JOURNAL OF NANO- AND ELECTRONIC PHYSICS
Vol. 17 No 3, 03025(6pp) (2025)

ЖУРНАЛ НАНО- ТА ЕЛЕКТРОННОЇ ФІЗИКИ
Том 17 № 3, 03025(6cc) (2025)

REGULAR ARTICLE

Dynamics and Optimization of Physical Processes in Information Systems Using Autonomous Mobile Robots and Multi-Agent Systems

Nemala Jayasri[1], Pellakuri Vidyullatha[2,*], A. Saravanan[3], Anil Kumar Muthevi[4], K.P. Dinakaran[5], Nageswara Rao Medikondu[6,†]

[1] Research Scholar, Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India
[2] Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India
[3] Department of Electronics and Communication Engineering, Jaya Engineering College Chennai 602024, Tamilnadu, India
[4] Department of Computer Science and Engineering, Aditya University Surampalem-533437, AP, India
[5] Department of EEE, Panimalar Engineering College, Bangalore Trunk Road, Varadharajapuram, Poonamallee, Chennai-600123, India
[6] Department of Mechanical Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India

(Received 12 April 2025; revised manuscript received 14 June 2025; published online 27 June 2025)

Optimizing physical processes in information systems is crucial for enhancing the efficiency of autonomous mobile robots (AMRs) and multi-agent systems in dynamic environments. This study presents an advanced path planning and coordination approach that integrates AMRs with multi-agent strategies to improve real-time navigation and task execution. The A* (A-Star) algorithm is employed and enhanced with adaptive heuristic modifications to optimize travel time, energy efficiency, and operational throughput. A dynamic cost function is introduced to adjust path selection based on environmental constraints, obstacle distributions, and real-time system dynamics. Additionally, a multi-agent coordination framework is developed to facilitate seamless interaction among multiple robots, ensuring efficient task allocation and collision-free movement. Simulation results in structured and unstructured environments demonstrate that the proposed methodology significantly reduces travel time, enhances system-wide productivity, and optimizes physical process execution in industrial and service robotics applications. By integrating intelligent heuristic adjustments and adaptive multi-agent coordination, this approach provides a robust solution for real-time autonomous navigation and process optimization in complex, constrained environments.

Keywords: Autonomous Mobile Robots (AMRs), Multi-agent systems, Path planning, *A Algorithm**, Real-time navigation and operational optimization.

DOI: 10.21272/jnep.17(3).03025 PACS numbers: 07.05.Mh, 07.07.Tw

1. INTRODUCTION

Autonomous Mobile Robots (AMRs) play a vital role in modern industries, including manufacturing, logistics, healthcare, and smart warehousing. Efficient path planning is a fundamental requirement for AMRs to navigate dynamically changing environments while avoiding obstacles and optimizing critical performance metrics such as travel time and operational completion time [1]. Traditional path planning algorithms often prioritize shortest-path calculations, but in real-world applications, additional factors such as execution time, energy efficiency, and dynamic constraints must be considered [2]. Among the various heuristic-based path planning techniques, the *A (A-Star) algorithm** is widely utilized due to its optimal and complete search properties [3]. It employs a heuristic function to determine the best path efficiently [4]. However, conventional A* implementations often focus solely on minimizing the path length without explicitly considering travel time reduction and overall operational efficiency, which are crucial in real-time autonomous systems [5]. This study enhances the A* algorithm for AMR path planning with key contributions: an optimized cost function prioritizing travel time, adaptive heuristic selection for efficiency, and real-time navigation improvements. Simulations validate their superiority over conventional A*. The paper is structured as follows: Section 2 covers related works, Section 3 presents the system model, Sections 4-7 detail methods, results, and implementation, and Section 8 concludes with future directions. Figure 1 illustrates AMRs in the smart manufacturing industry.

