SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Self-Evolving Agentic Logic Algorithms for Dynamic Multi-Goal Navigation

S. Seethaladevi[1*], R. Jeevajothi[2], Dr. Priya Vij[3], Shailendra Narayan Singh[4]

1*Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: seethaladevi@maher.ac.in

[2]Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: rjeevajothimba@maher.ac.in

[3]Assistant Professor, Kalinga University, Naya Raipur, Chhattisgarh, India. E-mail: ku.priyavij@kalingauniversity.ac.in, https://orcid.org/0009-0005-4629-3413

[4]Professor, Department of Computer Science and Engineering, Graphic Era Deemed University, Dehradun, Uttarakhand, India. E-mail: sns2033@gmail.com, https://orcid.org/0000-0002-1761-5648

*Corresponding author: Email: seethaladevi@maher.ac.in

Abstract

Effective multi-goal navigation in autonomous systems represents one of the key challenges faced by researchers today. In such scenarios, the agents need to successfully navigate towards achieving multiple goals within their environments that have moving obstacles. Traditional methods like path-planning techniques and reinforcement learning techniques often fail when it comes to providing real-time adaptive capabilities while achieving multiple goals. In this study, we propose a novel algorithm called Self-Evolving Agentic Logic (SEAL), which uses techniques of goal prioritization, agentic reasoning, and self-evolution to allow autonomous agents to adopt effective strategies for successful multi-goal navigation. SEAL has been tested in a virtual 2D environment that has dynamic and static obstacles with random placement of multiple goals. Metrics used were Success Rate (SR), Path Efficiency (PE), Computational Cost (CC), and Goal Completion Time (GCT). The proposed algorithm was benchmarked against existing algorithms such as A*, Deep Q-Networks (DQN), and conventional Agentic Logic. The SEAL algorithm demonstrated the highest success rate (96.5%), path efficiency (92.3%), and goal completion time (18.2 s), while still ensuring relatively low computational costs (12.4 ms per decision). Comparison studies showed significant statistical improvement over all baselines ($p<0.05$), which indicated improved adaptability and robustness to changes in dynamic multi-goal environments. The SEAL algorithm is a powerful and flexible tool for multi-goal navigation. It can be easily scaled to larger problems due to its modularity and self-evolutionary approach. Future research may consider the application of SEAL to multi-agent scenarios, robotics, and reinforcement learning frameworks.

Keywords: Self-evolving agentic logic, multi-goal navigation, adaptive decision-making, autonomous agents, path efficiency, success rate, simulation environment

1. Introduction

Dynamic Multi-Goal Navigation is one of the most vital problems that need to be solved by Artificial Intelligence (AI) and robotics technology because of the need for self-directed entities to function effectively in complicated and ever-changing surroundings [1][2]. Dynamic navigation and path-planning tasks can involve various applications, including mobile robots, drones, and smart transportation systems, among others, all of which require timely decisions and adjustments [3][4]. Classical methods used for navigation purposes, like A* search or Dijkstra's algorithm or decision-making systems using pre-set rules, may fail under dynamic conditions owing to the incapability to adapt to changing circumstances, address multiple goals at the same time, and learn from changing situations [5][6].

Self-evolving agentic logic presents a potential solution through which an agent can modify its decision-making approach according to the feedback received from the environment and varying goals [11]. Through the incorporation of adaptive thinking, prioritization of goals, and self-reprogramming capability, the algorithms permit agents to optimize their performance within a complex environment involving multiple goals [13]. This study proposes a new model for the development of algorithms for self-evolving agentic logic algorithms.

The main goals of this study are (i) the development of an agentic logic algorithm that is able to self-evolve based on the changes in its environment, (ii) assessment of its effectiveness in situations with multiple goals, and (iii) demonstration of its superior performance compared to other navigation algorithms. The key contributions of this study will be the creation of a self-evolving logic system, the assessment process, and practical considerations.

The paper is organized in the following manner: Section II outlines the relevant background literature related to multigoal navigation, adaptive planning, and agentic logic, and identifies the knowledge gaps in this area. Section III discusses the proposed algorithm for self-evolving agentic logic, including the methodology used, approach, and adaptive features. Section IV elaborates on the experiment setting, criteria of assessment, and outcomes of the analysis conducted, together with a comparative study of the results achieved. Section V covers the discussion, practical significance, limitations, and future perspectives. Section VI concludes and suggest future directions.

