نیست. بنابراین، هدف اصلی پیدا کردن مجموعه‌ای از راه‌حل‌های پارتو است که هیچ کدام نسبت به دیگری برتری ندارند. در این مقاله، ابتدا از روش اپسیلون محدودیت برای تبدیل تابع هدف مربوط به حداکثر کردن فاصله ایمن از موانع به یک قید به صورت

$$J_C \geq \epsilon \quad (9)$$

استفاده شده است. بدین ترتیب، در ادامه مسئله چندهدفه با دو تابع هدف اصلی (طول مسیر کمینه و چرخش زاویه‌ای کمینه) حل می‌شود در حالی که فاصله ایمن با پارامتر ϵ کنترل می‌گردد. سپس از روش WSM برای حل مسئله استفاده می‌گردد. مطابق با رویه

رایج در روش مجموع وزن‌دار WSM در مسائل بهینه‌سازی چندهدفه، ابتدا توابع هدف نرمال‌سازی شده و سپس ضرایب وزن بر آن‌ها اعمال می‌شود. این فرآیند باعث می‌شود اختلاف مقیاس و واحد اندازه‌گیری توابع هدف بر نتیجه نهایی تأثیر نامطلوب نداشته و هر یک سهم متعادلی در فرآیند بهینه‌سازی داشته باشند. بر این اساس، تابع هدف کلی $J(\mathbf{x}, \mathbf{y})$ به صورت زیر تعریف می‌شود:

$$J(\mathbf{x}, \mathbf{y}) = \omega_1 N_L + \omega_2 N_\theta \quad (10)$$

که در آن ω_1 و ω_2 وزن‌های تخصیص‌یافته به هر هدف هستند به‌گونه‌ای که $\omega_1 + \omega_2 = 1$. همچنین N_L و N_θ به ترتیب توابع نرمال‌شده مربوط به طول مسیر و مجموع چرخش هستند که به صورت $N_\theta = \frac{J_\theta}{J_\theta^{max}}$ و $N_L = \frac{J_L}{J_L^{max}}$ تعریف می‌شوند. در این روابط J_L^{max} بیانگر بدترین حالت ممکن برای طول مسیر (J_L) در فضای جستجو است (به عنوان مثال مسیری که در اطراف مرزهای محیط حرکت می‌کند). همچنین، J_θ^{max} حداکثر چرخش تجمعی ربات (J_θ) را نشان می‌دهد که به صورت $T \cdot \theta'$ در نظر گرفته می‌شود؛ در اینجا T تعداد گام‌های حرکت ربات و θ' حداکثر چرخش مجاز در هر گام است.

همان‌طور که از رابطه ۱۰ بر می‌آید، این روش امکان ترکیب توابع مختلف را با توجه به اولویت‌های خاص مسئله فراهم کرده و ما را قادر می‌سازد تا بهترین تعادل بین طول مسیر کوتاه‌تر و صافی بیشتر آن ایجاد کنیم. در مجموع، مسئله بهینه‌سازی برنامه‌ریزی مسیر به صورت

$$[\mathbf{x}^T, \mathbf{y}^T]^T = \min (J(\mathbf{x}, \mathbf{y}) = \omega_1 J_L + \omega_2 J_\theta) \quad (11)$$

s.t.

$$(\bar{A}-11), (\bar{A}-8) \text{ و } (\text{ب}-8) \text{ و } (\text{ج}-8) \quad (\bar{A}-11)$$

$$J_C \geq \epsilon \quad (\text{ب}-11)$$

بازنویسی می‌شود.

از آن‌جا که مسئله بهینه‌سازی مطرح‌شده در این پژوهش مستلزم تعیین بهترین ترتیب یا دنباله عبور از میان مجموعه‌ای از نقاط است، ماهیت بنیادین آن از نوع مسائل بهینه‌سازی ترکیبی^۱ است. در این‌گونه مسائل، تعداد راه‌حل‌های ممکن بسیار زیاد بوده و بررسی فضای وسیعی از جواب‌ها ضروری است؛ از این‌رو، مسائل ترکیبی از پیچیدگی محاسباتی قابل توجهی برخوردار بوده و در دسته مسائل NP-hard قرار می‌گیرند.

به‌طور کلی، برای مسائل NP-hard هیچ الگوریتم قطعی شناخته‌شده‌ای وجود ندارد که بتواند در زمان چندجمله‌ای، دستیابی به پاسخ بهینه سراسری را تضمین کند. بنابراین، استفاده

¹ Combinatorial

محاسبه می‌شود، که در آن p^* بهترین موقعیت پیدا شده در بین m خفاش و v_t^i سرعت i -امین خفاش در تکرار قبل را بیان می‌نماید. همچنین، f_t فرکانس ارسال پالس در تکرار t می‌باشد. لازم به ذکر است که فرکانس به صورت

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta_t \quad (14)$$

بیان می‌گردد که در آن $\beta_t = t * e^{(-p*r)}$ عدد تکرار، r یک عدد تصادفی در بازه $[0,1]$ و مقدار p معمولاً 0.10 در نظر گرفته می‌شود [۲۱]. بدین ترتیب، اطمینان حاصل می‌شود که فرکانس به تدریج با پیشرفت تکرارها افزایش خواهد یافت.

برای ایجاد تنوع و جلوگیری از همگرایی زود هنگام، الگوریتم تصمیم می‌گیرد که آیا از این موقعیت جدید استفاده کند یا خیر. بگونه‌ایکه خفاش با صدای کمتر (یعنی A کوچکتر) به احتمال بیشتری موقعیت جدید را قبول می‌کند. از این‌رو، هنگامی که یک نتیجه از بهترین راه‌حل‌های موجود در هر مرحله خاص انتخاب می‌شود، برای هر خفاش یک راه‌حل جدید به صورت محلی نیز تولید می‌شود. این فرآیند بر اساس اصل مسیریابی تصادفی^۱ انجام می‌گیرد و به الگوریتم اجازه می‌دهد تا به طور مؤثری در اطراف نقاط بهینه محلی حرکت کند و راه‌حل جدیدی به صورت

$$p_{new} = p^* + \sigma \delta A(t)1 \quad (15)$$

را کشف نماید. در رابطه فوق، δ یک عدد تصادفی در بازه $[-1,1]$ ، $A(t)$ متوسط بلندی صدای تمام خفاش‌ها در مرحله t و σ یک پارامتر مقیاس‌گذاری است که برای کنترل اندازه گام در نظر گرفته شده است. این فرآیند معمولاً با پارامترهایی مانند بلندی صدای زیاد و نرخ ارسال پالس پایین همراه است که باعث می‌شود خفاش‌ها بیشتر به جستجوی نقاط جدید بپردازند و از گیر افتادن در کمینه‌های محلی جلوگیری شود.

