ADVANCES IN CYBER-PHYSICAL SYSTEMS
Vol. 10, No. 2, 2025

PHYSICS-INFORMED PARTICLE SWARM OPTIMIZATION FOR COLLISION-AWARE SWARM NAVIGATION

Oleh Sinkevych[1], Bohdan Sokolovskyy[1], Yaroslav Boyko[1], Zenyk Matchyshyn[2]

[1]*Ivan Franko National University of Lviv, 50, M. Drahomanova str., Lviv, 79005, Ukraine,*
[2]*Capgemini Engineering, Cambridge, Massachusetts, United States.*
Authors' e-mails: *oleh.sinkevych@lnu.edu.ua, bohdan.sokolovskyy@lnu.edu.ua, yaroslav.boyko@lnu.edu.ua, zenykm@gmail.com*

https://doi.org/10.23939/acps2025.02.197

Submitted on 11.09.2025

***Abstract*: This paper presents an approach to modeling the movement of a multi-agent system in a two-dimensional space using a modified Particle Swarm Optimization (PSO) algorithm, adapted to account for the physical properties of the agents. The standard PSO, originally designed for solving optimization problems through swarm behavior, has been extended to simulate the motion of physical objects with defined mass, velocity, and inter-agent interactions. To ensure physically plausible motion and prevent collisions between agents, hybrid methods have been proposed that combine classical PSO with inter-particle potential functions. Trajectory planning and control over the direction and speed of agent movement have been governed by the modified PSOs, while collision avoidance is achieved through the influence of repulsive potential fields. Numerical simulations have been conducted to analyze the collective behavior of the swarm.**

***Index terms*: particle swarm optimization, collision-aware motion, swarm intelligence, robotics.**

I. INTRODUCTION

A collision-aware motion of swarm objects is among important research directions in fields like safe pathfinding, navigation, and robotics. While the individual motion of physical objects has been widely studied, safe and collision-free multi-agent movement is still active in engineering design. The integration of nature-inspired ideas with the stochastic elements into a robotic coordination effectively complements deterministic decisions, offering a perspective on the problem through the lens of coordination mechanics among collective living beings. The particle swarm optimization, first introduced in 1991 and overviewed in [1], is among nature-inspired metaphors applied to the variety of optimization [2], pathfinding [3] and robots pathplanning [4] problems. As it primarily serves as the basis for abstract formulations, it has rarely been extended into terms grounded in the physical world. To address this gap and explore the potential of transforming the PSO concept into a physically accurate motion engine for real-world objects, we propose a modified version. This modification serves as the foundation for a collision-aware swarm navigation algorithm. The most significant. changes to the standard PSO in our approach include the changes to the standard PSO in our approach include the incorporation of particle masses, and repulsive forces. These are modeled using a power-law potential function with varying exponent parameters; these functions define the repulsive force between particles, which is proportional to the distance between them and accounted for the physical size of the particles.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

The integration of physical-based concepts with standard PSO is not new. For example, in [5] the authors incorporated a bi-matrix game theory model and a novel RPQPSO algorithm to optimize multi-AUV target allocation in non-cooperative environments by computing Nash equilibria based on situational factors. In [6] the improved version of repulsive PSO algorithm to solve this inverse scattering problem was developed; in [7] the new PSO modification with repellent and attraction forces for the numerical optimization was proposed; the additional research in this area is outlined in [8]. Unfortunately, the main limitation of these concepts lies in their lack of swarm robotics applicability. The research in these papers is primarily focused on numerical optimization problems, where particle collisions are irrelevant to the outcome. Incorporating PSO for robotic path planning is shown in [9], where PSO is used as a navigator for one agent. The authors of [10] introduced a PSO algorithm that incorporates a Gaussian repulsion force, enabling its application to multiple interacting swarms. Although [11] introduced a repulsion-based modification to PSO for drones in 3D space, the work was limited by an insufficient study of the algorithm's parameters and the physics behind the proposed model. In [12] and [13] PSO is mostly used as optimizational routine mixed with the RRT* algorithm for multi-UAV path planning [12] and game-based model [13].