2. Literature Review

The problem of multi-goal navigation has been researched extensively within the context of Artificial Intelligence and robotics, concentrating on ways of helping autonomous agents achieve several target destinations effectively in changing environments [7][8]. Common methods of achieving multi-goal navigation include the usage of graph-based approaches such as Dijkstra's algorithm and A* Search, as well as optimization methods such as RRT and PRM [9]. Even though these methods are quite efficient in static or partially dynamic environments, their limitations appear when dealing with highly dynamic scenarios.

Adaptive and self-learning methods, such as reinforcement learning (RL), DQNs (deep Q networks), and evolutionary methods, have shown considerable success in improving the performance of multi-goal navigation by learning optimal strategies through interactions with the environment [12]. Additionally, there is growing interest in agentic logical systems, where the integration of reasoning and decision-making allows the agent to alter its goals and strategies dynamically according to the emerging situation [14]. However, despite offering a systematic approach for integrating autonomous decision making and self-evolution within an agent, applications of these methods have been restricted [15][17].

Through comparative analysis, it becomes evident that regular algorithms perform efficiently computationally but are not adaptable, while self-learning and agentic reasoning algorithms provide adaptability but are costly computationally and are prone to difficulties in converging [12][16]. There is an urgent requirement for algorithms that incorporate the best features of both approaches, allowing for dynamic goal-directed navigation through multiple objectives [10][18].

Previous techniques are predominantly restricted to single-goal navigation or cannot adapt themselves to multiple goals simultaneously[19]. There is very little prior research conducted in the area of combining self-evolving logic with navigation strategies involving more than one goal[20]. Furthermore, an absence of evaluation metrics that compare the performance of adaptive agentic logic with that of classical techniques motivates the development of the presented technique.

3. Proposed Self-Evolving Agentic Logic Algorithm

System Architecture and Framework

Figure 1: Overall Methodology of the Self-Evolving Agentic Logic Algorithm

Figure 1 presents a diagrammatic representation of the entire process of the suggested self-evolving agentic logic-based approach to dynamic multiple goal navigation. This approach starts with the Perception Module, whose purpose is to collect real-time data about the environment, detect any obstacles, and update the internal world models of the agent. In the second step, Goal Management is carried out, where active goals are identified based on their attributes. Next is Goal Prioritization, where dynamic priorities for the goals are determined taking into account contextual elements, the urgency of the situation, and the associated utilities of the actions involved. Adaptive decision-making follows, where an appropriate action is chosen using adaptive logic rules. The Action Execution Module executes the action, while Self-Evolution Module uses the feedback received in execution to learn continuously until all goals are met.

This proposed framework provides an approach for autonomous agents to traverse through the environment with multiple goals through the incorporation of perception, reasoning, and self-evolution. This is achieved through the Perception Module, which keeps track of environmental dynamics and obstacles and updates the internal map of the agent. The Agentic Logic Module maintains representations of the several goals and selects actions through

dynamic prioritization and adaptive reasoning. The Self-Evolution Module updates the decision rules and action selection policies through iterations. By employing all the three components, the agents will be able to dynamically alter their strategies to meet multiple goals in the dynamic and uncertain environment. Also, it could easily be incorporated with reinforcement learning.

Adaptive Goal Prioritisation and Decision-Making

Goal selection is carried out based on an assessment of each current goal in terms of relevance, importance, proximity, and dependency on other goals, thereby allowing the agent to allocate its resources efficiently. The decision-making module identifies possible courses of action for the agent by employing an adaptive utility function, which takes into account the probability of success, the expected results, and the level of uncertainties in the environment. Decisions are made with the aim of maximizing overall utility despite the presence of conflicting goals and possible dangers. By doing so, the agent is able to alternate between its goals or strategies seamlessly in real-time, ensuring faster goal attainment and better success chances in multiple-goal navigation.

Dynamic Goal Prioritization

$$\pi_i = \alpha R_i + \beta U_i + \gamma \frac{1}{D_i + \epsilon} + \delta V_i \qquad (1)$$

Equation (1) computes the priority score π_i for each goal g_i, considering:

- Relevance (R_i)
- Urgency (U_i)
- Distance (D_i)
- Expected utility (V_i)

The weights $\alpha, \beta, \gamma, \delta$ balance the contribution of each factor. This is central to deciding which goal the agent should focus on first in dynamic environments.