* Correspondence e-mail: pvidyullatha@kluniversity.in
† medikondu1979@gmail.com

Fig. 1 – AMRs in smart manufacturing industry

2. RELATED WORK

Mobile robots have gained significant attention due to their ability to assist in various tasks, particularly when operating autonomously in diverse environments [6-8]. Effective navigation is crucial for their success, enabling them to complete tasks like urban deliveries or emergency responses [9]. Navigation methods can be classified into global and local techniques [10]. Global navigation approaches, such as Voronoi diagrams, Dijkstra's algorithm, and potential field methods, are used for broader tasks like self-driving vehicles and warehouse robots [11]. Local navigation methods, including fuzzy logic, neural networks, and optimization algorithms, focus on safety and efficient movement in dynamic or crowded environments [12-14]. Although both global and local navigation techniques have been extensively researched, there remain challenges in optimizing these systems for complex, real-time environments [15-17]. A notable research gap lies in integrating these approaches seamlessly to enhance decision-making capabilities, particularly in unstructured environments, and improving the robots' ability to adapt autonomously in changing conditions.

3. SYSTEM MODEL-AMR

The AMR system model consists of multiple interconnected components that enable autonomous navigation, obstacle avoidance, and optimal path planning. The model can be described in terms of robot kinematics, environment representation, sensor integration, and optimization objectives.

3.1 Robot Kinematics

The movement of an AMR is typically modeled using differential drive kinematics or holonomic motion models, depending on the robot's design. For a differential **drive** AMR, the motion is governed by:

$$x = v\cos(\theta)\,, y = v\sin(\theta)\,, \theta = \frac{v_r - v_t}{d} \quad (1)$$

Where

- x, y are the coordinates of the AMR,
- θ is the orientation,
- v_r, v_l are the right and left wheel velocities,
- d is the distance between wheels.

For a **holonomic robot** (e.g., omni-wheeled), motion can be controlled in any direction independently using matrix transformations. Figure 2 illustrates Robot Kinematics, while Figure 3 presents the block diagram of the AMR driving circuit with set/reset functionality.

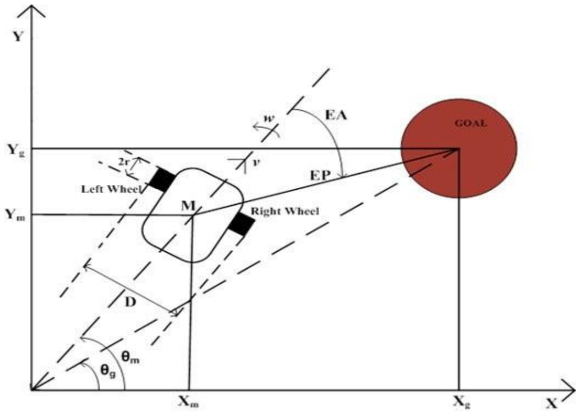


Fig. 2 – Kinematics of robot

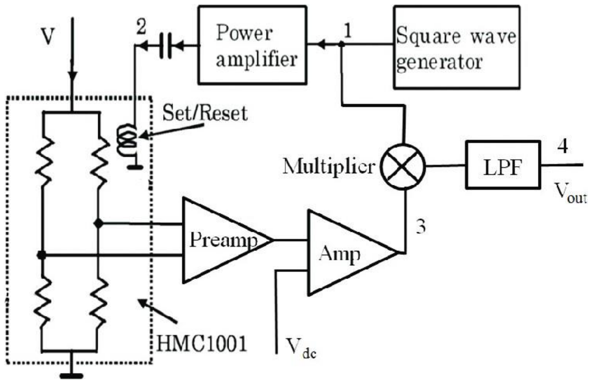


Fig. 3 – AMRs driving circuit

3.2 Sensor Integration

The AMR uses multiple sensors for environment perception and localization:

- LiDAR & Depth Cameras: Mapping
- Ultrasonic Sensors: Collision Avoidance
- IMU: Motion Estimation
- Wheel Encoders: Localization
- RGB Cameras: Object Detection