با ادامه تکرارها، بلندی صدا معمولاً هنگامی که خفاش طعمه خود را پیدا می‌کند کاهش و میزان انتشار پالس با توجه به معادلات زیر افزایش می‌یابد. نحوه بروزرسانی هر یک به ترتیب در زیر آورده شده است:

$$A_i(t+1) = \alpha A_i(t) \quad (16)$$

و

$$r_i(t+1) = r_i(0)[1 - \exp(-\gamma t)] \quad (17)$$

که در آن $0 < \alpha < 1$ و $\gamma > 0$ پارامترهای طراحی هستند. در الگوریتم خفاش، پارامترهای نرخ ارسال پالس و بلندی صدا نقش حیاتی در تعادل بین اکتشاف و بهره‌برداری فضای جستجو ایفا می‌کنند. در روش‌های مرسوم، این پارامترها به صورت ثابت یا

از الگوریتم‌های فراابتکاری برای یافتن راه‌حل‌های کارا در این دسته از مسائل کمک‌ساز است. در این مقاله از الگوریتم PSO-ABA استفاده می‌شود. در بخش بعد، جزئیات مربوط به پیاده‌سازی و عملکرد این روش ارائه خواهد شد.

۳- الگوریتم‌های بهینه‌سازی مبتنی بر جمعیت

هوش جمعی یک چارچوب محاسباتی الهام‌گرفته از طبیعت شناخته می‌شود که در آن، برهم‌کنشی میان موجودات یا عامل‌های ساده، منجر به شکل‌گیری رفتارهای هوشمندانه در سطح کلان می‌شود. این رویکرد به دلیل اثربخشی بالا در حل مسائل بهینه‌سازی، کاربردهای گسترده‌ای در حوزه‌های مختلف یافته است. فرآیند جستجو در الگوریتم‌های مبتنی بر هوش جمعی به صورت تکرارشونده انجام می‌پذیرد تا زمانی که معیارهای توقف، شامل دستیابی به پاسخ بهینه یا رسیدن به تعداد حداکثر تکرارهای تعیین‌شده، برآورده شود. توانایی منحصر به فرد این الگوریتم‌ها در حل مسائل پیچیده و چندبعدی، موجب توسعه گونه‌های متعددی از آن‌ها شده است. در این پژوهش، دو نمونه از این الگوریتم‌ها مورد بررسی و استفاده قرار گرفته‌اند که در ادامه مبانی نظری و نحوه دستیابی به جواب بهینه در هر یک تشریح شده است.

۳-۱- معرفی الگوریتم BA

مکان‌یابی از ویژگی‌های مهم رفتار خفاش است. خفاش‌ها پالس‌های صوتی منتشر می‌کنند و به صدای انعکاس‌یافته از موانع هنگام پرواز گوش می‌دهند. خفاش با استفاده از اختلاف زمان بین گوش‌هایش، بلند بودن پاسخ و زمان تاخیر می‌تواند سرعت، شکل و اندازه طعمه و موانع را دریابد. یک خفاش همچنین توانایی تغییر نحوه ردیابی آوایی خود را دارد. اگر پالس‌های صوتی را با سرعت بالایی ارسال کند، می‌تواند با کسب جزئیات دقیق در مورد محیط اطراف، برای مدت زمان کم‌تری پرواز کند. بروزرسانی موقعیت خفاش i در الگوریتم خفاش مرسوم به شرح زیر است:

$$p_{t+1}^i = p_t^i + v_{t+1}^i \quad (12)$$

لازم به ذکر است که بردارهای $p_t^i = [x_t^i, y_t^i]^T$ و $p_{t+1}^i = [x_{t+1}^i, y_{t+1}^i]^T$ به ترتیب بیان‌گر موقعیت خفاش در تکرار t و $t+1$ است. همچنین، v_{t+1}^i سرعت خفاش را در تکرار $t+1$ توصیف می‌نماید که به صورت

$$v_{t+1}^i = v_t^i + (p_t^i - p^*) * f_t \quad (13)$$

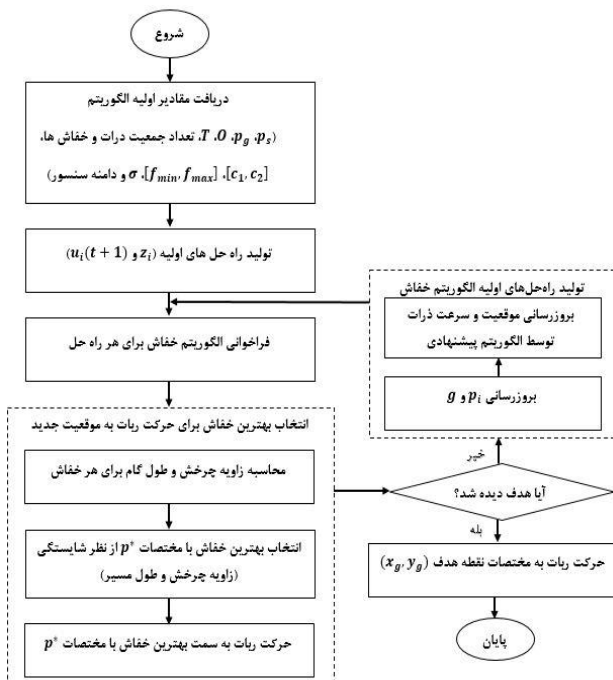
¹ Random Walk Principle

$$w_i = w_{min} + (w_{max} - w_{min}) \frac{d_i}{\max_i d_i}, i = 1, \dots, N_p \quad (20)$$

که در آن w_{min} و w_{max} به ترتیب بیان‌گر کم‌ترین و بیشترین مقدار وزن اینرسی است. d_i فاصله ذره i تا بهترین ذره و N_p تعداد ذرات الگوریتم PSO است. طبق رابطه (20)، مقدار این وزن در ابتدای اجرای الگوریتم با مقدار بزرگ‌تری تنظیم می‌شود که سبب جستجوی گسترده فضا می‌شود. به مرور زمان و با همگرایی ذرات کاهش می‌یابد که سبب تمرکز در فضای کوچک می‌شود.

