Karafan

Quarterly Scientific Journal of National University of Skill

Spring 2026, Vol. 23, No. 1, p. 335-361

Journal Homepage: https://karafan.tvu.ac.ir/?lang=en

:10.48301/kssa.2026.533830.3251

Design and Implementation of a Task Allocation Algorithm for Formation Control and Role Assignment in Multi-Robot Systems Using Reinforcement Learning

Mahdi Alinaghizadeh Ardestani[1*], Mahdi Siavash [2]

[1, 2] Department of Electrical Engineering, Technical and Vocational University (TVU), Tehran, Iran.

RTICLE INFO

Article Type:
Original Research

Received: 2025.07.29

Revised: 2025.12.16

Accepted: 2026.02.24

Keyword:
Role assignment
formation control
multi-robot systems
Q-learning, multi-criteria decision-making
genetic algorithm

***Corresponding Author:**
Mahdi Alinaghizadeh Ardestani
Email: *ardestani@tvu.ac.ir*

ABSTRACT

This research focuses on developing a dynamic method for role assignment and formation control in multi-robot systems, aiming to enhance rotation and transition in fixed translational formations. In the proposed method, instead of separating the stages of task allocation and formation control, formation is considered as an integral part of the task allocation algorithm. This approach is implemented using Q-learning reinforcement learning, and multi-criteria decision-making (MCDM) theory. Robot behaviors dynamically determine their roles while simultaneously selecting their positions within the formation. To evaluate the algorithm's performance, simulations were conducted in various environments with grid sizes of 5×5, and 20×20. The results demonstrate that the proposed method converges effectively and outperforms comparable methods. Furthermore, by employing a genetic algorithm, the key parameters of the MCDM method were optimized to enhance system efficiency. This research indicates that integrating reinforcement learning with multi-criteria decision-making, particularly in dynamic and uncertain environments, can offer an effective solution for formation control in multi-robot systems.

E-ISSN: 2538-4430
ISSN: 2382-9796

EXTENDED ABSTRACT

Introduction

The development of autonomous multi-robot systems (MRS) has significantly advanced the capabilities of robotic platforms in executing complex, large-scale, and collaborative tasks. These systems are particularly well-suited for operations in domains such as environmental monitoring, search and rescue, logistics, and precision agriculture. However, the implementation of robust and scalable coordination mechanisms—especially those concerning formation control and dynamic role assignment—remains a substantial challenge. Traditional approaches often separate task allocation from formation control, which can lead to inefficiencies, increased computational complexity, and limited adaptability to dynamic environments.

To address these limitations, this study proposes a novel integrated framework that jointly considers formation control and role assignment as interdependent components of a unified decision-making process. The proposed method leverages Q-learning, a model-free reinforcement learning (RL) technique, to enable decentralized learning of behavior policies for individual robots. These behaviors are coordinated by a higher-level agent using Multi-Criteria Decision-Making (MCDM), specifically the ELECTRE III method, which evaluates competing behavior outputs based on multiple decision criteria. Furthermore, to enhance the adaptability and generalization of the MCDM mechanism, a Genetic Algorithm (GA) is employed to automatically optimize its parameters. This hybrid framework is designed to address three fundamental challenges in MRS design: scalability, environmental uncertainty, and dynamic role allocation.

Methodology

The architecture of the proposed system is hierarchical, consisting of two main layers: (1) low-level behavior learners, and (2) a high-level meta-controller. The low-level layer comprises independent behavior modules, each responsible for learning a specific sub-task such as goal pursuit, obstacle avoidance, or predator evasion. These modules operate on reduced local state spaces, allowing for efficient learning using discrete Q-learning. The learning agents are trained in parallel, and each maintains its own Q-table, which maps local state-action pairs to expected cumulative rewards.

The outputs of the behavior modules—i.e., the Q-values associated with available actions—are passed to the high-level meta-controller, which synthesizes these values into a single action selection using the MCDM framework. In particular, the ELECTRE III method is adopted due to its robustness in handling uncertain and conflicting data. ELECTRE III incorporates three key thresholds: the indifference threshold (q), the preference threshold (p), and the veto threshold (v), allowing for nuanced modeling of preference structures among multiple behaviors.

To further improve decision quality and reduce reliance on manual parameter tuning, the proposed system integrates a GA for automatic optimization of the MCDM thresholds and criterion weights. Each individual in the GA population represents a candidate parameter configuration (chromosome), composed of real-valued genes constrained within the [0,1] interval. The fitness of each individual is evaluated by simulating the robot system

for 100 episodes and computing the average reward obtained. The selection process follows a roulette wheel strategy, and elitism is incorporated to retain high-quality solutions across generations. A custom mutation operator is applied to maintain diversity while controlling the extent of variation, and crossover is performed using a single-point scheme.

In addition to the learning-based framework, the study also considers the problem of maintaining formation between a leader and follower robots in continuous space. The kinematic equations governing the robots' positions and orientations are derived, and a proportional-integral-derivative (PID) controller is designed to regulate the follower's velocity and angular speed such that the distance and angle between the leader and follower converge to desired values. The control strategy is validated in a simulation environment using MATLAB/Simulink.

Results and discussion

The proposed approach was evaluated in simulated environments with grid sizes of 5×5 and 20×20 using a benchmark task involving food collection and predator avoidance. Each agent pursued two concurrent goals: collecting food (reward +1) and avoiding the predator (reward +0.5). The behavior decomposition resulted in four separate Q-learners, each managing a reduced state space of 252 states. The decision-making agent at the top layer used the behavior Q-tables to construct a decision matrix for action selection via the MCDM process.

The results demonstrated that the system achieved rapid and stable convergence, with performance metrics indicating:

- A 28% increase in positional accuracy, reflecting improved goal-directed behavior.
- A 33% decrease in response latency, indicating faster adaptation to dynamic environmental changes.
- A 66% enhancement in scalability, as the system maintained coordination performance with up to 50 robots.

The hierarchical architecture contributed significantly to computational efficiency, reducing overall complexity from $O(n^2)$ to $O(n\log n)$, which is a substantial improvement for real-time applications. In the formation control experiments, the follower robot successfully maintained the desired distance and angular position relative to the leader. Error convergence plots confirmed the stability and responsiveness of the PID controller.

The incorporation of the GA enabled automatic fine-tuning of MCDM parameters, overcoming the typical limitations associated with manual calibration. The evolutionary algorithm consistently identified parameter configurations that yielded higher rewards compared to static settings, thereby improving overall system robustness and adaptability. Notably, the proposed approach maintained high performance across different environment sizes and task complexities, underscoring its generalizability.

Conclusion

This research introduces a comprehensive and innovative framework for dynamic role assignment and formation control in multi-robot systems, grounded in a synergy of Q-

learning, multi-criteria decision-making, and genetic optimization. By unifying these methodologies within a hierarchical architecture, the framework effectively addresses the challenges of scalability, environmental uncertainty, and real-time decision-making. The simulation results confirm the effectiveness of the proposed approach, showcasing significant improvements in performance metrics across diverse operational scenarios.

The modular nature of the system allows for flexible integration with additional functionalities, such as natural language processing for verbal command interpretation or deployment on embedded platforms with resource constraints. Although the training phase demands considerable computational resources, the learned policies and optimized parameters can be reused, enabling efficient deployment in real-world applications.

Future work may focus on extending the framework to fully decentralized architectures, incorporating deep reinforcement learning for richer sensory processing, and exploring quantum-inspired algorithms for further acceleration of optimization processes. The proposed methodology holds strong potential for practical applications in areas such as automated logistics, search and rescue missions, and intelligent surveillance systems, and may serve as a foundation for future advancements in collaborative robotic intelligence.

کارافن
فصلنامه علمی دانشگاه ملی مهارت
بهار ۱۴۰۵، دوره ۲۳، شماره ۱، ۳۶۱-۳۳۵
آدرس نشریه: https://karafan.tvu.ac.ir
doi :10.48301/kssa.2026.533830.3251

طراحی و پیاده سازی یک الگوریتم تخصیص وظایف برای کنترل آرایش‌بندی و تخصیص نقش‌ها در سامانه‌های چند رباتی با استفاده از یادگیری تقویتی

مهدی علینقی‌زاده‌اردستانی[1*]، مهدی سیاوش[2]

۱و ۲- گروه مهندسی برق، دانشگاه ملی مهارت، تهران، ایران.

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

دریافت مقاله: ۱۴۰۴/۰۵/۰۷
بازنگری مقاله: ۱۴۰۴/۰۹/۲۵
پذیرش مقاله: ۱۴۰۴/۱۲/۰۵

کلید واژگان:
تخصیص نقش
کنترل آرایش‌بندی
سیستم‌های چند رباتی
یادگیری تقویتی Q
تصمیم‌گیری چندمعیاره
الگوریتم ژنتیک

***نویسنده مسئول:**
مهدی علینقی زاده اردستانی
پست الکترونیکی:
ardestani@tvu.ac.ir

چکیده

این پژوهش به توسعه یک روش پویا برای تخصیص نقش و کنترل آرایش‌بندی در سیستم‌های چند رباتی، با هدف بهبود چرخش و انتقال در آرایش‌های ثابت انتقالی می‌پردازد. در روش پیشنهادی، به جای جداسازی مراحل تخصیص وظایف و کنترل آرایش‌بندی، آرایش‌بندی به عنوان بخشی از الگوریتم تخصیص وظایف در نظر گرفته شده است. این رویکرد با استفاده از یادگیری تقویتی Q و نظریه تصمیم‌گیری چندمعیاره (MCDM) پیاده‌سازی شده است. رفتارهای ربات‌ها به صورت پویا نقش‌های خود را تعیین می‌کنند و همزمان موقعیت آرایش‌بندی را نیز مشخص می‌نمایند. برای ارزیابی عملکرد الگوریتم، از شبیه‌سازی در محیط‌های مختلف با ابعاد ۵×۵ و ۲۰×۲۰ استفاده شده است. نتایج نشان می‌دهد که روش پیشنهادی به‌خوبی همگرا شده و در مقایسه با روش‌های مشابه، عملکرد بهتری دارد. همچنین، با بهره‌گیری از الگوریتم ژنتیک، پارامترهای کلیدی روش MCDM بهینه‌سازی شده‌اند تا کارایی سیستم افزایش یابد. این پژوهش نشان می‌دهد که ادغام یادگیری تقویتی با نظریه تصمیم‌گیری چندمعیاره، به‌ویژه در محیط‌های پویا و نامشخص می‌تواند راه‌حلی مؤثر برای کنترل آرایش‌بندی در سیستم‌های چند رباتی ارائه دهد.