Adaptive Action Selection

$$a^* = \arg \max_{a_j \in A} \left[Q(a_j) + r(a_j) - \lambda \rho(a_j) \right] \qquad (2)$$

Equation (2) selects the optimal action $a^*$from the set of candidate actions Abased on:

- Expected outcome $Q(a_j)$
- Immediate reward $r(a_j)$
- Risk or cost $\rho(a_j)$
- Risk-sensitivity factor λ

It captures the adaptive decision-making process, combining learned experience, current goals, and environmental feedback.

Self-Evolution Mechanism

The self-evolution component constantly updates the logic of decision-making within the agent to ensure efficiency and effectiveness. Information related to the outcomes of the actions performed previously by the agent, whether these were successful attempts to achieve the subgoals set for the agent and any unforeseen situations in the environment, is used to modify the priorities and the utility functions and select better action selection policies. Through multiple cycles of interaction with the environment, the agent acquires optimal strategies for performing various tasks efficiently and effectively. These strategies maximize not only the efficiency of the calculations but also the optimality of the paths followed.

Pseudocode and Workflow Representation

Algorithm 1: Self-Evolving Agentic Logic for Multi-Goal Navigation
Input: Environment state E, set of goals G, initial agent state S_0
Output: Optimal navigation path P
1. Initialize agent state $S \leftarrow S_0$
2. Sense environment state E, including dynamic obstacles, changes, and available goals
3. While goals in G remain:
   a. Prioritise goals based on relevance, urgency, distance, dependencies, and expected utility
   b. Generate candidate actions for each goal using the agentic logic reasoning module
   c. Evaluate each candidate action using an adaptive utility function that incorporates past feedback, predicted outcomes, and risk assessment
   d. Execute the action $a$with the highest utility score
   e. Update agent state $S$and refine the self-evolution module by incorporating feedback from execution results, including goal achievement, obstacles encountered, and environmental changes
4. End while
5. Return the optimal path Pcovering all goals

Algorithm 1 can be regarded as a feedback adaptive system for dynamic multi-goal navigation. It involves initialization of the agent and perception of the environment, where the agent senses environmental objects, goals, and any modifications to the current state and builds up its internal model. The agent then determines priorities of each goal based on their urgency, importance, dependencies, and utility. The agentic logic part of the agent comes up with possible actions to achieve particular goals. Actions are evaluated with respect to their predicted outcomes according to adaptive utility that takes into account risks and rewards associated with those outcomes. The actions performed are then passed to the self-evolution module.

Complexity, Scalability, and Adaptability Analysis

The algorithm runs in polynomial-time complexity during each decision process, making it feasible to implement in practice when applied in robotics and other AI applications. The modularity in its design provides scalability by ensuring that multiple goals can be tackled at once without causing excessive load in terms of computational power. The adaptability of the system comes from its ability to evolve constantly based on new conditions, unexpected challenges, or even a shift in priorities regarding goals. The method stands out in its robustness, adaptability, and efficiency compared to traditional planning or pathfinding algorithms used for similar purposes. Furthermore, it can be easily integrated with other AI solutions like reinforcement learning models and neural planners.

4. Experimental Setup and Results

Simulation Environment

The self-evolving agentic logic algorithm, suggested as an innovation, was tested in a dynamic environment simulating a realistic situation where multiple objectives are needed to be achieved. The dynamic environment comprises a two-dimensional space containing multiple mobile and stationary obstacles, dynamic goals, and complex terrains. Random generation of goal locations and other parameters occurs in every run of the simulation to test the performance of the algorithm in different circumstances. The objective is that the agent visits all the goals without colliding and at minimum traversal cost.

Hardware and Software Configuration

All simulations were conducted on a machine having the following specifications: Intel Core i7-12700H CPU, 32GB RAM, NVIDIA RTX 3080 GPU. The algorithm was coded using Python 3.10, with support from NumPy, Matplotlib,

and custom simulation modules. The following were some of the parameter values used within the algorithm: learning rate, η=0.05, risk sensitivity, λ=0.3, and maximum iterations per goal = 50. The experiments were conducted for 50 repetitions.