An analysis of the existing literature reveals that while PSO is predominantly employed as an advanced numerical optimization algorithm, there is a notable scarcity of results concerning its direct application as a motion model for multi-agent systems.

The problem statement for this research is defined as follows: to design and formalize a multi-agent motion model based on PSO mechanics that satisfies the two primary constraints:

a) it must ensure collision-free navigation among all agents;

b) it must successfully guide the swarm to a predefined target.

The central challenge lies in effectively balancing the behavioral parameters to meet both constraints simultaneously.

III. SCOPE OF WORK AND OBJECTIVES

We consider a motion model of a multi-agent system consisting of particles p_i with a certain radius r in a two-dimensional space. To ensure safe navigation within a bounded map of size H x W free of obstacles, the following are necessary: a) an objective function that ‘attracts’ agents toward the target, simulating their goal-directed movement; b) a collision avoidance mechanism between agents. To realize the first point and coordinate agent’s movement a modified PSO algorithm is employed, providing simple and efficient navigation by minimizing Euclidean distance functions $F\left(x_i, x_j\right) = \sqrt{\| x_i - x_j \|^2}$. For collision avoidance, an extension to the PSO algorithm based on a power potential function is proposed, effectively modeling repulsive interactions among agents.

Thus, the scope of this work is to develop an efficient control method for multi-agent system movement that simultaneously achieves target convergence and collision avoidance within a constrained environment. The main objectives of the. study include formulating a mathematical motion model for agents, developing the modified PSO algorithm to incorporate repulsive forces, and conducting numerical experiments to evaluate the effectiveness of the proposed approach.

A. SWARM MOVEMENT

Given a two-dimensional map of pre-defined size H x W, a swarm of agents is initially distributed at random within a radius r_i around an initial position I_p. The collective goal of the swarm is to reach a target located at position $T_p = (x_t, y_t)$ through coordinated, collision-free motion while maintaining swarm cohesion. Agents are capable of communicating either within a local neighborhood or across the entire swarm to obtain the positions of neighboring agents.

Following the PSO paradigm, the motion of each agent is governed by the components of its velocity vector:

$$c = c_1 r_1 \left(p_i^{best} - p_i\right), \tag{1}$$

$$s = c_2 r_2 \left(l^{best} - p_i\right), \tag{2}$$

$$v_d = w v_i + c + s, \tag{3}$$

$$v_n = v_d - v_i, \tag{4}$$

where c is the cognitive component, representing the influence of the agent's own best-known position, s is the social component, representing the influence of the best-known position among neighboring agents (or the entire swarm), w is the inertia coefficient, controlling the influence of the previous velocity, v_i is the agent’s current velocity, v_d is the desired velocity and v_n is the velocity adjustment needed to achieve the desired state, c_1 is the cognitive weight, c_2 is the social weight, r_1 and r_2 are random values from range (0, 1]. The incorporation of random values serves to inject stochasticity into the algorithm, thereby facilitating escape from local minima. In the context of this paper, where obstacles are not considered, these stochastic elements can be conceptualized as inherent noise in the agents' motion. Agent position is defined as p_i personal best position by far is p_i^{best}, swarm best position is l^{best}

To ensure physically plausible motion based on the agent's mass, the velocity change is constrained by a maximum allowable acceleration:

$$v_{max} = \Delta t * a_{max} / m_p, \tag{5}$$

where $\Delta t \to \infty$ is the time difference, a_{max} is the maximum acceleration, m_p is the mass of the agent.

If $\| v_n \| > v_{max}$ the velocity change is scaled down to respect the acceleration limit. The agent's updated velocity is then given by. $v_{PSO} = v_i + v_n$.

B. REPULSIVE FORCES

To prevent collisions between agents, we propose the incorporation of repulsive potential forces, which influence agent behavior based on their relative proximity and direction of approach.