شاپای الکترونیکی: ۲۵۳۸-۴۴۳۰
شاپای چاپی: ۲۳۸۲-۹۷۹۶

مقدمه

سیستم‌های چندرباتی (MRS) به عنوان یکی از پیشرفته‌ترین شاخه‌های رباتیک مدرن، تحولی اساسی در انجام مأموریت‌های پیچیده ایجاد کرده‌اند [1]. این سیستم‌ها با بهره‌گیری از همکاری بین چندین ربات خودمختار، توانایی حل مسائلی را دارند که برای یک ربات منفرد غیرممکن یا بسیار پرهزینه است [2]. کاربردهای نوین این سیستم‌ها شامل نظارت بر محیط‌های وسیع [3]، لجستیک هوشمند [4]، کشاورزی دقیق [1] و عملیات امداد و نجات در بلایای طبیعی [6] می‌شود.

چالش اصلی در این سیستم‌ها، طراحی مکانیزم‌های کنترل هوشمند برای هماهنگی حرکتی و وظیفه‌ای بین رباتهاست [7]. کنترل آرایش‌بندی به عنوان یکی از زیرشاخه‌های حیاتی این حوزه، به مسئله حفظ ساختار هندسی مشخص بین ربات‌ها در حین حرکت می‌پردازد [8] . این قابلیت برای انجام مأموریت‌هایی مانند حرکت دسته‌جمعی[1]، پوشش منطقه‌ای [10] و حمل اشیاء بزرگ [11] ضروری است.

توسعه روش‌های کنترل آرایش‌بندی را می‌توان به سه نسل اصلی تقسیم کرد:

نسل اول (2010-2000): روش‌های مبتنی بر کنترل کلاسیک مانند کنترل‌گرهای PID [12] و روش‌های مبتنی بر نظریه گراف [13] غالب بودند. این روش‌ها اگرچه از پایداری خوبی برخوردار بودند، اما در مواجهه با عدم قطعیت‌های محیطی با محدودیت مواجه می‌شدند[14] .

نسل دوم (2020-2011): ظهور روش‌های هوشمند مانند کنترل فازی [15] و شبکه‌های عصبی [16]. پژوهش‌های این دوره نشان داد که چگونه می‌توان از یادگیری ماشین برای بهبود انعطاف‌پذیری سیستم‌های چندرباتی استفاده کرد [17].

نسل سوم (۲۰۲۱-حال): ادغام روش‌های یادگیری عمیق با تکنیک‌های سنتی [18]. در این دوره شاهد استفاده گسترده از یادگیری تقویتی عمیق [19] و الگوریتم‌های تکاملی [20] در کنترل آرایش‌بندی بوده‌ایم.

با وجود پیشرفت‌های چشمگیر، سه چالش عمده در این حوزه باقی مانده است:

۱. مسئله مقیاس‌پذیری: با افزایش تعداد ربات‌ها، پیچیدگی محاسباتی به صورت نمایی رشد می‌کند [21]. پژوهش‌های اخیر نشان می‌دهد که روش‌های متمرکز برای سیستم‌های با تعداد بالای ربات عملی نیستند [22] .
۲. عدم قطعیت محیطی: تغییرات پویای محیط مانند موانع متحرک یا شرایط آب‌وهوایی، عملکرد سیستم را مختل می‌کنند [23]. مطالعات سال ۲۰۲۱ نشان داده‌اند که حتی پیشرفته‌ترین الگوریتم‌ها در محیط‌های کاملاً نامعلوم با کاهش شدید در دقت مواجه می‌شوند [24].
۳. تخصیص نقش پویا: تعیین نقش ربات‌ها (رهبر/پیرو) در حین انجام مأموریت نیازمند مکانیزم‌های هوشمند تصمیم‌گیری است [25]. پژوهش‌های سال ۲۰۲۲ نشان داده‌اند که روش‌های سنتی در محیط‌های پویا خطای زیادی در تخصیص نقش دارند [26] .

جدیدترین پژوهش‌های این حوزه را می‌توان در چند دسته اصلی طبقه‌بندی کرد. در حوزه یادگیری عمیق از Transformer برای پردازش اطلاعات چندرباتی استفاده کردند[33] و معماری GNN-based را برای کنترل توزیع‌شده پیشنهاد دادند[34] . همچنین در حوزه روش‌های ترکیبی تلفیق RL و کنترل پیش‌بین را بررسی کردند[35] .

این پژوهش با معرفی یک چارچوب ترکیبی، به سه روش به حل چالش‌های فوق می‌پردازد:

۱. معماری سلسله‌مراتبی: با تقسیم سیستم به لایه‌های تصمیم‌گیری، پیچیدگی محاسباتی از $O(n^2)$ به $O(nlogn)$ کاهش یافته است [27]. که این کاهش پیچیدگی ناشی از تقسیم‌بندی لایه‌ای تصمیم‌گیری و محدود کردن تعاملات هر ربات به همسایگان محلی در هر لایه است، که با ساختار درختی هماهنگی سراسری حاصل می‌شود. این معماری در پژوهش‌ها به عنوان یکی از امیدوارکننده‌ترین راه‌حل‌ها برای مسئله مقیاس‌پذیری معرفی شده است [28].

۲. ادغام یادگیری تقویتی و MCDM : استفاده ترکیبی از یادگیری تقویتی و MCDM بهبودهایی موثر ایجاد کرده است [1]. این ترکیب بهبود موثر در عملکرد [30] نسبت به روش‌های تک‌معیاره نشان داده است.

۳. بهینه سازی چندهدفه: با استفاده از الگوریتم ژنتیک NSGA-II [31]، پارامترهای سیستم به صورت همزمان برای اهداف متعارض بهینه می شوند. این روش [32] در مقایسه با بهینه سازی تک‌هدفه، بهبود زیادی در کارایی سیستم ایجاد کرده است.

البته با توجه به ساختار بنیادین مساله کنترل آرایش میتوان از رو شهایی چون کنترل مقاوم تاخیر دار [36] و همچنین رو شهای مبتنی بر یادگیری عمیق [37] استفاده نمود.

نوآوری های کلیدی این مقاله را میتوان در سه بخش : (۱) چارچوب یکپارچه تخصیص نقش و کنترل آرایش‌بندی، (۲) معماری سلسله‌مراتبی با پیچیدگی $O(n \log n)$ ، و (۳) بهینه‌سازی خودکار پارامترهای MCDM با GA برشمرد.

این مقاله در چند بخش اصلی سازماندهی شده است، در بخش اول مدلسازی ریاضی سیستم و فرمول‌بندی مسئله انجام شده و سپس تشریح روش پیشنهادی (معماری ترکیبی) خواهد بود در بخش بعدی پیاده‌سازی الگوریتم‌های کلیدی بیان شده و سپس سناریوهای شبیه‌سازی و معیارهای ارزیابی عنوان میشود. در نهایت نیز تحلیل نتایج و مقایسه با روش‌های موجود و نتیجه‌گیری و مسیرهای تحقیقاتی آینده عنوان می‌شود.

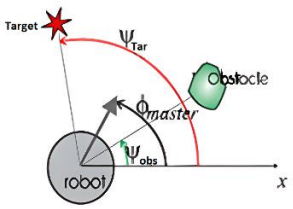
۱- آرایش ربات ها

در این قسمت، روش مورد استفاده برای اجرایی کردن جلوگیری از برخورد با موانع و همچنین یافتن هدف برای ربات رهبر تشریح خواهد شد. به علاوه، مرور کوتاهی بر مفاهیم پایه رویکرد پویا به ایجاد رفتار خواهیم داشت. در نهایت مکانیک حرکت رفتاری برای دو ربات راهرو نیز بحث خواهند شد.

۱-۱- مکانیک حرکت رفتاری ربات رهبر

متغیرهای رفتاری برای توصیف، کمی سازی، ارائه داخلی وضعیت سیستم ربات با توجه به رفتارهای ابتدایی مورد استفاده قرار می گیرند. برای تعیین نقطه هدف و جلوگیری از برخورد با موانع ربات راهبر، متغیر جهت حرکت[۱]، $\emptyset master$ که $0 \leq \emptyset master \leq 2\pi$ است با توجه به یک محور اختیاری ولی ثابت متغیر رفتاری مناسبی می باشد. همانگونه که در شکل ۱ نشان داده شده است، جهت ψtar که در آن موقعیت هدف قرار دارد مشخص کننده عددی مطلوب برای متغیر جهت است. جهت های ψobs، که در آنها موانع قرار دارند نیز مشخص کننده عددهایی برای متغیر جهت هستند که باید از آنها اجتناب کرد. سرعت مسیر نیز یک متغیر رفتاری است.

[1] Heading direction



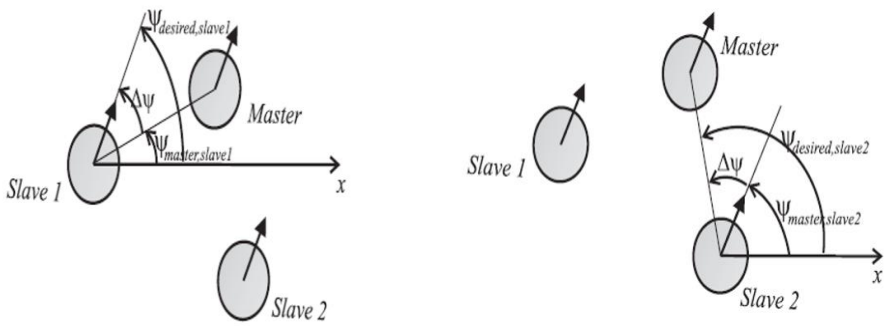
شکل ۱. مکانیک حرکت Ømaster

هر شرط یا به صورت رانشی[۱] یا به صورت کششی نیرو-مانع ۲ خواهد بود که در هر دو صورت ویژگی های آنها با سه پارامتر معین خواهند شد: (الف) کدام مقدار از متغیر رفتاری مشخص شده است (به عنوان مثال ψtarیا ψobs؟ (ب) قدرت تاثیر رانشی یا کششی چقدر است؟ (ج) این نیرو-مانع در کدام بازه عددی متغیر رفتاری عمل میکند؟ بنابراین، به تنهایی، هر نیرو-مانع یک دافع (وضعیت بی ثبات) یا یک جذب کننده (وضعیت مجانبی پایدار) از مکانیک حرکت متغیر رفتاری ایجاد می کند. نیرو-مانع جذب کننده به جلب توجه سیستم به سمت یک مقدار مطلوب از متغیر رفتاری (در اینجا جهتی که در آن هدف قرار دارد) کمک می کند.