۴- روش پیشنهادی PSO-ABA

در این بخش، یک رویکرد نوین برای برنامه‌ریزی مسیر ربات با استفاده از ترکیب الگوریتم‌های PSO (بهینه‌سازی ازدحام ذرات) و BA (الگوریتم خفاش) و فقی ارائه می‌شود که از این پس به اختصار PSO-ABA نامیده می‌شود. در الگوریتم پیشنهادی PSO-ABA، تلاش می‌شود تا با اعمال محدودیت‌های مناسب، مسیر ربات را به شرایط واقعی نزدیک‌تر کنیم. این الگوریتم تضمین می‌کند که ربات با حفظ فاصله ایمن از موانع، بدون برخورد با آن‌ها عبور کند و با پی‌موندن مسیری بهینه، کم‌ترین مسافت را با نرمی حرکت از نقطه شروع به موقعیت هدف برسد. خلاصه مراحل روش پیشنهادی PSO-ABA در شکل ۲ نشان داده شده است.



شکل ۲. فلوچارت خلاصه مراحل روش پیشنهادی برای برنامه‌ریزی مسیر بهینه ربات متحرک با استفاده از الگوریتم ترکیبی PSO-ABA

تصادفی تنظیم می‌شوند که ممکن است منجر به همگرایی زودرس یا ناکارآمدی الگوریتم شود.

۳-۲ معرفی روش PSO

الگوریتم PSO با شبیه‌سازی رفتار دسته‌جمعی پرندگان یا ماهی‌ها، توانایی جستجوی فضای بزرگ پارامترها را دارد. این الگوریتم از مجموعه‌ای از پاسخ‌ها به عنوان ذره استفاده می‌کند که این پاسخ‌ها تا زمانی که یک پاسخ بهینه یافت شود یا شرایط خاتمه الگوریتم محقق شود به حرکت خود ادامه می‌دهند. این کار با بروزرسانی مداوم موقعیت ذرات بر اساس بهترین تجربه فردی و جمعی انجام می‌شود. در این الگوریتم موقعیت هر ذره (z_i) و سرعت آن (u_i) به صورت تصادفی آغاز می‌شوند و در مراحل بعد با استفاده از روابط زیر بروزرسانی می‌شوند:

$$z_i(t+1) = z_i(t) + u_i(t+1) \quad (18)$$

و

$$u_i(t+1) = wu_i(t) + c_1r_1(p_i - z_i(t)) + c_2r_2(g - z_i(t)) \quad (19)$$

که در آن p_i بیان‌گر بهترین موقعیت پیدا شده برای ذره i و g بهترین موقعیت پیدا شده در کل جمعیت است. همان‌طور که در رابطه (19) مشخص است، سرعت ذره i تحت تاثیر سه مولفه است: سرعت قبلی ذره $u_i(t)$ ، یک جاذبه خطی به سمت بهترین موقعیت پیدا شده برای ذره $(p_i - z_i(t))$ که با وزن c_1 و یک عدد تصادفی $r_1 \in [0,1]$ مقیاس‌بندی می‌شود. مولفه نهایی، یک جاذبه خطی به سمت بهترین موقعیت پیدا شده در کل جمعیت است $(g - z_i(t))$ که با وزن c_2 و یک عدد تصادفی $r_2 \in [0,1]$ مقیاس‌بندی می‌شود. جمعیت اولیه در الگوریتم PSO به صورت تصادفی در سرتاسر فضای جستجو مقداردهی می‌شوند که این موقعیت‌های اولیه به عنوان بهترین تجربه شخصی ذرات نیز شناخته می‌شوند. در گام بعد بهترین ذره از میان ذرات موجود انتخاب و به عنوان بهترین پاسخ شناخته می‌شود. لازم به ذکر است که w وزن اینرسی است که تاثیر سرعت ذرات در گام قبل را بر سرعت فعلی تعیین می‌نماید. انتخاب مناسب این پارامتر تاثیر زیادی در ایجاد مصالحه بین اکتشاف^۱ و استخراج^۲ الگوریتم دارد. بگونه‌ای با انتخاب مقدار بزرگ‌تر این پارامتر قابلیت جستجوی عمومی الگوریتم بهبود می‌یابد و فضای بیشتری مورد استفاده قرار می‌گیرد. در مقابل، هر چه میزان کمتری داشته باشد فضای جستجو محدود شده و جستجو در فضای کوچک‌تری صورت می‌گیرد. به منظور بهبود عملکرد الگوریتم، این پارامتر برای هر یک از ذرات از طریق رابطه زیر به دست می‌آید:

² Exploitation

¹ Exploration

از چرخش‌های ناگهانی و به دام افتادن در بهینه‌های محلی می‌شود. همچنین حرکت ربات به سمت هدف شده و در نتیجه صاف‌تر شدن مسیر را به دنبال خواهد داشت. تفاوت اصلی روش پیشنهادی با الگوریتم اصلی بهینه‌سازی خفاش در بروزسانی موقعیت ذرات است و دیگر مراحل و روابط مانند الگوریتم بهینه‌سازی خفاش است. هم- چنین نحوه محاسبه مکان بهینه ذرات p^* به صورت زیر است:

$$p^* = \operatorname{argmax}_{i=1, \dots, N_a} F_{t,i}(x, y) \quad (24)$$

که در آن $F_{t,i}(x, y)$ مقدار تابع شایستگی خفاش i در گام t را نشان می‌دهد که در ادامه به بررسی آن پرداخته می‌شود. همچنین، N_a تعداد ذرات الگوریتم ABA است.