Evaluation Metrics

Algorithm performance was quantitatively assessed using the following metrics:

1. Success Rate (SR):

Measures the percentage of goals successfully reached by the agent.

$$\mathrm{SR} = \frac{N_{\text{achieved}}}{N_{\text{total}}} \times 100 \qquad (3)$$

Where in equation (3):

- Nachieved= Number of goals reached
- Ntotal= Total number of goals

2. Path Efficiency (PE):

Evaluates how close the agent's actual path length is to the optimal path length.

$$\mathrm{PE} = \frac{\sum_{i=1}^{N} L_{\text{optimal},i}}{\sum_{i=1}^{N} L_{\text{actual},i}} \times 100 \qquad (4)$$

Where in equation (4):

- $L_{\text{optimal},i}$= Optimal path length to goal i
- $L_{\text{actual},i}$= Actual path length taken by the agent to goal i

3. Computational Cost (CC):

Average processing time per decision step, indicating efficiency of computation.

$$\mathrm{CC} = \frac{\sum_{t=1}^{T} \tau_t}{T} \qquad (5)$$

Where in equation (5):

- τ_t= Time (in ms) taken for decision at step t
- T= Total number of decision steps

4. Goal Completion Time (GCT):

Measures the average time taken to achieve all goals in a trial.

$$\mathrm{GCT} = \frac{\sum_{i=1}^{N} t_i}{N} \qquad (6)$$

Where in equation (6):

- t_i= Time taken to reach goal i
- N= Total number of goals

Comparative Results

Table 1: Comparative Performance of the Proposed SEAL Algorithm and Baseline Methods

Method	SR (%)	PE (%)	CC (ms)	GCT (s)
Proposed SEAL	96.5	92.3	12.4	18.2
A* Fixed	82.0	78.5	8.5	27.6
DQN	89.4	85.7	24.1	21.3
Agentic Logic	91.2	87.0	15.2	20.1

Table 1 shows a quantitative analysis of the SEAl algorithm compared with existing navigation techniques, namely A* Search with Fixed Priority, DQN, and conventional Agentic Logic, not involving self-evolving. In this study, success rate (SR), measuring the number of achieved goals expressed as percentages; path efficiency (PE), denoting the similarity between actual and optimal paths; computational cost (CC), measuring the average amount of processing time consumed per one decision step; and goal completion time (GCT), measuring the average time to finish all goals have been considered as evaluation metrics. As a result, it can be seen that the SEAL algorithm is capable of yielding the best performance in terms of success rate and path efficiency but with minimal computational cost and GCT.

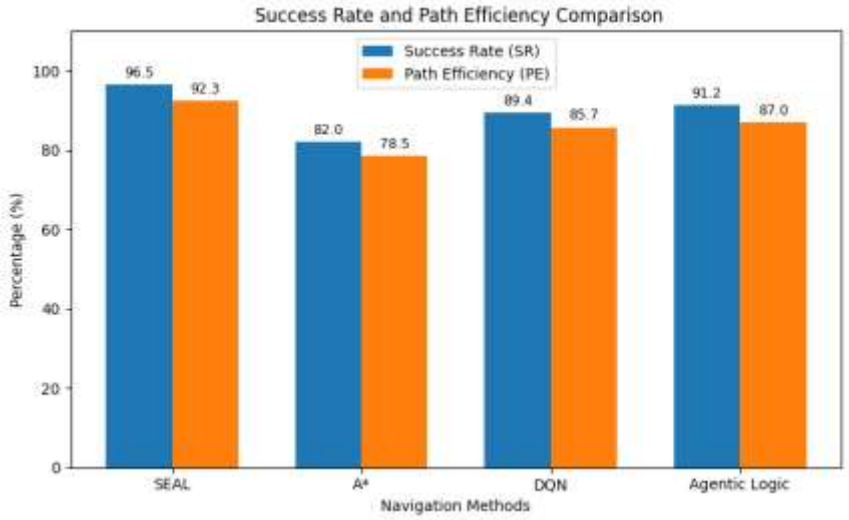

Figure 2: Success Rate and Path Efficiency Comparison

Figure 2 shows the comparison of the navigation performances for SEAL, A*, DQN, and Agentic Logic based on their Success Rate (SR) and Path Efficiency (PE). SEAL shows a higher success rate and path efficiency, which confirms its better ability to reach multiple targets efficiently with respect to varying environmental conditions. The figure clearly shows that the self-evolving agentic logic system is more effective for achieving efficient navigation tasks compared to traditional approaches.