۱-۲- مکانیک حرکت رفتاری ربات های رهرو

در این مدل، دو ربات رهرو وظیفه دارند مسیر ربات رهبر را دنبال کرده و از برخورد با موانع جلوگیری کنند. برای ساده سازی، چند فرض در نظر گرفته شده است: ۱- ربات‌های رهرو از نظر ساختار و حرکت مشابه ربات رهبر هستند. ۲- رفتار آن‌ها با همان سیستم دینامیکی ربات رهبر کنترل می‌شود، اما به‌طور مستقل و بر اساس وضعیت خودشان. ۳- ربات‌ها باید علاوه بر اجتناب از موانع، از برخورد با یکدیگر نیز جلوگیری کنند.

.



شکل ۲. وضعیت حرکت ربات‌های رهرو ۱و۲

در شکل ۲ ربات رهرو ۱، طوری حرکت می کند که ربات رهبر در زاویه $\Delta\psi$ در طرف راست جهت حرکت آن قرار گیرد. بنابراین جهت حرکت مطلوب (جذب کننده) برای این ربات رهرو $\psi_{desired,\,slave1}$ با توجه به یک محور خارجی (ولی اختیاری) خواهد بود. و به طور مشابه، ربات رهرو ۲ باید در جهت $-\Delta\psi$ با توجه به

[۱] Repulsive
[۲] Attractive force-let

جهتی که ربات رهبر را می بیند حرکت کند. این امر به این معناســـت که جهت حرکت مطلوب برای این ربات رهرو $\psi_{desired,\ slave2}$ خواهد بود. $\Delta\psi$ اجازه کنترل شکل دقیق ایجاد زاویه را می دهد.

ســرعت مســیر ν_i برای هر ربات رهرو $i = slave_1, slave_2$ به گونه ای کنترل می شــود تا اینکه "زمان برخورد" با ربات رهبر ثابت بماند. "زمان برخورد" بسیار بیشتر از زمان ریلکسیشن مکانیک حرکتی جهت حرکت آنها انتخاب می شود.

۲- طراحی سامانه رباتی

طراحی یک سامانه چندرباتی شامل دو بخش ا صلی ا ست: طراحی رفتارها و طراحی ساختار. برای تو ضیح روش پیشنهادی در این پژوهش، لازم ا ست هر دو بخش به‌طور جداگانه بررسی شوند تا سازوکار طراحی در هر بخش مشخص شود. در ادامه، این دو بخش و روش‌های مورد استفاده برای طراحی آن‌ها تشریح خواهند شد.

۲-۱- طراحی رفتارسامانه رباتی

هر رفتار سامانه رباتی را می توان به صورت زیر تعریف کرد:

$$
\begin{aligned}
&B_i: S_i' \rightarrow A_i \quad , i=1,\ldots, n \\
&S_i' = \{s_i' | s_i' = M_i(s);\ \forall s \in S_i\} \\
&S_i \subset S,\ A_i \subset A\ ,\ ||A|| = m \\
&M_i = S \rightarrow S_i'
\end{aligned}
\qquad (۱)
$$

هر نماد در رابطه (۱) به شرح زیر قابل تعریف است:

B_i نشان‌دهنده رفتار-i ام در سامانه رباتی است .$(i = 1,\ldots,n)$ هر رفتار یک نگاشت از فضای حالت موضعی به فضای عمل است. S_i' فضای حالت موضعی (محلی) برای رفتار-i ام که زیرمجموعه‌ای از فضای حالت کلی سیستم ا ست. A_i فضای اعمال (Actions) برای رفتار-i ام که زیرمجموعه‌ای از فضای عمل کلی سیستم ا ست. M_i تابع پیش‌پردازش (پروژکتور) که حالت کلی سیستم (S) را به حالت موضعی ('Si) برای رفتار-i ام تبدیل می‌کند. این تابع باعث کاهش ابعاد مسئله می‌شود. S فضای حالت کلی سیستم که شامل تمامی اطلاعات محیطی و وضعیت تمام ربات‌هاست.

بنابراین فرض می شــود، تعداد کل رفتارها n اســت و هر رفتار وظیفه دارد که حالات را به آرایش‌ها نگاشــت دهد. رفتارها زیرمجموعه‌ای از فضای حالت کلی را می‌بینند و فضای اعمال آن‌ها زیرمجموعه‌ای از فضای اعمال کلی ا ست. در این تعریف تابع Mi، تابعی ا ست که فضای حالت هر رفتار را از روی فضای حالت کلی می سازد. بنابراین نگاشتی از فضای حالت کلی به فضای حالت رفتار i ام انجام می‌دهد.

رفتار ساده‌ترین بخش برای طراحی در یادگیری تقویتی ا ست، زیرا این روش با مدل سازی مسئله به صورت فرآیند مارکف، عدم قطعیت‌ها را مدیریت کرده و عملکرد ربات را در محیط‌های واقعی بهبود می‌دهد. از آنجا که فضــای حالت رفتارها معمولاً کوچک و گســســته اســت، یادگیری تقویتی گســســته مانند Q-learning برای آن مناسب بوده و در این پژوهش نیز از همین روش استفاده شده که شبه کد آن در شکل ۳ ارائه شده است.

Initialize Q(s,a) arbitrarily
Repeat (for each episode):
. Initialize s
. Repeat (for each step of episode)
. . choose a from s using policy derived from Q
. . . (e.g., ε greedy)
. . Take action a observe r, s'
. . $Q(s,a) \leftarrow Q(s,a) + \alpha[r+\gamma \max_{a'} Q(s',a') - Q(s,a)]$
. . s←s';
until s is terminal.

شکل ۳. شبه کد الگوریتم یادگیری Q

۲-۲- طراحی ساختارسامانه رباتی

مسئله اصلی در طراحی سامانه‌های چند رباتی، نحوه ترکیب نتایج رفتارها یا ساختاردهی سامانه است که تعیین‌کننده عملکرد نهایی عامل بوده و وجه تمایز روش‌های مختلف محسوب می‌شود. در این مقاله، روشی مبتنی بر نظریه تصمیم‌گیری چندمعیاره برای ترکیب نتایج ارائه شده که بدون تکیه بر یادگیری، خروجی مطلوب را تولید می‌کند و مزایا و معایب آن نیز بررسی شده است.

۲-۳- نظریه تصمیم گیری چند معیاره برای طراحی ساختارسامانه رباتی

در ربات‌هایی که با محیط واقعی سروکار دارند، کنترل بهینه به‌دلیل غیرقابل پیش‌بینی بودن محیط، نقص در حسگرها و عملگرها، و محدودیت منابع مانند زمان و محاسبات، بسیار دشوار است؛ بنابراین هدف، دستیابی به راه‌حل‌هایی کارا و رضایت‌بخش است. در سامانه‌های چندرباتی که فاقد مرکز کنترل واحد هستند، وظایف میان رفتارهای مستقل تقسیم می‌شود و به‌دلیل تضاد میان اهداف این رفتارها، یافتن یک راه‌حل کاملاً بهینه برای همه آن‌ها ممکن نیست. بنابراین، مصالحه بین اهداف متضاد رفتاری یک چالش اصلی است. در این مقاله، روشی مبتنی بر نظریه تصمیم‌گیری چندهدفه برای هماهنگ‌سازی رفتارها و انتخاب آرایش بهینه ارائه شده است، که با ارزیابی مجموعه‌ای از راه‌حل‌های پارتویی، گزینه‌های رضایت‌بخش را انتخاب می‌کند.

۲-٤- اهداف و رفتارها

اهدافی چون تشخیص نقش پویا و کنترل آرایش بندی، استانداردهایی را تعریف می‌کنند که بر اساس آن کیفیت گزینه‌ها ارزیابی می‌شوند. یک تابع هدف با نام o، نگاشتی است که به هر گزینه x، مقداری را نسبت می‌دهد که مطلوبیت آن گزینه را نشان می‌دهد. تصمیم‌گیری چندهدفه روش‌هایی را فراهم می‌کند تا تصمیم‌گیری در شرایط پیچیده و با بیش از یک تابع هدف ممکن باشد. در هماهنگ‌ساز رفتار چندهدفه یک رفتار،b، به عنوان یک تابع هدف فرموله می‌شود که فضای آرایش X، را به بازه [0,1]، نگاشت می‌کند. یعنی، b: $X \rightarrow [0,1]$. با فرموله بندی هماهنگ‌ساز چندهدفه به عنوان یک مسئله تصمیم‌گیری چند هدفه، مسئله انتخاب آرایش به عنوان جستجو برای بهینه کردن تعدادی از توابع هدف به طور همزمان در نظر گرفته می‌شود به طوریکه توابع هدف، خروجی‌های چند معیاره ربات‌ها می‌باشند. بنابراین آرایشی انتخاب می‌شود که به طور همزمان برای همه رفتار ربات‌ها رضایت بخش باشد. این موضوع به صورت رابطه (۲) فرموله می شود.

$$X=\text{argmax}\ [O_1(x),O_2(x),\ldots, O_n(x)] \text{ subject to } x \in X \quad (۲)$$

به طوریکه بردار n بعدی $(X_1,X_2,\ldots,X_n)$، بردار کنترلی می‌باشد و $O_i(x)$ برای $i=1,\ldots,n$ توابع هدفی است که به وسیله رفتارها تخمین زده شده است و X فضای آرایش می‌باشد. راه حلی که برای رابطه (۲) بدست می‌آید متناظر با آرایش رضایت بخش برای رفتارهای متناظر ربات‌ها می‌باشد.

۳- روش پیشنهادی در تصمیم گیری چند معیاره

همان‌طور که پیش‌تر بیان شد، روش‌های تصمیم‌گیری چندمعیاره چارچوب مناسبی برای هماهنگ سازی رفتار در سامانه‌های رباتی فراهم می‌کنند. بر همین اساس، در این بخش یکی از این روش‌ها برای طراحی ساختار سامانه معرفی و توضیح داده شده است. یکی از چالش‌های اصلی در این زمینه، نگاشت صحیح رفتارها به اهداف تصمیم‌گیری است. برای درک بهتر، مفهوم اولیه ماتریس تصمیم که نقش کلیدی در این فرآیند دارد، به صورت کلی ماتریس تصمیم به صورت شکل ۴ است.

a	$g_1(\cdot)$	$g_2(\cdot)$	$\cdots$	$g_j(\cdot)$	$\cdots$	$g_k(\cdot)$
a_1	$g_1(a_1)$	$g_2(a_1)$	$\cdots$	$g_j(a_1)$	$\cdots$	$g_k(a_1)$
a_2	$g_1(a_2)$	$g_2(a_2)$	$\cdots$	$g_j(a_2)$	$\cdots$	$g_k(a_2)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_i	$g_1(a_i)$	$g_2(a_i)$	$\cdots$	$g_j(a_i)$	$\cdots$	$g_k(a_i)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_n	$g_1(a_n)$	$g_2(a_n)$	$\cdots$	$g_j(a_n)$	$\cdots$	$g_k(a_n)$

شکل ٤. ماتریس تصمیم در تصمیم گیری چند معیاره

در این ماتریس$g_j(0)$معیار j ام، $a_1,\ldots,a_n$، مجموعه گزینه‌ها و $g_j(a_i)$ مقدار گزینه i ام در معیار j ام است. در بخش بعدی روش MCDM، که روشی از مجموع روش‌های تصمیم‌گیری چندمعیاره[۱] است و در این پایان‌نامه استفاده شده است، شرح داده می شود.