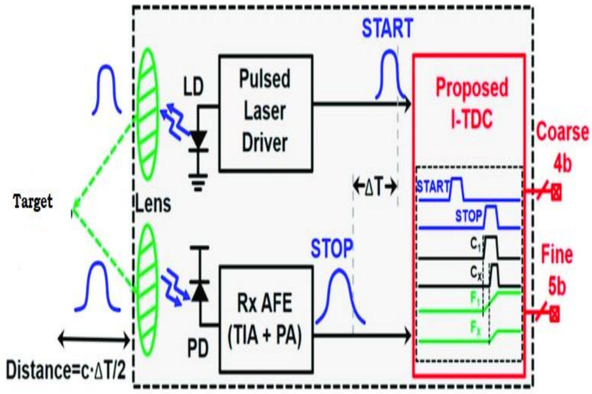


Fig. 4 – Circuit diagram for LiDAR sensor

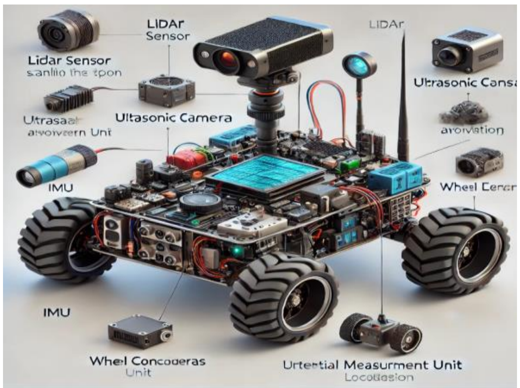


Fig. 5 – AMRs sensors integration

Figure 4 presents the block diagram of a typical LiDAR sensor, while Figure 5 illustrates the integration of sensors in AMRs.

3.3 Environmental Representation

The workspace of an Autonomous Mobile Robot (AMR) is represented as a 2D or 3D grid map or a continuous space with known and unknown obstacles.

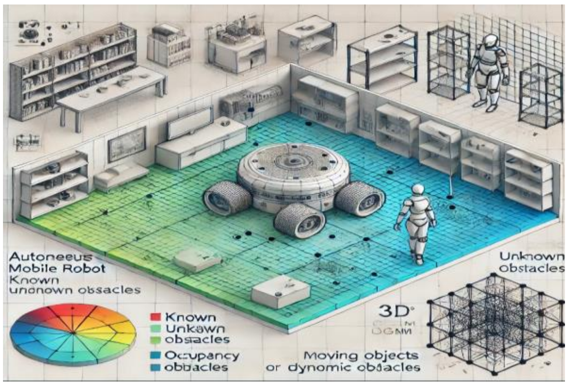


Fig. 6 – Workspace of the AMRs

The environment has static (walls, shelves) and dynamic (moving objects, humans) obstacles. Occupancy Grid Mapping and Voronoi Diagrams aid AMRs in efficient, collision-free navigation through complex and dynamic spaces.

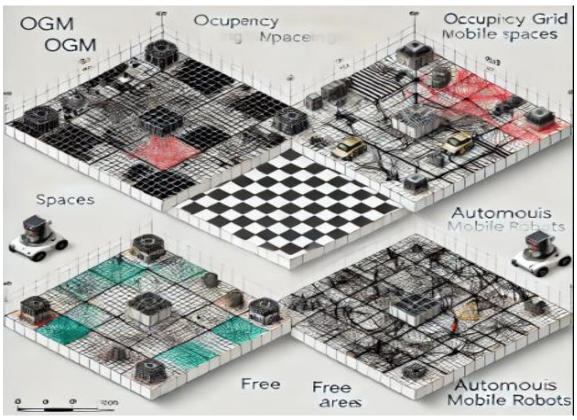


Fig. 7 – Occupancy grid mapping (OGM)

Figure 6 depicts the workspace of the AMRs, while Figure 7 illustrates Occupancy Grid Mapping.

3.4 Optimization Objectives

The path planning is optimized using:
Minimization of Travel Time (T_t):

$$T_t = \sum_{i=1}^{N} \frac{d_i}{v_i} \tag{2}$$

where d_i is the segment distance and v_i is the velocity.

Minimization of operational completion time (T_o):

$$T_o = T_t + T_{\text{processing}} \tag{3}$$

Where,
$T_{\text{processing}}$ is the computational time required for decision-making.

4. PROPOSED METHOD

The A* algorithm, developed in 1968 at SRI International, is widely used in robotics, AI, and route optimization. It finds the shortest path using actual travel cost (g-cost) and heuristic cost (h-cost) for efficient navigation and decision-making. The total cost function is given by:

$$f(n) = g(n) + h(n) \tag{4}$$

where:
$g(n)$ is represents the actual cost;
$h(n)$ is the heuristic estimate of the cost from n to the goal.
By balancing these two components, A* efficiently finds an optimal path while avoiding unnecessary exploration, making it faster than uninformed search methods like Dijkstra's Algorithm and more optimal than purely heuristic approaches like Greedy Best-First Search.