نکته مهم در برنامه‌ریزی مسیر این است که اگر فاصله نقطه تولید شده تا مرکز مانع m ($d_{t+1,m}$) از شعاع مانع به علاوه یک حداقل فاصله ایمن (ϵ) کمتر باشد، یعنی نقطه درون مانع قرار دارد. از این‌رو، بررسی شرط زیر باید صورت گیرد:

$$d_{t+1,m} < r_m + \epsilon, \quad m = 1, \dots, M \quad (25)$$

که در آن r_m شعاع مانع m ، ϵ حداقل فاصله ایمن ربات از مانع و $d_{t+1,m}$ فاصله اقلیدسی نقطه تولیدی $t+1$ تا مانع m است و بصورت

$$d_{t+1,m} = \sqrt{(x_{t+1} - x_m)^2 + (y_{t+1} - y_m)^2} \quad (26)$$

محاسبه می‌شود. در این مقاله، این محدودیت در تابع شایستگی در نظر گرفته شده است. بگونه‌ای که، عملکرد کلی تابع شایستگی هر یک از ذرات با رابطه زیر ارائه می‌شود:

$$fitness = \frac{1}{F(x, y) + q} \quad (27)$$

که در آن

$$F(x, y) = \left(\max \left(\epsilon - \min_{j \in 1, \dots, M} d_{t+1,j}, 0 \right) \right) J(x, y) + J(x, y) \quad (28)$$

که $J(x, y)$ تابع هدف تعریف شده در رابطه (۱۰) است. همان‌گونه که از رابطه (۲۸) مشخص است، چنان‌چه فاصله هر گام هر یک از ذرات از هر یک از موانع کمتر از فاصله ایمن باشد، مقدار تابع شایستگی به طور قابل توجهی کاهش می‌یابد. این امر موجب حذف ذرات (راه‌حل‌های) نامعتبر می‌گردد. به علاوه، q یک مقدار مثبت برای جلوگیری از صفر شدن مخرج مورد استفاده قرار می‌گیرد.

لازم به ذکر است که به منظور جلوگیری از به دام افتادن در مسیرهای غیربهینه، پس از طی هر گام، از یک همسایگی دایروی طبق رابطه (۱۵) حول بهترین مسیر یافته شده استفاده می‌شود. ایجاد این همسایگی به میزان بلندی صدای خفاش و نرخ ارسال پالس وابسته است، به‌گونه‌ای که در صدای پایین و نرخ ارسال بالا، احتمال به کارگیری آن کاهش می‌یابد.

در حقیقت، در الگوریتم خفاش، پارامترهای نرخ ارسال پالس و بلندی صدا نقش حیاتی در تعادل بین اکتشاف و بهره‌برداری فضای

در این روش، الگوریتم ABA وظیفه به‌روزرسانی موقعیت ربات را بر عهده دارد، در حالی که الگوریتم PSO به یافتن مقادیر بهینه برای پارامترهای ABA می‌پردازد. در طراحی این روش، سه نکته کلیدی به شرح زیر مورد توجه قرار گرفته است:

۱. اصلاح تعریف تابع شایستگی: تابع شایستگی به گونه‌ای تعریف شده است که ضمن حفظ حداقل میزان ایمنی مسیر و جلوگیری از بروزسانی مکان ربات در داخل موانع، طول مسیر را تا حد امکان کوتاه و هموار نگه دارد. این اصلاحات، عملکرد الگوریتم را در یافتن مسیرهای مناسب بهبود می‌بخشد.

۲. انتخاب هوشمندانه نقاط اولیه: نقاط اولیه برای شروع فرآیند بهینه‌سازی به گونه‌ای انتخاب می‌شوند که از تکرار بیش از حد الگوریتم و همگرایی زود هنگام به سمت نقاط بهینه محلی جلوگیری شود.

۳. اصلاح روابط الگوریتم خفاش: روابط مورد استفاده در الگوریتم خفاش به منظور انطباق با مسئله برنامه‌ریزی مسیر، اصلاح شده‌اند. این اصلاحات به جلوگیری از به دام افتادن الگوریتم در نقاط بهینه محلی کمک می‌کند و امکان جستجوی گسترده‌تر در فضای راه‌حل‌ها را فراهم می‌سازد.

در ادامه این بخش، جزئیات مربوط به رویکرد پیشنهادی به طور کامل مورد بحث و بررسی قرار می‌گیرد.

۴-۱- الگوریتم ABA

در این مقاله، جهت بهبود روند مسیریابی و مراحل بعدی تکرار الگوریتم، مختصات نقطه اولیه را طوری تولید می‌کنیم که در راستای موقعیت هدف باشد.

از آن‌جا که هدف در این مقاله کاهش انحراف زاویه‌ای است، جهت بروزسانی موقعیت ذره، یک ضریب q ($0 < q < 1$) برای سرعت بروزسانی شده و زاویه ربات نسبت به موقعیت هدف (ϕ) طبق رابطه زیر استفاده شده است:

$$x_{t+1}^i = x_t^i + |v_{t+1}^i| \times q \times \cos \phi, \quad (29)$$

$$y_{t+1}^i = y_t^i + |v_{t+1}^i| \times q \times \sin \phi, \quad (30)$$

که در آن

$$\phi = \tan^{-1} \frac{y_g - y_t}{x_g - x_t} \quad (31)$$

همچنین، $p_{t+1}^i = [x_{t+1}^i, y_{t+1}^i]^T$ و $p_t^i = [x_t^i, y_t^i]^T$ به ترتیب بیانگر موقعیت ذره در تکرار t و $t+1$ را نشان می‌دهد. همچنین، سرعت ذره در تکرار $t+1$ را نشان می‌دهد که در رابطه (۱۳) تعریف شده است.

این عمل منجر به کاهش طول گام‌های ربات و در نتیجه جلوگیری

همچنین، پارامترهای شبیه‌سازی در جدول ۱ آورده شده‌اند. به علاوه، تمامی الگوریتم‌های مورد بررسی در یک محیط شبیه‌سازی یکسان و تحت شرایط آزمایشی مشابه اجرا شده‌اند. بدین ترتیب، ساختار محیط، موقعیت موانع، شرایط اولیه و پارامترهای شبیه‌سازی برای تمامی روش‌ها یکسان در نظر گرفته شده است.

۵-۱- مقایسه با الگوریتم PSO-MFB

در بخش اول شبیه‌سازی، عملکرد روش پیشنهادی با رویکرد برنامه-ریزی مسیر PSO-MFB مقایسه شده است.