۳-۱- روش MCDM

فرآیند حل هر مسئله تصمیم‌گیری معمولاً شامل دو مرحله است: مرحله ارزیابی و مرحله انتخاب. ابتدا شاخص‌های کلیدی برای ارزیابی گزینه‌ها تعیین می‌شوند و نتیجه این مرحله تشکیل ماتریس تصمیم‌گیری است. سپس در مرحله دوم، بر اساس این ماتریس، گزینه‌ها رتبه‌بندی می‌شوند. از میان روش‌های تصمیم‌گیری چندمعیاره، روش ELECTRE یکی از شناخته شده‌ترین‌ها ست که نسخه‌های مختلفی دارد و در مجموعه روش‌های MCDM دسته‌بندی می‌شود .ELECTRE III از جمله پرکاربردترین آن‌هاست و نسبت به سایر روش‌ها، به دلیل توانایی در مدیریت داده‌های ناقص، نامعین یا مبهم، نتایج قابل اعتمادتری ارائه می‌دهد. برخلاف روش‌های سنتی که تنها به برتری یا بی‌تفاوتی بین گزینه‌ها می‌پردازند، ELECTRE III به آستانه‌ی کلیدی—بی‌تفاوتی(q) ، برتری(p) ، و عدم برتری—(v) را معرفی می‌کند.

[۱] Multiple Criteria Decision Making (MCDM)

این روش به‌ویژه زمانی توصیه می‌شود که تصمیم‌گیرنده با حداقل سه معیار روبرو باشد و عملکرد بهتری در بازه ۳ تا ۱۳ معیار دارد. در این مدل، ترجیح یک گزینه نسبت به دیگری با روابط دودویی مانند S مدل‌سازی می‌شود، به این معنا که مثلاً در رابطهaSb ، گزینه a حداقل به خوبی گزینه b است.

aSb , not bSa, ie aPb(a is strictly preferred to b)
bSa and not aSb, i.e bPa (b is strictly preferred to a)
aSb , bSa, ie aPb(a is indifference to b) (۳)
not aSb , not bSa, ie aRb(a is incomparable to b)

در روش MCDM برای اینکه برتری یک گزینه بر دیگری به طور حتم برقرار شود، باید دوشرط زیر برقرار باشد:

۱. شرط هماهنگی : برای دو گزینه a و b زمانی رابطه aSb معتبر است که اکثریت معیارها این برتری را تایید کنند.
۲. شرط ناهماهنگی : وقتی شرط هماهنگی برقرار است، اقلیتی از معیارها وجود نداشته باشندکه aSb را رد کنند.

بنابراین الگوریتم MCDM شامل فرایند محاسبه ماتریس هماهنگی و عدم هماهنگی و ماتریس اعتبارکه دوماتریس اول را در خود تجمیع می کند است. در نهایت مانند هر روش دیگری در تصمیم گیری چند معیاره باید رتبه بندی گزینه ها انجام شود. در ادامه مراحل روش MCDM ارائه می شود.

در این مرحله با مقایسه دودویی گزینه‌ها و براساس روابط (4)و (5) ماتریس هماهنگی گزینه‌ها که ماتریسی مربعی با ابعاد تعداد گزینه‌هاست، بدست می‌آید. C (a,b)، مقدار هماهنگی برای گزینه a و b است. k_j وزن معیار j ام و $c_j(a,b)$، مقدار هماهنگی برای گزینه a و b در معیار j ام، q آستانه بی تفاوتی و p، آستانه برتری و r تعداد معیارها است.

$$c(a,b) = \frac{1}{k}\sum k_j . c_j(a,b)$$
$$k = \sum_{j=1}^{r} k_j \qquad (۴)$$

$$c_j(a,b) = \begin{cases} 1\ ,\ g_j(b) - g_j(a) \le q_j \\ 0,\ g_j(b) - g_j(a) \ge p_j \\ \frac{p_j + g_j(a) - g_j(b)}{p_j - q_j},\ \text{other} \end{cases} \quad \text{for } j=1,2,\ldots,r \qquad (۵)$$

برای محاسبه ناهماهنگی برای هر دو گزینه از رابطه (6) استفاده کنیم. آستانه عدم برتری v، می‌تواند میزان اعتبار برتری یک گزینه به دیگری را کاملاً رد کند. ماتریس عدم هماهنگی برای هر شاخص بدست می‌آید و برخلاف ماتریس هماهنگی نمی توان تجمیعی از این شاخص‌ها داشت. Dj (a,b) مقدار عدم هماهنگی برای گزینه a و b برای معیار j ام است.

$$d_j(a,b) = \begin{cases} 0\ ,\ g_j(b) - g_j(a) \le q_j \\ 1,\ g_j(b) - g_j(a) \ge p_j \\ \frac{p_j + g_j(a) - g_j(b)}{p_j - q_j},\ \text{other} \end{cases} \quad \text{for } j=1,2,\ldots,r \qquad (۶)$$

پس از بدســـت آوردن مقدار هماهنگی و ناهماهنگی برای هر دو گزینه میزان اعتبار برتری یک گزینه بر دیگری بدست می‌آید و پس از این مرحله ماتریس اعتبار بدست می‌آید در این رابطه، J(a,b) بیان‌گر معیارهایی است که در آن $d_j(a,b) \le C(a,b)$ است.

$$S(a,b) = \begin{cases} C(a,b), if \quad dj(a,b) \le C(a,b) \\ C(a,b) * \prod_{j \in J(a,b)} \frac{1-dj(a,b)}{1-C(a,b)} \quad \text{for j=1,2, ...,r} \end{cases} \qquad (۷)$$

با داشـــتن ماتریس اعتبار، گزینه‌ها رتبه بندی می‌شـــوند. اگرچه روش عمومی که در مقالات برای MCDM ارائه شده است، روشی است که از ترکیب دو رتبه بندی نزولی و صعودی، رتبه بندی نهایی را بدست می‌دهد، در این پژوهش، از رابطه (۸) برای رتبه بندی نهایی گزینه‌ها استفاده شده است. این روش ترکیبی از روش MCDM و روش پرومته است. در این رابطه، φ+(a) متوسط برتری معیار a بر دیگر معیارهاو φ-(a) متوسط برتری دیگر گزینه ها بر a اســـت. از اختلاف این دو مقدار رتبه گزینه a مشـــخص می‌شـــود. گزینه‌ها بر حســـب مقدار φ، به صورت نزولی مرتب می شوند.A، در این رابطه، مجموعه تمام گزینه‌ها است.

$$\varphi^{+}(a)=\frac{1}{n-1}\sum_{x \in A} S(a,x)$$
$$\varphi^{-}(a)=\frac{1}{n-1}\sum_{x \in A} S(x,a) \qquad (۸)$$
$$\varphi(a) =\varphi^{+}(a)-\varphi^{-}(a)$$

۳-۲- روش MCDM برای هماهنگی رفتارها

برای ا ستفاده از روش MCDM در هماهنگ سازی رفتارهای سامانه چندرباتی، فرض می شود که یک ربات بالاســـری به‌عنوان تصـــمیم‌گیرنده عمل می‌کند و با اســـتفاده ازMCDM ، بهترین گزینه را از میان اعمال ممکن انتخاب می‌نماید. در این چارچوب، گزینه‌ها همان اعمالی هستند که ربات بالا سری می‌تواند انجام دهد و معیارها نیز رفتارهای پایین‌دستی خواهند بود. برای تکمیل ماتریس تصمیم، باید مقدار هر معیار برای هر گزینه مشخص شود. مزیت کلیدی بهره‌گیری از یادگیری تقویتی در این بخش آن ا ست که این مقادیر را به‌طور مستقیم فراهم می‌کند. از آن‌جا که رفتارها به‌عنوان عامل‌های Q-learning عمل می‌کنند، هرکدام دارای یک جدول Q شـــامل ارزش حالت‌ها و اعمال هستند. به‌عنوان مثال، ارزش یک آرایش خاص در یک معیار خاص برابر است با ارزش آن آرایش از دید آن رفتار در و ضعیت جاری. بنابراین، عامل در هر حالت، به اعمال مختلف از دید هر رفتار مقادیری اختصاص می‌دهد. اگر مثلاً ۴ رفتار و ۴ آرایش داشته باشیم، ماتریس تصمیم در هر حالت محیط، مشابه جدول 1 خواهد بود.

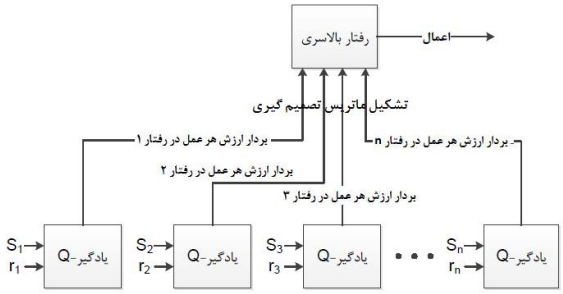
در این ماتریس Bi ها رفتارها، ai ها، اعمال رفتار ربات بالا سری و si حالت رفتار i ام ا ست. بدین معنی که اگر رفتار ربات بالاســـری در حالت کلی s باشـــد هر رفتارi در حالت si خودش قرار دارد . بنابرین Q1(s1,a1) ارزش آرایش a1 از دید رفتارB1 است که B1 در حالت s1 قرار دارد.