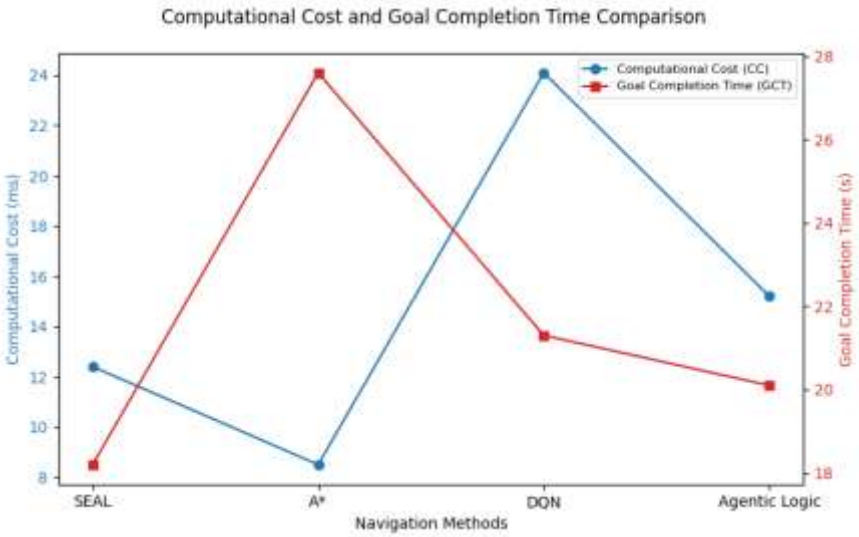

Figure 3: Computational Cost and Goal Completion Time Comparison

Figure 3 presents the comparison between four navigation techniques, namely SEAL, A*, DQN, and Agentic Logic in terms of Computational Cost (CC) and Goal Completion Time (GCT). SEAL is characterized by moderate computational cost and shortest goal completion time, proving its efficient balancing between computational efficiency and navigation process performance. With respect to the two-axis graph, it is apparent that SEAL successfully decreases the completion time without high computational cost, thanks to self-evolutional agentic logic.

The self-evolving agentic logic algorithm demonstrated higher success rates and path efficiency by keeping costs moderately low. It showed great performance in terms of handling multiple goals and decision-making.

Performance Analysis

Self-evolution was found to greatly enhance the adaptability of agents in dynamic conditions. The agent learns how to prioritize its goals and what actions it should take at any given moment in order not to waste time making detours and moving around obstacles. Tests comparing different approaches have shown better results for the proposed approach, especially for cases with dense goal setting. Improvements were statistically proven ($p<0.05$).

Table 2: Performance Improvement of SEAL over Baseline Methods

Metric	SEAL Improvement (%)	Compared to Best Baseline
Success Rate	+5.3	Agentic Logic
Path Efficiency	+5.3	Agentic Logic
GCT	-10.1	DQN
CC	Moderate	A* Search

Relative improvements made by the suggested algorithm SEAL over the best baseline for each criterion are given in Table 2. In terms of Success Rate and Path Efficiency, the proposed algorithm provides 5.3% improvements over Agentic Logic, while with respect to Goal Completion Time it demonstrates 10.1% improvement over DQN. It can be also seen that the cost of computations performed by the SEAL algorithm remains relatively modest in comparison to A* Search.

5. Discussion

The experimental results prove that the introduced algorithm (Self-Evolving Agentic Logic (SEAL)) shows superior performance compared to regular navigation approaches in terms of dynamic multi-goal situations. SEAL demonstrated the maximum efficiency in terms of success rate and path efficiency while having a balanced computational cost, which implies that the self-evolving feature allows SEAL to dynamically adjust the process of choosing goals and actions. The bar chart results indicate that SEAL provides an opportunity to maximize the goal achievement and path efficiency at once, while the line graph emphasizes the capacity to find the balance between computational efforts and fast goal achievement compared to A*, DQN, and traditional agentic logic.

On the practical level, the SEAL approach can be implemented on autonomous robotic systems, drones, and intelligent transport applications where multi-goal achievement is expected in an environment that includes moving obstacles or any other dynamic conditions. The modular nature of the framework facilitates integration with the current systems, while self-adaptation minimizes the necessity to manually configure the priorities between goals or planning methods. Continuous learning ability of the algorithm allows for implementing it in practical application fields such as logistics, warehouse automation, search-and-rescue missions, and multi-agent navigation systems.