در شبیه‌سازی نخست، عملکرد روش پیشنهادی در محیطی با سه مانع بررسی شده است. مختصات و ابعاد هر یک از موانع در جدول ۲ آورده شده است. مسیر برنامه‌ریزی شده طبق روش پیشنهادی و روش PSO-MFB در شکل ۳ ترسیم گردیده است. همچنین، جهت سهولت در مقایسه، نتایج به‌دست آمده از آزمایش در جدول ۳ ارائه شده است. مطابق با شکل ۳ و جدول ۳، مشاهده می‌شود که مسیر طراحی شده توسط الگوریتم پیشنهادی به‌طور قابل توجهی صاف‌تر است، به‌طوری که زاویه چرخش ربات نسبت به روش PSO-MFB، ۱۹/۴۳۱۲ درجه کمتر است. این کاهش در زاویه چرخش، نشان‌دهنده عدم وجود تغییرات ناگهانی در مسیر طراحی شده است و طول گام‌ها به‌طور معقولی کمتر از روش پیشین تعیین شده است. هرچند مسیر طراحی شده توسط الگوریتم PSO-MFB، ۰/۱۶۲۶ متر کوتاه‌تر است، اما مزایای چشمگیر الگوریتم پیشنهادی در راستای کاهش چرخش‌های ناگهانی و بهبود ثبات مسیر، اهمیت بیشتری دارد. به‌عبارتی، با کاهش چرخش‌های ناگهانی و بهینه‌سازی طول گام‌ها، عملکرد ربات در محیط‌های پیچیده و با موانع متعدد به‌طور چشمگیری بهبود می‌یابد.

جدول ۱. پارامترهای شبیه‌سازی

نام پارامتر	مقدار
تعداد تکرار	۵
تعداد جمعیت	۵
$[c_1, c_2]$	$[2, 2]$
$[f_{min}, f_{max}]$	$[0, 10]$
$[w_{min}, w_{max}]$	$[0.4, 0.9]$
ϵ	0.15
محدوده سنسور	۰.۸ متر

جدول ۲. مشخصات موانع محیطی با ۳ مانع

مختصات مرکز	شعاع مانع	شماره مانع
$[3, 3.2]$	1.8	1
$[5, 7]$	1.4	2
$[7, 5]$	1	3

جستجو ایفا می‌کنند. در روش‌های مرسوم، این پارامترها به صورت ثابت یا تصادفی تنظیم می‌شوند که ممکن است منجر به همگرایی زودرس یا ناکارآمدی الگوریتم شود. برای رفع این چالش، استفاده از الگوریتم PSO پیشنهاد می‌شود. در این روش، هر ذره در PSO نماینده‌ی یک مجموعه از پارامترهای الگوریتم خفاش (α, γ) است و با ارزیابی عملکرد الگوریتم خفاش برای هر ذره، بهترین ترکیب پارامترها شناسایی می‌شود.

۴-۲- الگوریتم PSO

الگوریتم PSO با شبیه‌سازی رفتار دسته‌جمعی پرندگان یا ماهی‌ها، توانایی جستجوی فضای بزرگ پارامترها را دارد تا بهترین تنظیمات ممکن را برای پارامترهای شدت صوت و نرخ ارسال پالس پیدا کند. این کار با بروزرسانی مداوم موقعیت ذرات بر اساس بهترین تجربه فردی و جمعی انجام می‌شود. هر ذره در PSO به صورت $z = [\alpha, \gamma]^T$ تعریف می‌شود و بروزرسانی هر یک مطابق رابطه

$$z_j(n+1) = z_j(n) + u_j(n+1), j = 1, \dots, N_p \quad (29)$$

و

$$u_j(n+1) = wu_j(n) + c_1r_1(z_j^*(n) - z_j(n)) + c_2r_2(z^* - z_j(n)) \quad (30)$$

است. عملکرد هر ذره با اجرای الگوریتم خفاش با پارامترهای روی مسئله هدف ارزیابی می‌شود، بدین معنی که $z_j^*(n)$ و z^* از طریق محاسبه عبارات ذیل حاصل می‌شود:

$$z_j^*(n) = \operatorname{argmax}_{i=1, \dots, n} F_{i,j}(x, y), \forall j \quad (31)$$

و

$$z^* = \operatorname{argmax}_{j=1, \dots, N_p, i=1, \dots, n} F_{i,j}(x, y). \quad (32)$$

$F_{i,j}(x, y)$ همان مقدار تابع شایستگی است که در تکرار i توسط ذره j محاسبه می‌شود.

۴-۳- پایان الگوریتم

بعد از رد شدن از آخرین مانع، ربات با دیدن موقعیت هدف به سمت آن حرکت کرده و دیگر نیازی به اجرای الگوریتم نیست و ربات وقتی در یک فاصله معین از هدف قرار گرفت، بدون اجرای الگوریتم، نقطه بعدی خود را نقطه‌ی هدف قرار داده و الگوریتم را پایان می‌بخشد.

۵- نتایج شبیه‌سازی

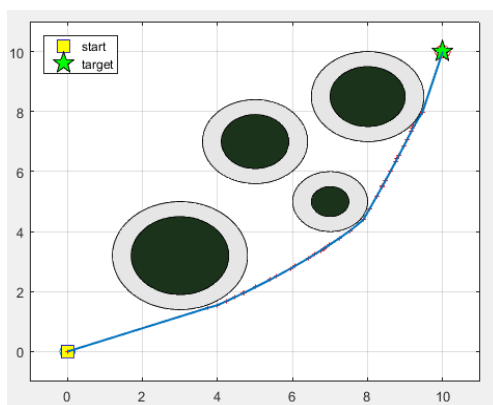
در این بخش، عملکرد الگوریتم پیشنهادی در محیط‌های مختلف با موانع گوناگون ارزیابی می‌شود. جهت شبیه‌سازی مسئله از برنامه متلب استفاده شده است. مطابق با چارچوب ارائه‌شده در [۲۴]، شعاع موانع به اندازه شعاع ربات (۵/۰ متر) گسترش یافته‌اند.