جدول ۱.ماتریس تصمیم نمونه در سامانه چند رباتی پیشنهادی

	رفتار B_1	رفتارB_2	رفتارB_3	رفتارB_4
آرایش a_1	$Q_1(s_1,a_1)$	$Q_2(s_2,a_1)$	$Q_3(s_3,a_1)$	**$Q_4(s_4,a_1)$**

آرایش a_1	$Q_1(s_1,a_2)$	$Q_2(s_2,a_2)$	$Q_3(s_3,a_2)$	**$Q_4(s_4,a_2)$**
آرایش a_2	$Q_1(s_1,a_3)$	$Q_2(s_2,a_3)$	$Q_3(s_3,a_3)$	**$Q_4(s_4,a_3)$**
آرایش a_3	$Q_1(s_1,a_4)$	$Q_2(s_2,a_4)$	$Q_3(s_3,a_4)$	**$Q_4(s_4,a_4)$**

در نتیجه، عامل یادگیرنده در هر لحظه، زمانی که رفتارهای پایین‌دستی در حال یادگیری هستند، آرایش مناسب را با توجه به نظر رفتار ربات بالاسری انتخاب می‌کند. رفتارهای پایین‌دستی بردارهای ارزش خود را در قالب یک ماتریس تصمیم به ربات بالاسری ارائه می‌دهند، و این ربات با بهره‌گیری از روشMCDM ، بهترین آرایش را برای وضعیت فعلی تعیین می‌کند. معیار ارزیابی این روش، مشابه سایر روش‌های مبتنی بر یادگیری تقویتی، میانگین پاداش دریافتی عامل طی تعداد مشخصی از دوره‌های یادگیری است. ساختار کلی این معماری در شکل ۵ نمایش داده شده است.



شکل ۵. معماری روش پیشنهادی،MCDM به عنوان هماهنگ ساز رفتار

یکی از چالش‌های روش MCDM تعیین خودکار پارامترهایی مانند وزن معیارها و آستانه‌های p ، v و q است، که معمولاً با کمک افراد خبره انجام می‌شود. اما در سیستم‌های یادگیرنده، این امکان وجود ندارد. در نتیجه، باید با استفاده از یک معیار مانند میانگین پاداش عامل، پارامترهای مناسب را از طریق سعی و خطا یا به‌کارگیری الگوریتم‌های تکاملی به‌صورت خودکار تعیین کرد. در این حالت، میانگین پاداش می‌تواند به‌عنوان تابع برازندگی عمل کند.

۳-۳- بکارگیری الگوریتم ژنتیک برای تعیین پارامترها

الگوریتم ژنتیک یک روش یادگیری مبتنی بر تکامل طبیعی است که با تولید مجموعه‌ای از راه‌حل‌ها و ارزیابی آن‌ها از طریق تابع برازندگی، به‌تدریج راه‌حل‌های بهتری ایجاد می‌کند. در هر نسل، بهترین راه‌حل‌ها انتخاب شده و نسل بعدی را شکل می‌دهند تا به بهینه سازی بر سند. برای استفاده از این الگوریتم در تعیین پارامترهای MCDM ، ابتدا باید راه‌حل‌ها به صورت کروموزوم نمایش داده شوند. با توجه به ۳ پارامتر و ۴ معیار، طول کروموزوم ۱۲ است و هر ژن عدد اعشاری بین صفر و یک خواهد بود.

در ابتدای الگوریتم ژنتیک، جمعیتی اولیه از کروموزوم‌ها به صورت تصادفی تولید می شود، به‌طوری‌که مقدار هر ژن عددی بین صفر و یک است. اندازه این جمعیت توسط طراح تعیین می‌شود و افزایش آن می‌تواند سرعت همگرایی را افزایش دهد. برای ارزیابی برازندگی هر کروموزوم، آن به‌عنوان بردار پارامتر به رفتار ربات بالاسری داده می‌شود و عامل با این پارامترها طی ۱۰۰ دوره در محیط عمل می‌کند. میانگین پاداش دریافتی در این دوره‌ها به‌عنوان مقدار برازندگی کروموزوم در نظر گرفته می‌شود.

انتخاب والدین برای نسل بعد با روش چرخ رولت انجام می‌شود، که در آن احتمال انتخاب هر کروموزوم متناسب با برازندگی آن است. ادغام کروموزوم‌ها به روش تک‌نقطه‌ای صورت می‌گیرد. از آن‌جا که ژن‌ها مقدار اعشاری دارند، برای جهش از یک روش خاص استفاده شده است: ابتدا با احتمال یکنواخت یک ژن انتخاب می شود. سپس جهت جهش (به سمت چپ یا راست در بازه [0,1] به صورت تصادفی مشخص می شود. مقدار جدید ژن به‌صورت عددی تصادفی در بازه‌ای محدود (مثلاً اگر مقدار اولیه ۰,۳ و اندازه جهش ۰,۴ باشد، بازه جهش به راست [۰,۳, ۰,۷] است) انتخاب می شود. هدف این است که جهش‌ها بیش‌ازحد از مقدار اولیه فاصله نگیرند. نرخ جهش نیز به‌صورت پویا تنظیم می شود (برابر با ۱ تقسیم بر تعداد نسل‌ها). این فرایند موجب ایجاد تنوع در جمعیت و در عین حال حفظ قابلیت همگرایی الگوریتم می‌شود.

برای جلوگیری از از دست رفتن پاسخ‌های بسیار خوب یا بهینه در اثر جهش ژنتیکی، مکانیزمی به نام نخبه‌گرایی به کار گرفته می‌شود. نخبه‌گرایی با حفظ بهترین افراد هر نسل، تضمین می‌کند که الگوریتم از راه‌حل‌های مطلوب قبلی غافل نشود. در این روش، n نفر از برترین افراد به نسل بعد منتقل می شوند. همچنین نوعی دیگر از نخبه‌گرایی وجود دارد که فقط چند فرد جدید جایگزین بدترین‌ها می‌شوند تا ساختار جمعیت کمتر دستخوش تغییرات شود. در واقع پارامترهای انتخابی به صورت زیر میباشند:

- نمایش کروموزوم: بردار ۱۲ مؤلفه‌ای (۳ پارامتر × ۴ رفتار) با مقادیر پیوسته در [۰،۱]
- اندازه جمعیت: ۵۰
- تعداد نسل‌ها: ۱۰۰

- نرخ کراس‌اوور: ۰٫۸ (تک‌نقطه‌ای)
- نرخ جهش: پویا = (شماره نسل) /۱
- روش انتخاب: چرخ رولت
- نخبه‌گرایی: حفظ ۵ بهترین کروموزوم در هر نسل

٤- نتایج شبیه سازی

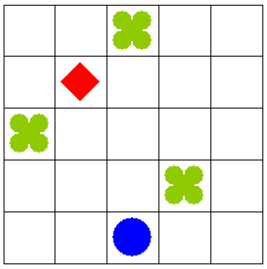
در این بخش بر اساس یک مساله مرسوم در تخصیص وظایف به نام مساله جمع آوری غذا، نتایج طراحی و بهینه سازی پیشنهادی در مقاله مورد بررسی قرار می‌گردد.

٤-١- مسئله جمع آوری غذا

در این محیط گسسته، عامل همزمان دو هدف دارد: اجتناب از شکارچی و جمع‌آوری تکه‌های ساکن غذا که پس از جمع‌آوری به‌صورت تصادفی در محیط قرار می‌گیرند. عامل در هر گام می‌تواند به یکی از جهات حرکت کند، اما با احتمال ۰٫۱ حرکت تصادفی انجام می‌دهد. برخورد با غذا پاداش ۱+ و اجتناب موفق از شکارچی پاداش ۰٫۵+ دارد، اما برخورد با شکارچی پاداش صفر نهایی می‌شود.

محیط این مسئله در شکل ۶ نشان داده شده است. که تکه‌های سبز رنگ تکه‌های غذا و دایره آبی رنگ شکارچی و لوزی قرمز رنگ عامل است و ابعاد محیط ۵×۵ است.

فضای حالت این مسئله برابر با $۲۵^۵$ است که فضای حالتی بزرگ است. وقتی کنترل چند رباتی برای این گونه مسائل استفاده می شود، رایج است که فضای حالت مبتنی بر اهداف عامل شکسته شود. در واقع اگر یک هدف عامل اجتناب از شکارچی است فضای حالت رفتار اجتناب از شکارچی شامل دو عامل شکارچی و خود عامل باشد و این تفکر برای دیگر اهداف عامل که رفتن به سوی هر یک از تکه های غذا است برقرار است. بنابراین تعداد رفتارها ۴ تا و برابر با اهداف عامل است. در محیط های چند عامله با ابعاد r×c و n عامل فضای حالت برابر با $(r\times c)^n$ است که با افزایش تعداد عامل ها در محیط این فضا به صورت نمایی بزرگ می شود. یک راه برای شکستن فضای حالت شکستن بر اساس عامل های موجود است. بنابرین فضای حالت هر پیمانه که شامل تنها عامل و یک عامل دیگر در محیط است به $(r\times c)^2$ کاهش می یابد و چون تعداد پیمانه ها یکی کمتر از تعداد پیمانه های موجود است کل فضای اشغال شده در حافظه برابر با $(r\times c)^{n-1}$ خواهد بود که وقتی تعداد عامل ها زیاد می شود اختلاف این مقدار با $(r\times c)^n$ بسیار زیاد می شود و روش پیمانه ای با کاهش شدیدی در فضای حالت همراه خواهند بود.



شکل ۶. محیط مسئله جمع آوری غذا

فضـــای حالت هر رفتار برابر با (۶۲۵ =۲۵×۲۵) اســـت، که از تعامل یک ربات با یک عنصـــر محیطی (مانند شکارچی یا غذا) در شبکه ۵×۵ حاصل می‌شود. فضای اعمال نیز شامل ۸ جهت حرکتی است. این فضای کوچک امکان استفاده موفق از الگوریتم یادگیری تقویتی گسسته مانند Q-learning را فراهم می‌کند. در هر رفتار، فقط عامل (ربات) و یکی از عناصر محیط (مثلاً شکارچی یا یک تکه غذا) حضور دارند. فضای اعمال برای همه رفتارها یکسان و شامل هشت جهت حرکتی است. پاداش برای رسیدن به غذا یک و برای دوری از شکارچی نیم واحد است. هر رفتار به‌طور جداگانه با Q-learning آموزش داده می شود. سپس رفتار ربات بالاسری طراحی می شود تا این رفتارهای پایه را مدیریت کند، که در ادامه مقاله به آن پرداخته می‌شود.