Let’s define the potential function

$$U_{i,j}\left(r\right) = r^{-n} = (\left|\left|p_i - p_j\right|\right| - 2R)^{-n}. \tag{6}$$

Here is the surface-to-surface distance between agents, $n \geq 1$ is the positive exponent controlling the strength of the repulsion, $\|p_i - p_j\|$ is the distance between the centers of the agents, $2R$ accounts for the radius of both agents (assuming equal radii). The repulsive force between these two agents is given as $F_{i,j} = -\nabla_{p_i} U_{i,j}\left(r\right) = -\frac{d}{dr}\left(r^{-n}\right)$ which turns into:

$$F_i = n \sum_{j \neq i} \left(p_i - p_j\right) / \left(r^{n+1} \left\|p_i - p_j\right\|\right). \tag{7}$$

For computational efficiency and numerical stability, we approximate the analytical repulsive force expression (7) with a numerically stable formulation:

$$F_{doundary} = \frac{n}{gap^{n+1}} \frac{p_i - p_{closest}}{\| p_i - p_{closest} \| + \varepsilon}, \tag{8}$$

where ε is a small positive constant added to avoid division by zero. This approximation avoids explicit normalization of the direction vector and merges its into the scalar coefficient, while preserving the qualitative behavior of the original force function.

C. COLLISION-FREE PSO ALGORITHM

The combination of the swarm motion concept and the repulsive forces results in the following framework:

1. Swarm initialization: n agents of radii R are generated within a radius r_i around an initial position I_p given the target $T_p = (x_t, y_t)$; maximum velocity (the upper bound on how fast the agent is allowed to move), communication distance, mass, maximum acceleration and Δt for the simulation are initialized.

2. If two or more particles collide or tunnel through each other, they are discarded from the simulation by being removed as broken physical agents. The rest of the particles are tagged as 'survived'.

3. Stopping criteria: simulation is considered complete when the all '*survived*' agents have arrived within a predefined tolerance distance (goal radius) R_G from the target position.

4. Swarm movement calculation: during each iteration, agent's velocity is calculated via (1)-(5) formulas. For each agent the repulsion acceleration is calculated as (8) multiplied by $w_r * F_i / m_p$, where w_r is the repulsion weight. Here, F_i consists of repulsion from the boundaries $F_i = F_i + F_{boundary,}$ where

$$F_{boundary} = \frac{n}{gap^{n+1}} \frac{p_i - p_{closest}}{\left\| p_i - p_{closest} \right\| + \varepsilon}, \qquad (9)$$

$gap = \|p_i - p_{closest}\| - R$ is the surface-to-wall distance and $p_{closest}$ is the nearest point on the boundary. The power-law formulation provides tunable repulsion characteristics: higher values create more localized, stronger repulsion near contact, while lower values produce gentler, longer-range repulsion fields.

5. .Hence repulsion velocity change via acceleration $a_r = (w_r * F_i * \Delta t)/m_p$, then we need to convert PSO velocity into acceleration as

$$a_{pso} = (v_{pso} - v_i) / \Delta t, \qquad (10)$$

$$a_{total} = a_{pso} + a_r, \qquad (11)$$

$$v_i = v_i + a_{total} \Delta t. \qquad (12)$$

6. If the computed velocity exceeds the maximum allowed speed, the velocity vector should be rescaled to the maximum magnitude while preserving its direction.

7. Based on these formulas, position update is given as

$$p_i = p_i + v_i * \Delta t. \qquad (13)$$

8. In the simulation framework, agents are designed to move sequentially in discrete time steps. Nevertheless, as the time increments $\Delta t \to 0$, the updates of all agents' positions occur nearly simultaneously, thereby approximating parallel motion dynamics

To study algorithm performance we propose the following metrics: number of broken and survived agents, total overlap time, convergence time, success flag, total energy used as $\sum_{steps} 0.5 * m_p v_c^2$, where $v_c = \left\| v_i - v_i^{prev} \right\|$, swarm cohesion as the mean distance to the center of the swarm, path efficiency which is calculated as the average ratio of straight-line distance from initial to target position over total path length traveled by each agent, capped at 1.0, average speed, and energy efficiency as the ratio of total energy used and number of active agents.