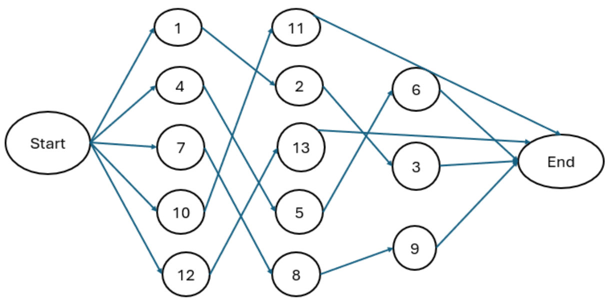
5. IMPLEMENTATION

The A* algorithm optimally solves the Bilge and Ulusoy (1995) problem by minimizing travel and operational time in job scheduling. Using heuristics and actual costs, it ensures efficient path planning. Job set 1 and Layout 2 illustrate its implementation.

Step 1: Considering the job set

Job Set No	Layout	No of Jobs	No of operations	Sequence of Machines
1	2	5	13	Job 1: 1-2-4 Job 2: 1-3-2 Job 3: 3-4-1 Job 4: 4-2 Job 5: 3-1

Step 2. Operations as network diagram



Step 3. Generate Heuristic function value $h(n)$:

O.No	1	2	3	4	5	6	7
$h(n)$	10	12	6	8	13	3	7
O.No	8	9	10	11	12	13	
$h(n)$	7	10	2	9	4	1	

Step 4. Total Cost Function calculation

O.No	M.No	$g(n) = L/U\text{-}M$	$h(n)$	$f(n)$	Priority
1	M1	4	10	14	4
4	M1	4	8	12	2
7	M3	8	7	15	5
10	M4	6	2	8	1
12	M3	8	4	12	3

Order of execution = 10 – 4 – 12 – 1 – 7

O.No	M.No	POMN	g(n)	h(n)	f(n)	Priority
11	M1	M2	2	9	11	7
2	M1	M2	2	12	13	9
13	M3	M1	10	1	11	8
5	M4	M3	12	13	25	10
8	M3	M4	2	7	9	6

Order of execution for the second set of operations = 8 – 11 – 13 – 9 – 5

O.No	M.No	POMN	$g(n)$	$h(n)$	$f(n)$	Priority
6	M2	M4	4	3	7	11
3	M3	M2	12	6	18	12
9	M4	M1	8	10	18	13

Order of execution for the final set of operations: 6 – 3 – 9

Step 4: Final Sequence of the operation according to A*
10 – 4 – 12 – 1 – 7 – 8 – 11 – 13 – 9 – 5 – 6 – 3 – 9
Step 5: Find the maximum travel time.
Step 6: Identify the machine number for each task.
4 – 1 – 3 – 1 – 3 – 4 – 2 – 1 – 1 – 3 – 2 – 4 – 1:
Step 7: Select AMR 1 for the operation.
Step 8: Identify the vehicle's previous location (VPL).
Step 9: Determine the previous operation machine number (POMN).
Step 10: Identify the vehicle ready time (VRT = 0).
Step 11: Retrieve the previous operation completion time (POCT = 0).
Step 12: Calculate vehicle empty trip time (VET = VRT + TRT1 = 0 + 0 = 0).
Step 13: Determine max vehicle empty travel time (Max(VET) = max(POCT, VET) = max(0, 0) = 0).
Step 14: Compute total travel time (TT = VET + travel time, TT = 0 + 3 = 3).
Step 15: Identify machine ready time (MRT = 0).
Step 16: Find max travel time of AMRs (Max Travel Time = max(TT, MRT) = max(3, 0) = 3).
Step 17: Add max travel time to process time for operational completion time (OCT).
Step 18: Repeat Steps 7 to 16 for remaining operations.
Step 19: Determine max travel time as job set completion time are presented in Table 1.