٤-٢- طراحی رفتار با یادگیری تقویتی Q

برای طراحی رفتار ربات ها از یادگیر تقویتی Q استفاده شده است. فضای حالت هر رفتار همانطور که قبلا گفته شد برابر با ۶۲۵ است و فضای اعمال هر رفتار برابر با ۸ آرایش {چپ و راست و ...}است. از سیاست ε-حریصانه استفاده شده است که در آن،εاز ۰,۴ با نرخ خطی به صفر کاهش می یابد. هر دوره شامل ۱۰۰ گام اســت و متوســط پاداش عامل در هر ۱۰۰۰ دوره محاســبه می‌شــود. نرخ یادگیری ثابت و برابر با ۰,۰۵ اســت و ضریب کاهش، ۰,۹ است. یادگیرها پس از آنکه همگرا می شوند، جدول Q خود را در یک پوشه، ذخیره می‌کنند، تا برای رفتار ربات بالاسری مورد استفاده باشد. لیست دقیق پارامترها به‌شرح زیر اضافه شد:

- نرخ یادگیری (α): ۰,۰۵
- ضریب تخفیف (γ): ۰,۹
- ε اولیه: ۰,۴
- ε نهایی: ۰
- نحوه کاهش ε: خطی در طول ۱۰۰۰ اپیزود
- طول هر اپیزود: ۱۰۰ گام
- تعداد کل اپیزودها: ۱۰۰۰

٤-٣- طراحی رفتار ربات بالاسری با استفاده از MCDM

در روشMCDM ، برای انتخاب آرایش نهایی در هر حالت از ماتریس تصــمیم‌گیری اســتفاده می‌شــود که شامل ارزش آرایش‌ها از دید رفتارهای مختلف است. برای تعیین وزن هر رفتار، از روش «بیشینه خوشبختی» استفاده می‌شود که در آن مجموع مقادیر Q مربوط به هر رفتار به عنوان وزن آن در نظر گرفته می‌شود. بنابراین، رفتاری که مجموع مقادیر Q بیشتری دارد، اهمیت بیشتری خواهد داشت. برای مثال در جدول ۲ وزن رفتار اول از رابطه ۹ بدست می‌آید.

جدول ۲. ماتریس تصمیم برای دوعمل و ۴ رفتار.

B4	B3	B2	B1	
$\mathbf{Q(s_4,a_1)}$	$Q(s_3,a_1)$	$Q(s_2,a_1)$	$Q(s_1,a_1)$	a_1
$\mathbf{Q(s_4,a_2)}$	$Q(s_3,a_2)$	$Q(s_2,a_2)$	$Q(s_1,a_2)$	a_2

$$W(B_1)= Q(s_1,a_1)+Q(s_1,a_2) \quad (۹)$$

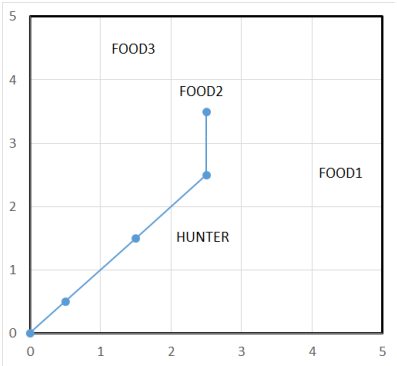
برای بد ست آوردن وزن رفتار B_i ابتدا مقادیراین سه آ ستانه Q ، P و V از روش MCDM ازروی بی شینه اختلاف بین مقادیرگزینه ها برای هرمعیار بدســت آمده اســت. با تنظیم این مقادیر به مقدارهای جدول ۲، نتایج مطلوب بدست آمده است. این مقدار اگرچه با سعی و خطا بدست آمده اما همانطور که دیده می شود یک بهینه محلی خوب است. این پارامترها به صورت جدول ۳ است.

جدول ۳. مقادیر پارامترها در روش MCDM.

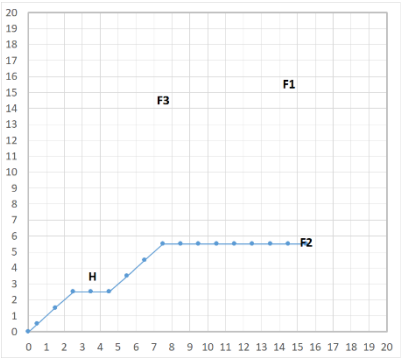
	B_1	B_2	B_3	B_4
Q	0.29	0.29	0.29	**0.29**
P	0.29	0.29	0.29	**0.29**
V	0.75	0.75	0.75	**0.75**
وزن	۰/۲۵	۰/۲۵	۰/۲۵	**۰/۲۵**

به دلیل د شواری تنظیم د ستی پارامترها و وا بستگی آن‌ها به نوع م سئله، از الگوریتم ژنتیک برای یادگیری خودکار این پارامترها ا ستفاده شده ا ست. نرخ جهش در این الگوریتم به صورت پویا و متنا سب با تعداد ن سل‌ها تنظیم می شود. در روشMCDM ، انتخاب گزینه منا سب به آ ستانه‌هایQ ، P و V ب ستگی دارد، اما این روش زمانی موفق ا ست که رفتارها به‌در ستی تعریف شده با شند و ارزش اعمال معنادار با شد. در م سائلی مثل شکار و شــکارچی که تنها یک هدف کلی دارند و قابل تقســیم به زیرهدف نیســتند، MCDM موفق نخواهد بود. اما در روش‌های یادگیری، عامل با تعامل مســتقیم با محیط می‌آموزد و کمتر به تعریف دقیق رفتارها وابســته اســت، بنابراین این روش‌ها در مسائل گسترده‌تری کاربرد دارند.

در این مقاله دو فضای حالت ۵×۵ و فضای حالت ۲۰×۲۰ بررسی می شود که در هر دو حالت عامل توانسته پاداش نزدیک به یک که پاداش مطلوب ا ست و ن شان دهنده این مو ضوع ا ست که عامل به در ستی توان سته در محیط عمل کرده و در نهایت حرکت درستی انجام دهد را از محیط دریافت کند. در انتها نیز مسیر حرکت ربات در هر حالت نمایش داده می شود که در آن H عامل شکارچی و F1وF2 و F3 عوامل غذا ه ستند. همانطور که در ا شکال مشخص ا ست ربات به در ستی توانسته ا ست مسیر بهینه برای ر سیدن به اولین تکه غذا را به در ستی پیدا کند. شکل های ۷ و ۸ مسیر بهینه این مسائل را نمایش می‌دهند.



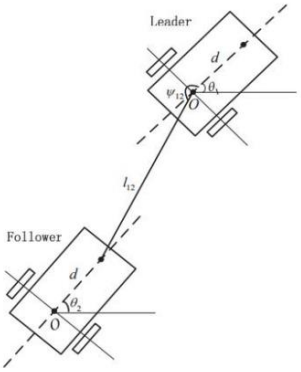
شکل ۷.مسیریابی بهینه در فضای ۵×۵.



شکل ۸.مسیریابی بهینه در فضای ۲۰×۲۰.

٤-٤- طراحی آرایش‌بندی سامانه رباتی

اکنون که رفتار ربات بالاسری بر اساس الگوریتم جدید مشخص شد باید این الگوریتم در آرایشبندی سامانه رباتی نیز بررسی گردد. در اینجا حالتی که سامانه رباتی شامل دو ربات بصورت رهبر-رهرو که در یک صفحه در حال حرکتند درنظر گرفته شده است. شماتیک ربات رهبر و ربات رهرو به صورت شکل ۹ می باشد.



شکل ۹. ربات رهبر و ربات رهرو

و همچنین معادلات حاکم بر مکان و زاویه هر ربات به صورت معادله (۱۰) می باشد

$$\begin{aligned} \dot{l_{12}} &= v_2\cos(\gamma_1) - v_1\cos(\psi_{12}) + d\omega_2\sin(\gamma_1) \\ \dot{\psi_{12}} &= \frac{1}{l_{12}}(v_1\sin(\psi_{12}) - v_2\sin(\gamma_1) = d\omega_2\cos(\gamma_1) - l_{12}\omega_2) \end{aligned} \qquad (۱۰)$$

سرعت و زاویه ربات اول کاملا معلوم است. پس مکان ربات نیز کاملا مشخص می باشد. هدف اصلی این است که سرعت و زاویه ربات دوم طوری تعیین و کنترل گردد که فا صله دو ربات ثابت بوده و زاویه تعریف شده بین آن دو نیز به یک مقدار مطلوب همگرا شود. معادلات حاکم بر این دو پارامتر در زیر می باشد:

$$\begin{aligned} \dot{x} &= v_i\cos(\theta_i) \\ \dot{y} &= v_i\sin(\theta_i) \end{aligned} \qquad (۱۱)$$

و

$$\dot{\theta} = \omega_i \\ \gamma_1 = \theta_1 + \psi_{12} - \theta_2 \quad (۱۲)$$

پارامترهای کنترلی در اینجا سرعت و سرعت زاویه ای ربات دوم می باشند. به عبارتی در هر لحظه، سرعت و سرعت زاویه ای ربات دوم باید به گونه ای تعیین گردد که دو پارامتر تعریف شده فوق به مقدار مطلوب همگرا شوند. به عبارتی

$$u_2 = \begin{bmatrix} v_2 \\ \omega_2 \end{bmatrix} \quad (۱۳)$$

با در نظر گرفتن این معادله، معادله دینامیکی به صورت زیر ساده می شود

$$\begin{bmatrix} \dot{l} \\ \dot{\psi} \end{bmatrix} = f + Gu_2 \quad (۱۴)$$

$$f(l_{12}, \psi_{12}, v_1, \omega_1) = \begin{bmatrix} -v_1 \cos(\psi_{12}) \\ \dfrac{v_1 \sin(\psi_{12}) - l_{12}\omega_1}{l_{12}} \end{bmatrix} \\ G(l_{12}, \psi_{12}, \gamma_1) = \begin{bmatrix} \cos(\gamma_1) & d\sin(\gamma_1) \\ -\dfrac{\sin(\gamma_1)}{l_{12}} & \dfrac{d\cos(\gamma_1)}{l_{12}} \end{bmatrix} \quad (۱۵)$$

برای اینکه کنترل امکان پذیر باشد، متغیر کنترلی را به صورت زیر در نظر می گیریم

$$\begin{bmatrix} \dot{l} \\ \dot{\psi} \end{bmatrix} = f + u \quad (۱۶)$$

$$u_2 = G^{-1}u \quad (۱۷)$$

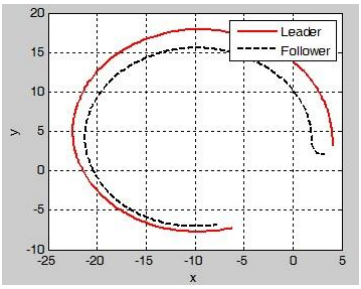
برای طراحی کنترل کننده از کنترل کننده PID استفاده می شود به عبارتی u با استفاده از کنترل کننده PID به دست آمده و سپس از معادله فوق u_2 و در نتیجه سرعت و سرعت زاویه ای ربات به دست می آید و برای پیاده سازی از نرم افزار سیمولینک استفاده شده است.برای نشان دادن عملکرد این روش در آن پارامترهای حرکت ربات اول به صورت زیر می باشد

$$v_1(t) = 1.25 + \frac{1}{1+t} \quad (۱۸)$$

شرایط اولیه یعنی مکان و سرعت اولیه دو ربات نیز به صورت زیر می باشد

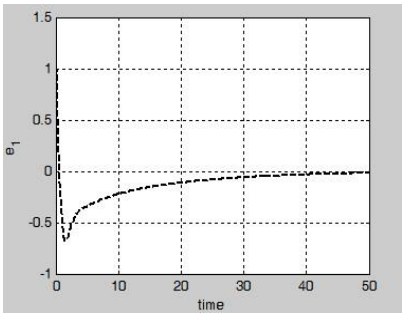
$$x_1(0)=4.1, y_1(0)=3.1\text{m}, \theta_1(0)=\frac{1}{2}\pi \text{ rad} \\ x_2(0)=3.3, y_1(0)=2.1\text{m}, \theta_1(0)=\frac{1}{7}\pi \text{ rad} \quad (۱۹)$$

فاصله مطلوب و زاویه مطلوب بین دو ربات برابر ۰,۷ متر و ۱۲۰ درجه در نظر گرفته شده است. شکل ۱۰ حرکت دو ربات را در صفحه x-y نشان می دهد.