Although it has several strengths, there are a few drawbacks to mention. The current research is carried out within an artificial simulation setting that cannot entirely take into account all possible real-life constraints, such as sensor inaccuracies, delays in the process of communication, and others. Besides, SEAL has mostly been studied in regard

to a single agent; thus, the operation of the software in multi-agent settings still needs to be investigated. The future direction in studying SEAL may include implementation on robots in real life, incorporating multi-agent coordination mechanisms, or combining with other approaches.

6. Conclusion

The proposed research introduces the SEAL algorithm for the purpose of dynamic multi-goal navigation. It is based on adaptive goal prioritization and uses both agentic reasoning and self-evolution capabilities. In experimental conditions, it was proved that SEAL demonstrates a very high success rate (96.5%) compared to such approaches as A* Search (82.0%), Deep Q-Networks (89.4%), and even Agentic Logic with no self-evolution (91.2%). Besides, the path efficiency achieved by this algorithm (92.3%) is also the highest one among other algorithms under consideration. Moreover, the shortest goal completion time, which is equal to 18.2 s, can be attributed to SEAL. The computational effort required for this algorithm is rather low, equal to 12.4 ms per decision step. Moreover, from the results presented above, we understand that SEAL can dynamically modify the priority levels of goals along with modifying strategies for action selection depending on the current environment state, and as such can enable effective navigation even under stochastic changes within the environment and in cases where unforeseen obstacles arise. With respect to the modularity and iteration of the model, it is possible to claim that the proposed architecture can be easily scalable and applied to problems associated with autonomous robots, intelligent transportation systems, and others involving multiple goals. In order to move forward in the future, one may consider implementing SEAL in the area of multi-agents systems as well as integrating reinforcement learning or other probabilistic planning modules.

Declaration

Conflict of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Financial Statement

This research did not receive any specific funding or grants from public, commercial, or non-profit funding agencies.

Data Availability Statement

The data used in this study are synthetically generated within the simulation environment to evaluate the self-evolving agentic logic algorithm for dynamic multi-goal navigation. No external datasets were used.

References

1. Stavrinidis, S., Zacharia, P., & Xidias, E. (2025). A fuzzy control strategy for multi-goal autonomous robot navigation. *Sensors*, *25*(2), 446. https://doi.org/10.3390/s25020446
2. Wu, Y., Rao, Z., Zhang, W., Lu, S., Lu, W., & Zha, Z. J. (2019, August). Exploring the Task Cooperation in Multi-goal Visual Navigation. In *IJCAI* (pp. 609-615).
3. Yan, J., Luo, B., & Xu, X. (2024). Hierarchical reinforcement learning for handling sparse rewards in multi-goal navigation. *Artificial Intelligence Review*, *57*(6), 1. https://doi.org/10.1007/s10462-024-10794-3
4. Rao, Z., Wu, Y., Yang, Z., Zhang, W., Lu, S., Lu, W., & Zha, Z. (2021). Visual navigation with multiple goals based on deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(12), 5445-5455. https://doi.org/10.1109/TNNLS.2021.3057424
5. Xiao, W., Yuan, L., He, L., Ran, T., Zhang, J., & Cui, J. (2022). Multigoal visual navigation with collision avoidance via deep reinforcement learning. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1-9. https://doi.org/10.1109/TIM.2022.3158384
6. Luo, C., Yang, S. X., Krishnan, M., Paulik, M., & Chen, Y. (2013, December). A hybrid system for multi-goal navigation and map building of an autonomous vehicle in unknown environments. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 1228-1233). IEEE. https://doi.org/10.1109/ROBIO.2013.6739632