Table 1 – Travel time through A* algorithm

O.No	M.No	AMR No	PO MN	VRT	VET	MRT	VLT
10	4	1	0	0	0	0	3
4	1	2	0	0	0	0	2
12	3	1	0	3	5	0	9
1	1	2	0	2	5	42	42
7	3	2	0	7	10	29	29
8	4	1	3	9	53	31	54
11	2	2	4	14	31	0	36
13	1	2	3	36	37	58	58
9	1	1	4	77	77	118	118
5	3	2	1	54	58	53	60
6	2	2	3	60	80	72	86
3	4	1	2	74	75	70	77
9	1	1	4	77	77	118	118

From the above table it is observed that travel time for all operations is calculated as 118 units

6. COMPUTATIONAL INTRICACY

The complexity of A* in task scheduling depends on jobs, machines, and AMRs. For 4 layouts with 2 AMRs and 4 machines each, scheduling 10 jobs leads to $O(128^{10})$ complexity. Optimizations like heuristic pruning, branch-and-bound, or hybrid metaheuristic approaches can improve efficiency. Table 2 presents the computational complexity of different methods.

Table 2 – Computational Complexity of various methods

Methods	Complexity
Breadth-First Search (BFS)	$O(b^d)$ (where b is the branching factor, d is the depth)
Beam Search	$O(b^d)$ in worst case, but reduced to $O(W \cdot d)$ (where W is the beam width)
Tabu Search	$O(I \cdot n)$ (where I is the number of iterations, n is the problem size)
*A Search**	$O(b^d)$ in worst case, but depends on heuristic quality
*IDA(Iterative Deepening A)***	$O(b^d)$ in worst case (similar to A* but with iterative deepening)
Hill Climbing	$O(I \cdot n)$ (where I am the number of iterations, n is the number of neighbors per iteration)

7. SIMULATION RESULTS

This section analyzes travel time and operational completion time for proposed and conventional methods in FMS. The A* method was tested on ten job sets from Bilge and Ulusoy (1995) with tool requirements, creating 40 test scenarios. A case with four layouts examined TT and OCT, with results shown in Figures 8-11.

For work station-1, A* achieved a total energy saving of 48.9% compared to BFS [18] in the cases where it was more efficient. However, in several cases, A* consumed more energy than BFS. This suggests that while A* can be beneficial in some scenarios, its efficiency varies depending on specific conditions.

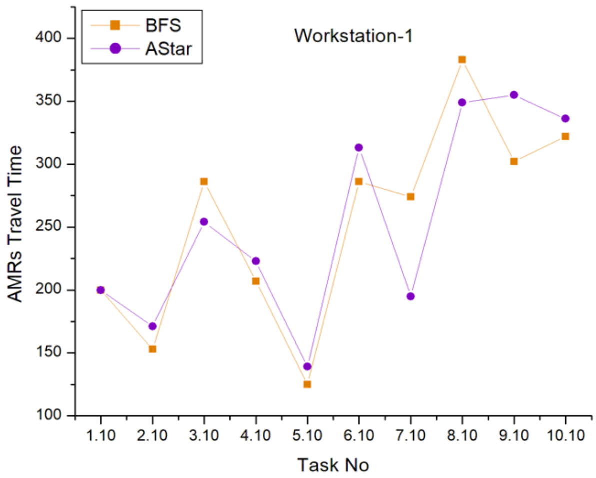


Fig. 8 – AMRs activities in WS-1

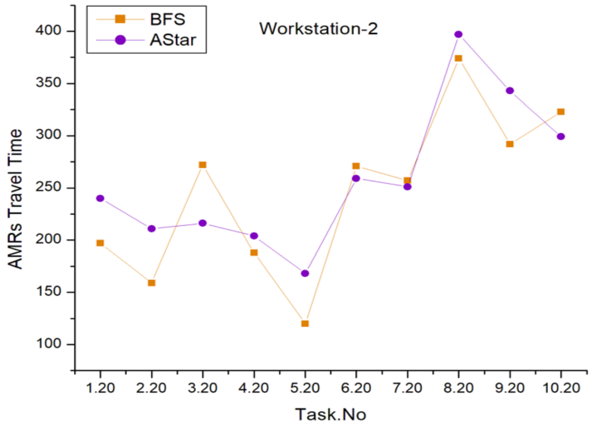


Fig. 9 – AMRs activities in WS-2

For work station-2, A* achieved a total energy saving of 34.78 % in the cases where it was more efficient than BFS. However, in multiple instances, A* consumed more energy, indicating that its effectiveness depends on specific conditions within the layout.