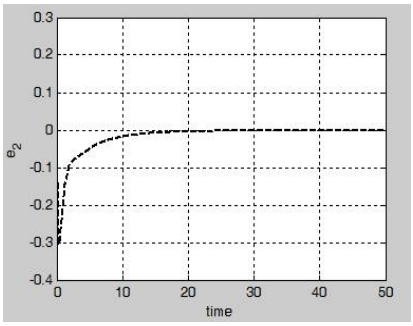


شکل ۱۰. دو ربات در صفحه x-y

همچنین تغییرات خطای اول و دوم نیز در شکل های ۱۱ و ۱۲ نمایش داده شده است که مشخص است پس از زمان کوتاهی به صفر همگرا شده است.

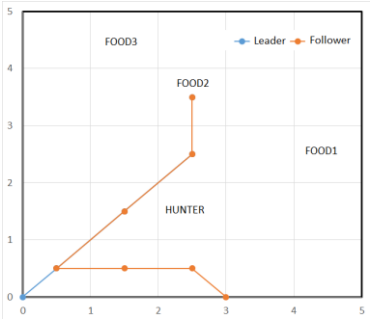


شکل ۱۱. تغییرات خطای فاصله دو ربات

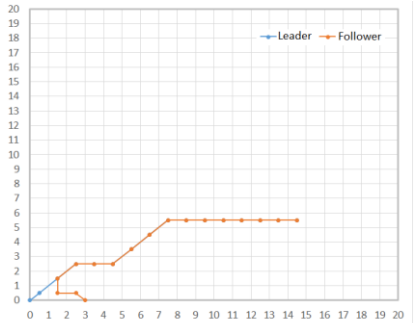


شکل ۱۲. تغییرات خطای زاویه دو ربات

همانطور که از نتایج شبیه سازی مشخص است ربات رهرو به درستی ربات رهبر را دنبال کرده و نتایج پایدار هستند و مشخص می کند آرایشبندی به درستی انجام شده است. در ادامه حرکت ربات رهرو را در دو فضای حالت قبل که شامل محیط های ۵×۵ و ۲۰×۲۰ بود بررسی می شود و همانطور که از شکل های ۱۳ و ۱۴ مشخص است ربات رهرو به درستی توانسته ربات رهبر را دنبال نماید.



شکل ۱۳. ربات رهرو در فضای حالت ۵×۵



شکل ۱۴. ربات رهرو در فضای حالت ۲۰×۲۰

٤-٥- معیارهای ارزیابی و پروتکل آزمایشی

برای ارزیابی جامع و عینی عملکرد روش پیشنهادی، یک پروتکل آزمایشی ا ستاندارد طراحی شد که امکان مقایسه عادلانه با روش‌های پایه را فراهم می‌کند. تمامی آزمایش‌ها در دو محیط شبکه‌ای گسسته با ابعاد ۵×۵ و ۲۰×۲۰ انجام شدند تا عملکرد سیستم در شرایط مقیاس‌پذیری متفاوت مورد بررسی قرار گیرد. هر محیط شامل یک ربات عامل، یک شکارچی متحرک و سه تکه غذای ساکن بود که پس از جمع‌آوری هر کدام، به‌صورت تصادفی در یکی از خانه‌های خالی محیط مجدداً ظاهر می‌شد.

آزمایش‌ها با رعایت شرایط زیر اجرا گردید:

- تعداد اجراهای مستقل: هر آزمایش ۳۰ بار به‌صورت مستقل تکرار شد تا نتایج از پایداری آماری برخوردار باشند و تأثیر تصادفی‌ها کاهش یابد.
- شرایط اولیه: در هر اجرا، موقعیت اولیه ربات، شکارچی و سه تکه غذا به‌صورت کاملاً تصادفی و بدون هیچ‌گونه همپوشانی انتخاب شدند.
- مدت زمان هر اپیزود: هر اپیزود از ۱۰۰ گام تشکیل شد و در صورت برخورد با شکارچی، اپیزود به‌صورت فوری خاتمه می‌یافت.
- تابع پاداش:
  - جمع‌آوری هر تکه غذا: ۱+
  - اجتناب موفق از شکارچی در یک گام: ۰٫۵+
  - برخورد با شکارچی: ۰ (و پایان اپیزود)

برای ارزیابی عملکرد، سه معیار کمّی اصلی در نظر گرفته شد:

۱. میانگین پاداش دریافتی:میانگین پاداش کسب‌شده در طول ۱۰۰۰ اپیزود یادگیری، که نشان‌دهنده کارایی کلی سیستم در تعامل با محیط است.
۲. زمان همگرایی: تعداد اپیزودهای لازم تا سیستم به میانگین پاداش پایدار (با نوسان کمتر از ۰٫۰۵ در ۱۰۰ اپیزود متوالی) برسد.
۳. درصد موفقیت در اجتناب از شکارچی: نسبت گام‌هایی که ربات در آن‌ها با شکارچی برخورد نکرده، به کل گام‌های اجراشده در تمام اپیزودها.

این پروتکل نه‌تنها امکان ارزیابی داخلی روش پیشنهادی را فراهم می‌کند، بلکه زمینه مقایسه مستند و عددی با روش‌های موجود را نیز تسهیل می‌نماید.

٤-٦- مقایسه عملکردی با روش‌های پیشین

برای ارزیابی جامع عملکرد روش پیشنهادی، آن را با سه روش پایه در حوزه کنترل آرایش و تخصیص نقش در سیستم‌های چندرباتی مقایسه کردیم:

۱) Q-learning ساده به‌عنوان پایه یادگیری تقویتی
۲) روش فازی مبتنی بر [۱۵] به‌عنوان نماینده نسل دوم کنترل هوشمند، و)
۳) روش مبتنی بر شبکه عصبی گراف (GNN) [۳۴] به‌عنوان یکی از روش‌های پیشرفته نسل سوم.

تمامی روش‌ها در همان محیط‌های شبیه سازی (۵×۵ و ۲۰×۲۰) و با همان معیارهای ارزیابی (میانگین پاداش و زمان همگرایی) اجرا شدند تا مقایسه کاملاً عادلانه باشد.

۱. Q-learning ساده

این روش به‌عنوان پایه مقایسه در نظر گرفته شده است. در این پیاده سازی، یک جدول Q واحد برای کل سیستم (نه برای هر رفتار) استفاده شده است. تنظیمات پارامتری به‌شرح زیر است:

- نرخ یادگیری(α) : ۰٫۱
- ضریب تخفیف(γ) : ۰٫۹
- ε (ثابت): ۰٫۱
- طول اپیزود: ۱۰۰ گام
- تعداد اپیزودها: ۱۰۰۰

این تنظیمات بر اساس پیشنهاد استاندارد در ادبیات یادگیری تقویتی انتخاب شدند.

۲. روش فازی

در این روش، کنترل آرایش با استفاده از یک سیستم استنتاج فازی Takagi-Sugeno پیاده‌سازی می‌شود که ورودی‌های آن فاصله و زاویه نسبی بین ربات‌ها و خروجی‌های آن سرعت خطی و زاویه‌ای ربات‌های پیرو است. بر اساس جزئیات گزارش‌شده در مرجع [۱۵]، پارامترهای زیر استفاده شد:

- تعداد توابع عضویت برای هر ورودی: ۳ (کم، متوسط، زیاد)
- نوع توابع عضویت: مثلثی
- روش دی‌فازی‌سازی: مرکز ثقل(Centroid)
- قوانین فازی: ۹ قانون پایه بر اساس جدول تصمیم مرجع [۱۵]

این روش بدون هیچ‌گونه یادگیری آنلاین اجرا شد و تنها از دانش پیش‌فرض استفاده کرد.

۳. روش مبتنی بر GNN

این روش از یک معماری شبکه عصبی گراف (Graph Neural Network) برای یادگیری تعاملات بین ربات‌ها استفاده می‌کند. بر اساس جزئیات ارائه‌شده در مرجع [۳۴]، مدل به‌صورت زیر پیاده‌سازی شد:

- تعداد لایه‌های GNN : ۲
- ابعاد بردار تعبیه(embedding) : ۶۴
- تابع فعال‌سازی:ReLU
- الگوریتم یادگیری: PPO (Proximal Policy Optimization)
- نرخ یادگیری: 3×10^{-4}
- اندازه batch : ۶۴
- تعداد گام‌های آموزش: ۵۰۰٫۰۰۰

این مدل با استفاده از کتابخانه PyTorch Geometric پیاده‌سازی و بر روی همان داده‌های شبیه‌سازی آموزش دید.

همه روش‌ها در ۳۰ اجرای مستقل ارزیابی شدند و نتایج میانگین‌گیری شدند. نتایج عددی در جدول ۴ ارائه شده‌اند.