این نتایج نشان‌دهنده توانمندی الگوریتم در مدیریت چالش‌های پیچیده‌تر مسیریابی است و تأییدکننده بهبودهای قابل توجهی است که در مقایسه با آزمایش قبلی مشاهده شد. به‌ویژه، الگوریتم پیشنهادی همچنان قادر است تا مسیرهای بهینه‌تری را با کاهش چرخش‌های ناگهانی و افزایش ثبات در حرکت ربات ارائه دهد. این امر به بهبود کارایی و ایمنی ربات‌ها در محیط‌های پیچیده و با موانع متعدد کمک می‌کند، به‌ویژه در کاربردهای واقعی که شامل موانع استاتیک و متنوعی مانند سایت‌های تولید، انبارها و سایر محیط‌های صنعتی می‌شود.

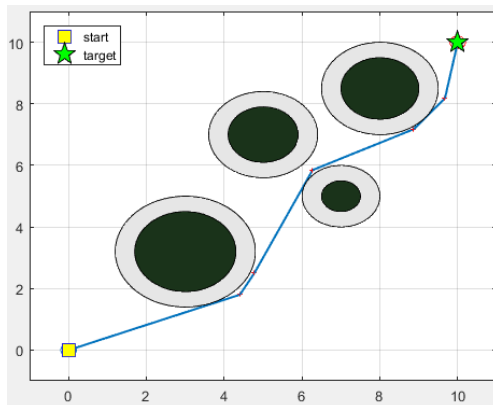
در پایان، با تغییر ناحیه کاری ربات و افزایش تعداد موانع به پنج عدد، به بررسی عملکرد روش پیشنهادی پرداخته شده است. جدول ۶ جزئیات موانع موجود در محیط با پنج مانع را ارائه می‌کند.

جدول ۴. مشخصات موانع محیطی با ۴ مانع

شماره مانع	مختصات مرکز	شعاع مانع
۱	[3,3.2]	1.8
۲	[5,7]	1.4
۳	[7,5]	1
۴	[8,8.5]	1.5

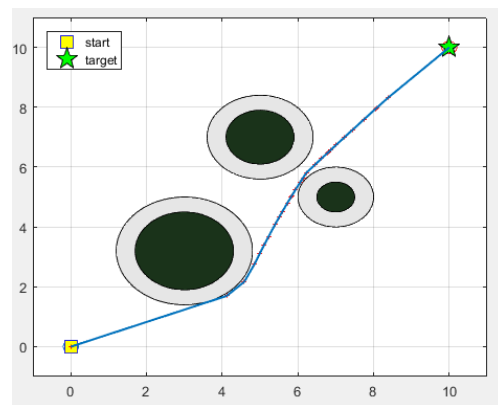


الف) روش پیشنهادی PSO-ABA

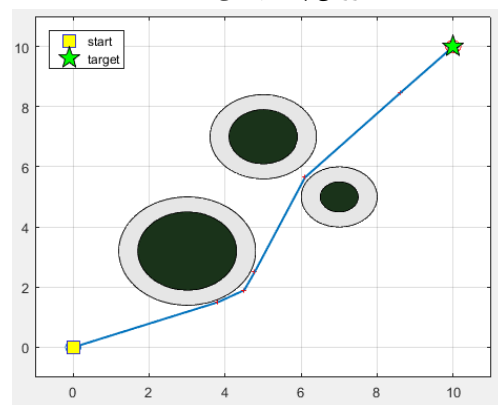


ب) روش PSO-MFB

شکل ۴. مسیر طراحی شده با ۴ مانع



الف) روش پیشنهادی PSO-ABA



ب) روش PSO-MFB

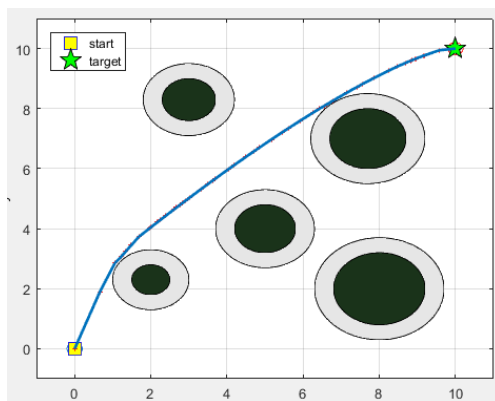
شکل ۳. مسیر طراحی شده با وجود ۳ مانع

جدول ۳. نتایج شبیه‌سازی

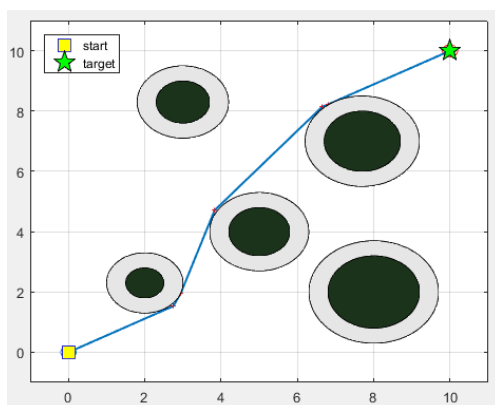
نام الگوریتم	طول مسیر	زاویه چرخش
PSO-MFB	14.8115	35.5766
PSO-ABA	14.9741	16.0853

همان‌طور که می‌دانیم، با افزایش تعداد موانع، پیچیدگی مسیریابی به‌طور قابل توجهی افزایش می‌یابد. از این‌رو، در شبیه‌سازی دیگری به تحلیل عملکرد الگوریتم پیشنهادی در محیطی با چهار مانع پرداخته می‌شود. مشخصات دقیق موانع در جدول ۴ ارائه شده است و نتایج حاصل از این شبیه‌سازی در شکل ۴ نمایش داده شده‌اند. همچنین، میزان طول مسیر و زاویه چرخش ربات در جدول ۵ آورده شده است.

مطابق با شکل ۴ و جدول ۵، مشاهده می‌گردد که مسیر طراحی شده توسط الگوریتم پیشنهادی صاف‌تر است و زاویه چرخش ربات نسبت به روش PSO-MFB، ۲۰/۵۱۱۸ درجه کمتر است. همچنین، مسیر طراحی شده توسط الگوریتم پیشنهادی، ۰/۱۶۲۶ متر کوتاه‌تر است. این نتایج گویای این است که با افزایش پیچیدگی مسیر، عملکرد روش PSO-MFB افت می‌کند.