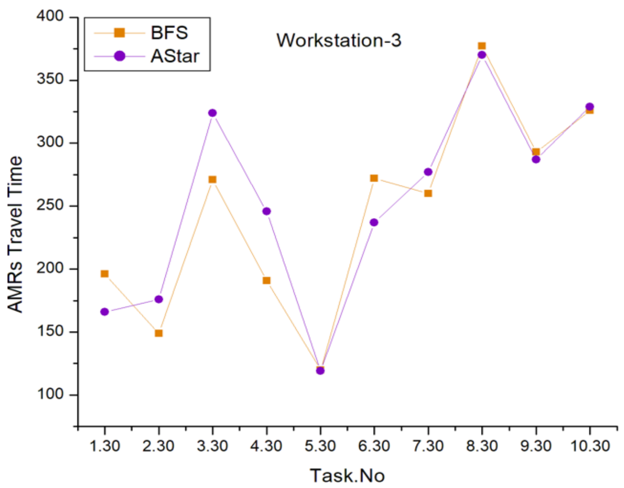


Fig. 10 – AMRs activities in WS-3

For work station-3, A* achieved a total energy saving of 32.91 % in the cases where it was more efficient than BFS. Similar to other layouts, A* shows potential energy savings, but its performance can vary depending on the specific scenario.

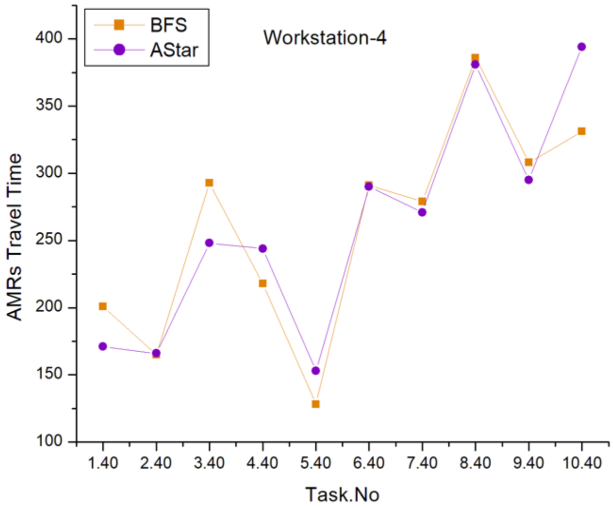


Fig. 11 – AMRs activities in WS-4

For work station-4, A* achieved a total energy saving of 39.01 % in the cases where it was more efficient than BFS. As with the other layouts, A* demonstrated significant energy savings in certain cases, but its efficiency can vary depending on the specific scenario.

8. CONCLUSIONS

This study compared the energy efficiency of the A* and BFS algorithms across four layouts. A* showed energy savings in certain cases, with Layout-1 achieving 48.9 %, Layout-2 34.78 %, Layout-3 32.91 %, and Layout-4 39.01 %. However, A* also consumed more energy than BFS in some instances. The results indicate that A* can be more energy-efficient depending on the specific problem. Future work could focus on optimizing A* for better energy savings through heuristic adjustments and hybridizing it with BFS. Testing larger, more complex problems would also provide insights into its scalability and real-world efficiency.