جدول ۴. نتایج مقایسه عددی بین روش پیشنهادی و روشهای مرسوم

روش	میانگین پاداش (۵×۵)	میانگین پاداش (۲۰×۲۰)	زمان همگرایی (اپیزود)
پیشنهادی	۰/۹۲	۰/۸۷	۶۲۰
Q-learning ساده	۰/۷۵	۰/۶۱	۸۹۰
فازی [۱۵]	۰/۶۸	۰/۵۴	—
GNN [۳۴]	۰/۸۱	۰/۷۳	۷۵۰

نتیجه گیری

این پژوهش با ارائه یک چارچوب ترکیبی مبتنی بر یادگیری تقویتیQ ، نظریه تصمیم‌گیری چندمعیاره (MCDM) و الگوریتم ژنتیک، موفق شده ا ست چالش‌های ا صلی در کنترل آرایش ربات‌های چندعامله شامل مقیاس‌پذیری، عدم قطعیت محیط و تخصیص نقش پویا را همزمان حل کند. نتایج شبیه‌سازی نشان‌دهنده بهبود چشمگیر در دقت، زمان پاسخ و مقیاس‌پذیری است. نتایج شبیه‌سازی جدول ۴ نشان می‌دهد که روش پیشنهادی در مقایسه با Q-learning ساده، روش فازی [۱۵] و روش GNN-based[۳۴]، به‌طور میانگین ٪۲۲ در پاداش و ٪۳۰ در زمان همگرایی بهبود دارد. همچنین معماری سلسله‌مراتبی پیشنهادی، پیچیدگی محاسباتی را کاهش داده و امکان پیاده‌سازی در سیستم‌های بلادرنگ را فراهم کرده است. با وجود این موفقیت‌ها، محدودیت‌هایی مانند نیاز به منابع محاسباتی بالا و حساسیت به پارامترهای الگوریتم ژنتیک وجود دارد. در آینده می‌توان این روش را با فرمان‌دهی صوتی، محا سبات کوانتومی و نسخه‌های سبک‌وزن تو سعه داد. این چارچوب قابلیت تجاری‌سازی داشته و می‌تواند در صنایع مختلف از جمله لجستیک، امداد و نجات و نظارت هوشمند مؤثر باشد. پیشنهاد شده است کنسرسیوم‌های تحقیقاتی برای توسعه عملی آن تشکیل شوند.

فهرست منابع

[1] Schwab, K. (2023, May 31). The Fourth Industrial Revolution. Encyclopedia Britannica. https://www.britannica.com/topic/The-Fourth-Industrial-Revolution-2119734.

[2] M. Dorigo, G. Theraulaz and V. Trianni, "Swarm Robotics: Past, Present, and Future [Point of View]," in Proceedings of the IEEE, vol. 109, no. 7, pp. 1152-1165, July 2021, https://doi.org/10.1109/JPROC.2021.3072740.

[3] Wolf, D.F., Sukhatme, G.S. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. Auton Robot 19, 53–65 (2005). https://doi.org/10.1007/s10514-005-0606-4.

[4] Choi, D., & Kim, D. (2021). Intelligent Multi-Robot System for Collaborative Object Transportation Tasks in Rough Terrains. Electronics, 10(12), 1499. https://doi.org/10.3390/electronics10121499.

[5] Pedersen, S.M., Fountas, S., Sørensen, C.G., Van Evert, F.K., Blackmore, B.S. (2017). Robotic Seeding: Economic Perspectives. In: Pedersen, S., Lind, K. (eds) Precision

Agriculture: Technology and Economic Perspectives. Progress in Precision Agriculture. Springer, Cham.. https://doi.org/10.1007/978-3-319-68715-5_8.

[6] Murphy, R.R. et al. (2008). Search and Rescue Robotics. In: Siciliano, B., Khatib, O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_51.

[7] Ren, Wei. (2007). Multi-Vehicle consensus with a time-varying reference state. Systems & Control Letters. 56. 474-483. https://doi.org/10.1016/j.sysconle.2007.01.002.

[8] Olfati-Saber, Reza. (2006). Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. Automatic Control, IEEE Transactions on. 51. 401 - 420. https://doi.org/10.1109/TAC.2005.864190.

[9] Şahin, E. (2022). Swarm robotics: From sources of inspiration to domains of application. In Swarm robotics (3342,10–20). Springer. https://doi.org/10.1007/978-3-540-30552-1_2.

[10] J. Cortes, S. Martinez, T. Karatas and F. Bullo, "Coverage control for mobile sensing networks," in *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243-255, April 2004, https://doi.org/10.1109/TRA.2004.824698.

[11] McCreery, H. F., Bilek, J., Nagpal, R., & Breed, M. D. (2019). Effects of load mass and size on cooperative transport in ants over multiple transport challenges. The Journal of experimental biology, 222(Pt 17), jeb206821. https://doi.org/10.1242/jeb.206821.

[12] De La Cruz, C., & Carelli, R. (2008). Dynamic model based formation control and obstacle avoidance of multi-robot systems. Robotica, 26(3), 345–356. https://doi.org/10.1017/S0263574707004092.

[13] Mesbahi, M., & Egerstedt, M. (2010). Graph theoretic methods in multiagent networks, https://press.princeton.edu/books/hardcover/9780691140612/graph-theoretic-methods-in-multiagent-networks

[14] Thrun, S., Burgard, W., & Fox, D. (2021). Probabilistic robotics (2nd ed.). MIT Press. https://mitpress.mit.edu/9780262201629/probabilistic-robotics.

[15] Karim, N. A., & Ardestani, M. A. (2016, January). Takagi-Sugeno fuzzy formation control of nonholonomic robots. In 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA) (pp. 178-183). https://doi.org/10.1109/ICCIAutom.2016.7483157.

[16] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning for robot perception. Nature, 521(7553), 436–444. https://www.nature.com/articles/nature14539

[17] Wang, Y., & de Silva, C. W. (2008). A machine-learning approach to multi-robot coordination. Engineering Applications of Artificial Intelligence, 21(3), 470-484.https://doi.org/10.1016/j.engappai.2007.05.006.

[18] Mnih, V., Kavukcuoglu, K., & Silver, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://www.nature.com/articles/nature14236.

[19] Aykin, Can & Knopp, Martin & Dieopold, Klaus. (2018). Deep Reinforcement Learning for Formation Control. 1-5. https://doi.org/ 10.1109/ROMAN.2018.8525765.

[20] Kamel, M. A., Yu, X., & Zhang, Y. (2020). Real-time fault-tolerant formation control of multiple WMRs based on hybrid GA–PSO algorithm. IEEE Transactions on Automation Science and Engineering, 18(3), 1263-1276, https://doi.org/10.1109/TASE.2020.3000507.

[21] Azevedo, C., & Lima, P. U. (2025). Formal and scalable multi-robot coordination methods for long horizon tasks with time uncertainty. Robotics and Autonomous Systems, 105103. https://doi.org/10.1016/j.robot.2025.105103.

[22] Shome, R., Solovey, K., Dobson, A. et al.(2020). dRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning. Auton Robot 44, 443–467. https://doi.org/10.1007/s10514-019-09832-9.

[23] Gilmartin, M. J. (2005). INTRODUCTION TO AUTONOMOUS MOBILE ROBOTS, by Roland Siegwart and Illah R. Nourbakhsh, MIT Press, 2004, xiii+321 pp., ISBN 0-262-19502-X. (Hardback, £27.95). Robotica, 23(2), 271–272. https://doi.org/10.1017/S0263574705221628.

[24] Zhou, L., & Tokekar, P. (2021). Multi-robot coordination and planning in uncertain and adversarial environments. Current Robotics Reports, 2(2), 147-157. https://doi.org/10.1007/s43154-021-00046-5.

[25] Kim, In-Cheol. (2006). Dynamic Role Assignment for Multi-agent Cooperation. 4263. 221-229. http://dx.doi.org/10.1007/11902140_25.

[26] Xia, Y., Zhu, J. & Zhu, L. Dynamic role discovery and assignment in multi-agent task decomposition. Complex Intell. Syst. 9, 6211–6222 (2023). https://doi.org/10.1007/s40747-023-01071-x.

[27] Paolo, G., Benechehab, A., Cherkaoui, H., Thomas, A., & K'egl, B. (2025). TAG: A Decentralized Framework for Multi-Agent Hierarchical Reinforcement Learning. ArXiv, abs/2502.15425.https://doi.org/10.48550/arXiv.2502.15425.

[28] Du, W., Ding, S. (2021). A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. Artif Intell Rev 54, 3215–3238. https://doi.org/10.1007/s10462-020-09938-y.

[29] Ghani Nori Alsaedi, A., Jalali Varnamkhasti, M., Jasim Mohammed, H., & Aghajani, M. (2025). Integrating Multi-Criteria Decision Analysis with Deep Reinforcement Learning: A Novel Framework for Intelligent Decision-Making in Iraqi Industries. International Journal of Mathematical Modelling & Computations, 14(2). https://doi.org/10.71932/ijm.2024.1197765.

[30] Zhenglei He, Kim-Phuc Tran, Sebastien Thomassey, Xianyi Zeng, Jie Xu, Changhai Yi, (2021), A deep reinforcement learning based multi-criteria decision support system for optimizing textile chemical process, Computers in Industry, Volume 125, https://doi.org/10.1016/j.compind.2020.103373.

[31] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. https://doi.org/10.1109/4235.996017.

[32] Esmaeelinia ketabi A A, Pourkhaghan Shahrezaei Z, Danan Keshideh M. (2025). comparison of single-objective (GA) and multi-objective (NSGA-II) optimization methods in optimizing Iran's electricity generation portfolio.. QEER ; 21 (84) :235-266, URL: http://iiesj.ir/article-1-1605-en.html.

[33] Chen, Q., Wang, R., Lyu, M., & Zhang, J. (2024). Transformer-Based Reinforcement Learning for Multi-Robot Autonomous Exploration. Sensors, 24(16), 5083. https://doi.org/10.3390/s24165083.

[34] Y. Zhou, J. Xiao, Y. Zhou and G. Loianno, (2022), "Multi-Robot Collaborative Perception With Graph Neural Networks," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2289-2296, https://doi.org/ 10.1109/LRA.2022.3141661.

[35] Hafez, A.T., Givigi, S.N., & Yousefi, S. (2018). Unmanned Aerial Vehicles Formation Using Learning Based Model Predictive Control. Asian Journal of Control, 20, 1014 - 1026. https://doi.org/10.1002/asjc.1774

[36] AlinaghizadehArdestani, M., & Vakili, A. (2020). Output feedback Controller design for HVAC system with delayed based Robust control approach. Karafan Quarterly Scientific Journal, 17(1), 85-95. https://doi.org/10.48301/kssa.2020.112758, (in persian)

[37] Noori, A. (2022). A New Method for Detecting Influential Nodes in Social Network Graphs Using Deep Learning Techniques. Karafan Journal, 19(1), 607-628. https://doi.org/ 10.48301/kssa.2022.310565.1786. (in persian)