الف) روش پیشنهادی PSO-ABA



ب) روش PSO-MFB

شکل ۵. مسیر طراحی شده با ۵ مانع

جدول ۸. نتایج مقایسه با سایر الگوریتم‌ها در حضور ۵ مانع

الگوریتم	طول مسیر
روش پیشنهادی PSO-ABA	14.6543
PSO-MFB	14.7785
MFB	14.7930
ABC	14.8821
DABC	14.7422

جهت بررسی عملکرد الگوریتم پیشنهادی در محیط‌های با تراکم موانع بالاتر، مسئله مسیریابی ربات در حضور تعداد قابل‌توجهی از موانع شبیه‌سازی شد. مشخصات موانع در جدول ۹ آورده شده است. نقطه شروع ربات در مختصات $[-6, -8]$ تعریف گردید و موقعیت اولیه تمام ذرات نیز بر همین نقطه متمرکز شد.

نتایج مقایسه‌ای مسئله مسیریابی در جدول ۱۰ گزارش شده است. این جدول شامل مقایسه عملکرد روش پیشنهادی با الگوریتم‌های IPSO, IGSA, IPSO-IGSA و PSO-MFB می‌باشد. همان‌طور که ملاحظه می‌شود، طول مسیر در روش پیشنهادی، نسبت به الگوریتم‌های ذکر شده، کمتر است. این نتایج نشان می‌دهد که رویکرد چندهدفه و اصلاح‌شده‌ی ما، در محیط‌های پیچیده با تعداد

جدول ۵. نتایج شبیه‌سازی

نام الگوریتم	طول مسیر	حداکثر زاویه چرخش
PSO-MFB	15.2681	39.6997
PSO-ABA	15.2087	19.1878

مطابق با شکل ۵ و جدول ۷، مشاهده می‌گردد که مسیر طراحی شده توسط الگوریتم پیشنهادی صاف‌تر است و زاویه چرخش ربات نسبت به روش PSO-MFB، $25/063$ درجه کمتر است. همچنین، مسیر طراحی شده توسط الگوریتم پیشنهادی، $0/1242$ متر کوتاه‌تر است.

جدول ۶. مشخصات موانع محیطی با ۵ مانع

شماره مانع	مختصات مرکز مانع	شعاع مانع
۱	(2, 2.3)	1
۲	(5, 4)	1.3
۳	(8, 2)	1.7
۴	(7.7, 7)	1.5
۵	(3, 8.3)	1.2

جدول ۷. نتایج شبیه‌سازی برنامه‌ریزی مسیر ربات در حضور ۵ مانع

نام الگوریتم	طول مسیر	حداکثر زاویه چرخش
PSO-MFB	14.7785	36.0724
PSO-ABA	14.6543	11.0094

۵-۲- مقایسه با سایر الگوریتم‌های فراابتکاری

در این بخش از شبیه‌سازی، عملکرد الگوریتم پیشنهادی با سایر الگوریتم‌های بهینه‌سازی مقایسه شده است. جدول ۸، نتایج مقایسه‌ای طول مسیر الگوریتم پیشنهادی را با سایر روش‌های ابتکاری از جمله PSO-MFB, MFB, ABC و DABC در حضور ۵ مانع (با مشخصات ذکر شده در جدول ۶) ارائه می‌دهد. این نتایج، توانمندی روش پیشنهادی در غلبه بر چالش‌های مسیریابی در محیطی دارای پنج مانع را تأیید می‌کند و برتری عملکردی آن را نسبت به روش‌های رقابتی در شرایط پیچیده نشان می‌دهد.

بر اساس داده‌های شبیه‌سازی، کوتاه‌ترین طول مسیر بدست‌آمده توسط الگوریتم پیشنهادی برابر با $14/6543$ متر است. در شرایط مشابه، این مقدار نسبت به الگوریتم‌های PSO-MFB, MFB, ABC و DABC کمتر بوده که نشان‌دهنده برتری و کارایی بالاتر الگوریتم پیشنهادی در یافتن مسیر بهینه می‌باشد.

مراجع

- [1] Tran, V.P., Perera, A., Garratt, M.A., Kasmari, K. and Anavatti, S.G., 2023. Coverage path planning with budget constraints for multiple unmanned ground vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(11), pp.12506-12522.
- [2] Anwit, R., Jana, P.K. and Obaidat, M.S., 2023. Obstacle adaptive smooth path planning for mobile data collector in the internet of things. *IEEE Transactions on Sustainable Computing*, 8(4), pp.727-738.
- [3] Mohammadi, Z., Soleimanpour-Moghadam, M., Abbasi moghadam, D., and Talebi, S., 2021. Maximizing the lifetime of underwater acoustic sensor networks by optimal relay node placement, 45(12), p.115-130.
- [4] Goerzen, C., Z. Kong, and B. Mettler, A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *Journal of Intelligent and Robotic Systems*, 2010. 57(1): p. 65-100.
- [5] Viet, H.H., S.H. An, and T.C. Chung, Dyna-Q-based vector direction for path planning problem of autonomous mobile robots in unknown environments. *Advanced Robotics*, 2013. 27(3): p. 159-173.
- [6] Zhang, H.-y., W.-m. Lin, and A.-x. Chen, Path planning for the mobile robot: A review. *Symmetry*, 2018. 10(10): p. 450.
- [7] Liang, W., Research on Robot Path Planning for Dynamic Environment and Cooperation. 2010, Zhejiang University Hangzhou, China.
- [8] Wu, Y., A survey on population-based meta-heuristic algorithms for motion planning of aircraft. *Swarm and Evolutionary Computation*, 2021. 62: p. 100844.
- [9] Ab Wahab, M.N., S. Nefti-Meziani, and A. Atyabi, A comparative review on mobile robot path planning: Classical or meta-heuristic methods? *Annual Reviews in Control*, 2020. 50: p. 233-252.
- [10] D. Karaboga, "An Idea based on Honey Bee Swarm for Numerical Optimization," Technical Report, Erciyes University, Engineering Faculty, Computer Engineering Department, pp. 1-10, 2005.
- [11] .B. Wu and C. Qian, "Differential Artificial Bee Colony Algorithm for Global Numerical Optimization," *Journal of Computer*, vol. 6, no. 5, pp.841-848 May 2011.
- [12] Abbas, N.H. and F.M. Ali, Path planning of an autonomous mobile robot using directed artificial bee colony algorithm. *International Journal of Computer Applications*, 2014. 96(11).
- [13] Ajeil, F.H., et al., Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. *Sensors*, 2020. 20(7): p. 1880.
- [14] Sierakowski, C.A. and Coelho, L.D.S., 2005, July. Study of two swarm intelligence techniques for path planning of mobile robots. In 16th IFAC world congress, Prague.
- [15] Sierakowski, C.A. and Coelho, L.D.S., 2006. Path planning optimization for mobile robots based on bacteria colony approach. In *Applied soft computing technologies: The challenge of complexity* (pp. 187-198). Springer Berlin Heidelberg.
- [16] Abdi, M.I.I., Khan, M.U., Güneş, A. and Mishra, D., 2020. Escaping local minima in path planning using a robust bacterial foraging algorithm. *Applied Sciences*, 10(21), p.7905.
- [17] Song, B., Z. Wang, and L. Zou, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-