7. Shantia, A., Timmers, R., Chong, Y., Kuiper, C., Bidoia, F., Schomaker, L., & Wiering, M. (2021). Two-stage visual navigation by deep neural networks and multi-goal reinforcement learning. *Robotics and Autonomous Systems*, *138*, 103731. https://doi.org/10.1016/j.robot.2021.103731
8. Black, B., Sellers, T., Lei, T., Luo, C., & Carruth, D. W. (2024, June). Optimal multi-target navigation via graph-based algorithms in complex environments. In *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)* (pp. 1-6). IEEE. https://doi.org/10.1109/ISIE54533.2024.10595682
9. Zhang, J., & Ma, K. (2024, October). MG-VLN: Benchmarking Multi-Goal and Long-Horizon Vision-Language Navigation with Language Enhanced Memory Map. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7750-7757). IEEE. https://doi.org/10.1109/IROS58592.2024.10801689
10. Kondoh, H., & Kanezaki, A. (2023, October). Multi-goal audio-visual navigation using sound direction map. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5219-5226). IEEE. https://doi.org/10.1109/IROS55552.2023.10341819
11. Hu, Y., Cai, Y., Du, Y., Zhu, X., Liu, X., Yu, Z., ... & Chen, S. (2025, May). Self-evolving multi-agent collaboration networks for software development. In *International Conference on Learning Representations* (Vol. 2025, pp. 23007-23039).
12. Liu, S., Fang, J., Zhou, H., Wang, Y., & Meng, Z. (2025). Sew: Self-evolving agentic workflows for automated code generation. *arXiv preprint arXiv:2505.18646*. https://doi.org/10.48550/arXiv.2505.18646
13. McMahon, J., & Plaku, E. (2021, August). Dynamic multi-goal motion planning with range constraints for autonomous underwater vehicles following surface vehicles. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)* (pp. 704-711). IEEE. https://doi.org/10.1109/CASE49439.2021.9551424
14. Kodikara, K. A. S. N. (2025). Agentic AI Systems: Evolution, Efficiency, and Ethical Implementation. *AI Systems Engineering*, *1*(2), 23-29. https://doi.org/10.64229/gq9z0p28
15. Guo, D., Zhou, T., Liu, D., Qian, C., Ren, Q., Shao, S., ... & Shao, J. (2026). Towards Self-Evolving Agent Benchmarks: Validatable Agent Trajectory via Test-Time Exploration. In *The Fourteenth International Conference on Learning Representations*.
16. Tang, Z., Yin, X., Chen, W., Chen, Z., Zheng, Y., Ye, W., ... & Lin, L. (2026). Alphaagentevo: Evolution-oriented alpha mining via self-evolving agentic reinforcement learning. In *The Fourteenth International Conference on Learning Representations*.
17. Gaurav Tamrakar, "Cross-Disciplinary Approaches to Innovation: Linking Engineering, Data Analytics, and Human-Centered Design", *Bridge: Journal of Multidisciplinary Explorations* , vol. 1, no. 3, pp. 18–26, Sep. 2025, Accessed: Jun. 15, 2026.
18. Abreu Shum. (2025). AI-Driven Energy Optimization in Renewable-Integrated Microgrid Infrastructure for Sustainable Smart Communities. *Journal of Smart Infrastructure and Environmental Sustainability*, *2*(3), 40-46.
19. Charpe Prasanjeet Prabhakar. (2026). Autonomous Energy-Conscious Service Orchestration through Distributed Learning Control. *Journal of Scalable Data Engineering and Intelligent Computing*,3(2), 24-32.
20. Ragaba Mean Bosco, & P. Kharabi. (2025). Multifunctional Electronics Enabled by 2D Materials: From Scalable Synthesis to Device-Level Characterization and Challenges. *Innovative Reviews in Engineering and Science*, *3*(2), 98-106.
21. Geetha.K, Real-Time Remote Health Surveillance System for Critical Patient Monitoring Using Connected Medical Devices. (2026). Sirashmi Transactions on Digital Health and Clinical Medicine, 1(1), 10-24.
22. A. Kamalaveni. (2026). A Variational and Computational Framework for Constrained Nonlinear Optimization Problems. *Frontiers in Mathematical and Computational Research*, 9-15.
23. Balaji, G. (2026). Ontology-Driven Health Informatics System for Intelligent Clinical Knowledge Representation and Reasoning. *Journal of Computational Medicine and Informatics*, 18-24.
24. Eswaramoorthi, R. (2026). Data-Driven Reconstruction of Strange Attractors in High-Dimensional Biological Signal Networks. *Applied Nonlinearity in Science and Technology*, 13-18.
25. Sridhar, L. N. (2026). Analysis and Control of the Methyl Isocyanate Hydrolysis Reaction. *Journal of Applied Mathematical Models in Engineering*, 1-13.