REFERENCES

1. B. Rehman, R. Yagfarov, A. Klimchik, *IFAC-PapersOnLine* **51**, 354 (2018).
2. J. Ni, L. Wu, X. Fan, S.X. Yang, *Comput. Intell. Neurosci.* **2016**, 3810903 (2016).
3. O. Takahashi, R.J. Schilling, *IEEE Trans. Robot. Autom.* **5** No 2, 143 (1989).
4. P. Bhattacharya, M.L. Gavrilova, *IEEE Robot. Autom. Mag.* **15** No 2, 58 (2008).
5. E.J. Gómez, F.M. Santa, F.H.M. Sarmiento, *2nd International Congress of Engineering Mechatronics and Automation, CIIMA* (2013).
6. R. Abiyev, D. Ibrahim, B. Erin, *Adv. Eng. Software* **41**, No 10–11, 1179 (2010).
7. A.R. Soltani, H. Tawfik, J.Y. Goulermas, T. Fernando, *Adv. Eng. Inform.* **16** No 4, 291 (2002).
8. E. Masehian, M.R. Amin-Naseri, *J. Robot. Syst.* **21** No 6, 275 (2004).
9. M. Weigl, B. Siemiatkowska, K.A. Sikorski, A. Borkowski, *Robot. Auton. Syst.* **11** No 1, 13 (1993).
10. K.-H. Park, Y.-J. Kim, J.-H. Kim, *Rob. Auton. Syst.* **35** No 2, 109 (2001).
11. M.B. Montaner, A. Ramirez-Serrano, *Expert Syst. Appl.* **14** No 1-2, 179 (1998).
12. I. Engedy, G. Horváth, *11th IEEE International Symposium on Computational Intelligence and Informatics, CINTI*, 213 (2010).
13. A. Zhu, S.X. Yang, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **37** No 4, 610 (2007).
14. A. Ghorbani, S. Shiry, A. Nodehi, *International Conference on Future Computer and Communication, ICFCC 2009*, 164 (2009).
15. H. Miao, Y.C. Tian, *Appl. Math. Comput,* **222**, 420 (2013).
16. N.B. Hui, V. Mahendar, D.K. Pratihar, *Fuzzy Set. Syst.* **157** No 16, 2171 (2006).
17. D.N.M. Abadi, M.H. Khooban, *Journal of King Saud University - Engineering Sciences* **27**, 92 (2015).
18. A.V.R. Surya Prasad, N.R. Medikondu, M.B.S. Sreekara Reddy, K. Rakesh, K.S. Raghuram, Tribhuwan Kishor Mishra, AIP Conf. Proc. **2707** No 1, 020028 (2023).
19. K.V. Subbaiah et al., *International Journal of Physical Sciences* **4** No 2, 139 (2009).

Динаміка та оптимізація фізичних процесів в інформаційних системах з використанням автономних мобільних роботів та багатоагентних систем

Nemala Jayasri[1], Pellakuri Vidyullatha[2], A. Saravanan[3], Anil Kumar Muthevi[4], K.P. Dinakaran[5], Nageswara Rao Medikondu[6]

[1] *Research Scholar, Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India*
[2] *Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India*
[3] *Department of Electronics and Communication Engineering, Jaya Engineering College Chennai 602024, Tamilnadu, India*
[4] *Department of Computer Science and Engineering, Aditya University Surampalem-533437, AP, India*
[5] *Department of EEE, Panimalar Engineering College, Bangalore Trunk Road, Varadharajapuram, Poonamallee, Chennai-600123, India*
[6] *Department of Mechanical Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram-522302, AP, India*

Оптимізація фізичних процесів в інформаційних системах має вирішальне значення для підвищення ефективності автономних мобільних роботів (AMR) та багатоагентних систем у динамічних середовищах. Це дослідження представляє вдосконалений підхід до планування та координації шляхів, який інтегрує AMR з багатоагентними стратегіями для покращення навігації та виконання завдань у реальному часі. Алгоритм A* (A-Star) використовується та вдосконалений адаптивними евристичними модифікаціями для оптимізації часу подорожі, енергоефективності та операційної пропускної здатності. Вводиться динамічна функція вартості для коригування вибору шляху на основі обмежень навколишнього середовища, розподілу перешкод та динаміки системи в реальному часі. Крім того, розроблено структуру багатоагентної координації для забезпечення безперебійної взаємодії між кількома роботами, забезпечуючи ефективний розподіл завдань та рух без зіткнень. Результати моделювання у структурованих та неструктурованих середовищах демонструють, що запропонована методологія значно скорочує час подорожі, підвищує продуктивність усієї системи та оптимізує виконання фізичних процесів у промислових та сервісних робототехнічних застосуваннях. Завдяки інтеграції інтелектуальних евристичних налаштувань та адаптивної багатоагентної координації, цей підхід забезпечує надійне рішення для автономної навігації та оптимізації процесів у реальному часі в складних середовищах з обмеженими можливостями.

Ключові слова: Автономні мобільні роботи (AMR), Багатоагентні системи, Планування шляху, Алгоритм A*, Навігація в режимі реального часу та операційна оптимізація.