موانع بالا، تعادل مناسبی بین معیارها برقرار کرده و کارایی قابل قبولی را ارائه می‌دهد.

علاوه بر این، بررسی‌ها حاکی از آن است که الگوریتم‌های ترکیبی (مانند روش پیشنهادی) با بهره‌گیری همزمان از مزایای دو الگوریتم پایه، توانسته‌اند محدودیت‌های روش‌های تک‌الگوریتمی را پوشش دهند و عملکرد بهتری را ارائه کنند.

جدول ۹. مشخصات موانع محیطی با ۸ مانع

شماره مانع	مختصات مرکز	شعاع مانع
۱	[3,3]	1.5
۲	[-3,0]	2
۳	[2,-2]	1.7
۴	[-2,4]	1
۵	[8,-2]	1
۶	[4,7]	1
۷	[-8,-2]	1
۸	[-2,-6]	0.7

جدول ۱۰. نتایج مقایسه با سایر الگوریتم‌ها در حضور ۸ مانع

الگوریتم	طول مسیر
روش پیشنهادی PSO-ABA	24.01
PSO-MFB	26.42
IGSA	33.29
IPSO	31.01
IPSO-IGSA	30.08

۶- نتیجه‌گیری

در این مقاله، یک چارچوب نوین برنامه‌ریزی مسیر مبتنی بر بهینه‌سازی چندهدفه ارائه شد که سه معیار کلیدی کمینه‌سازی طول مسیر، کاهش نرخ تغییرات زاویه‌ای و بیشینه‌سازی فاصله ایمن از موانع را به صورت همزمان مد نظر قرار گرفته است. با توجه به NP-سخت بودن مسئله، ابتدا از ترکیب روش اپسیلون محدودیت و رویکرد جمع وزنی برای تبدیل مسئله بهینه‌سازی چندهدفه به تک‌هدفه استفاده شد. در گام بعدی، یک الگوریتم فراابتکاری ترکیبی تطبیقی با عنوان PSO-ABA توسعه یافته است که از ادغام الگوریتم بهینه‌سازی ذرات (PSO) و الگوریتم خفاش وفقی (ABA) بهره می‌برد. ارزیابی عملکرد در محیط‌های شبیه‌سازی شده با چیدمان موانع پیچیده نشان داد که چارچوب پیشنهادی نسبت به روش‌های مرسوم دستیابی به تعادل بین اهداف مسئله برنامه‌ریزی مسیر ربات به دنبال دارد.

- [24] Ajeil, F.H., et al., Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Applied Soft Computing*, 2020. 89: p. 106076.
- [25] S. Agrawal, B. K. Patle, and S. Sanap, "A systematic review on metaheuristic approaches for autonomous path planning of unmanned aerial vehicles," *Drone Systems and Applications*, vol. 12, pp. 1–28, 2024.
- [26] P. K. Das, H. S. Behera, P. K. Jena, and B. K. Panigrahi, "An intelligent multi-robot path planning in a dynamic environment using improved gravitational search algorithm," *International Journal of Automation and Computing*, vol. 18, no. 6, pp. 1032–1044, 2021.
- [27] Sakcak, B. and LaValle, S.M., 2021, May. Complete path planning that simultaneously optimizes length and clearance. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 10100-10106). IEEE.
- [28] Yopez-Figueroa JJ, Victores JG, Oa ED, Balaguer C, Jardn A. Design and Development of an Omni directional Three-Wheeled Industrial Mobile Robot Platform. *Applied Sciences*. 2025 May 9;15(10):5277.
- [29] Bu S, Yan L, Gao X, Wang G, Zhao P, Chen IM. Design and motion control of spherical robot with built-in four-wheel omnidirectional mobile platform. *IEEE Transactions on Instrumentation and Measurement*. 2023 Apr 17;72:1-0.
- [30] Liu Z, Wang H, Wei H, Liu M, Liu YH. Prediction, planning, and coordination of thousand-warehousing robot networks with motion and communication uncertainties. *IEEE Transactions on Automation Science and Engineering*. 2020 Aug 19;18(4):1705-17.
- degree Bezier curve. *Applied Soft Computing*, 2021. 100: p. 106960.
- [18] Ibraheem, I.K. and F.H. Ajeil, Path planning of an autonomous mobile robot using swarm based optimization techniques. *Al-Khwarizmi Engineering Journal*, 2016. 12(4): p. 12-25.
- [19] Nezamabadi-pour H, Soleimanpour-moghadam M. Multi-robot Path Planning in a 3D Environment by Modified Particle Swarm Optimization Algorithm. *Journal of Iranian Association of Electrical and Electronics Engineers*. 2022 Jul 10;19(3):163-74.
- [20] P. K. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm and Evolutionary Computation*, vol. 28, pp. 14–28, 2016.
- [21] Yang, X.-S., Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 2011. 3(5): p. 267-274.
- [22] Saraswathi, M., G.B. Murali, and B. Deepak, Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm. *Procedia computer science*, 2018. 133: p. 510-517.
- [23] Ajeil, F.H., Ibraheem, I.K., Humaidi, A.J. and Khan, Z.H., 2021. A novel path planning algorithm for mobile robot in dynamic environments using modified bat swarm optimization. *The Journal of Engineering*, 2021(1), pp.37-48.