IV. NUMERIC SIMULATION

For the numeric experiments, we generated a 2D map with dimensions of 300×300 units. All agents had identical properties, with a radius R=1 and mass parameter m_p=1, and were capable of global communication across the entire swarm. This global communication was required for two purposes: (a) determining the global best position, and (b) calculating the repulsive influence of other agents on a given iii-th agent. During initialization, the agents were randomly distributed around the point I_p=(10, 10) within a radius of 30 units. The target was at T_p = (290, 296) and the tolerance distance around the target is R_G = 15.

For all the experiments maximum acceleration has been set to 3 *units* /s^2, maximum allowed velocity is set v_{max} = 5 *units* /s and $\Delta t \in \{0.1\text{s}, 0.05\text{s}\}$.

An example of agent movement visualization on the map is shown in Fig. 1. For this simulation $\Delta t = 0.05$, number of agents equals 10, w=1, $c_1 = c_2$ =2, w_r =3, n = 6. The result of this simulation is as follows: all ten survived particles reached target at time 89.8s, total steps are 1797, path efficiency equals 0.87, average agents' speed is 4.94 and swarm cohesion is 3.79.

The obtained results demonstrate that, under the selected swarm and algorithmic parameters, all agents successfully reached the designated target without collisions or mutual tunneling. However, it should be noted that this represents a single simulation run; due to the stochastic nature of the algorithm, variations in outcomes are expected across multiple trials. A more comprehensive statistical analysis of these variations will be presented in the following section.

Conversely, in the absence of the proposed repulsion method, the agent movement simulation terminates after an average of 5.1 seconds due to complete inter-agent collision occurring within a few iterations.

To analyze the influence of combining PSO and potential-based repulsion parameters, namely the inertia coefficient: inertia w, cognitive c_1 and social c_2 coefficients, repulsion weight w_r and power n on the swarm's ability to successfully reach the target and maintain agent survival, we conducted 300 numerical experiments and collected metrics describing the swarm dynamics.

The first experiment was conducted with a time step of $\Delta t = 0.05$. For the remaining parameters, we defined the following ranges we chose the range of their change as: w: (0.3, 0.9), c_1: (1, 2.5), c_2: (1.5, 3), w_r: (0.5, 3), n: (1, 6). A Latin hypercube sampling method was then used to ensure an even coverage of the parameter space. Based on these parameter combinations, 299 out of 300 simulations (99%) were successful, with only one unsuccessful run. In 283 of the experiments, all 10 agents reached the target area; in 9 cases, 8 agents reached it; in 6 cases, 6 agents reached it; in 1 case, 4 agents reached it; and in 1 case, no agents reached the target.

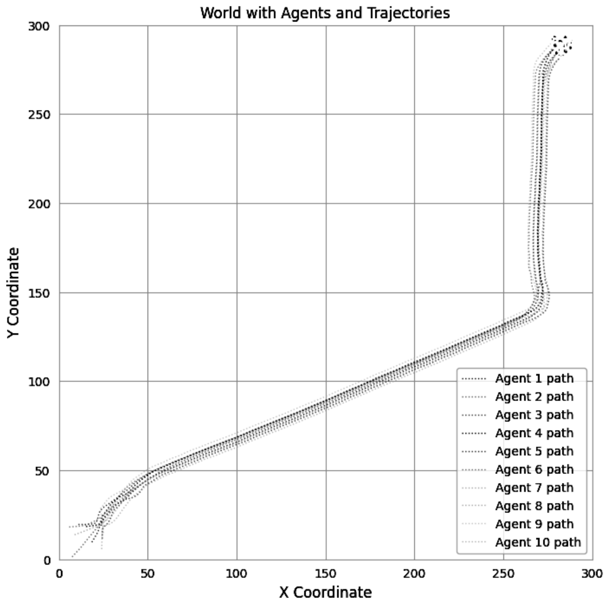


Fig. 1. Visualization of agents' movement, Δt = 0.05

The experimental evaluation yielded the following performance characteristics: mean total energy consumption was 240.1 (range: 47.1–1647.1), with an average agent speed of 4.6, mean swarm cohesion of 3.75, mean path efficiency The second experiment, where $\Delta t = 0.1$ showed slightly worse results. Out of 300 simulation runs, 295 were successful, with an average of eight surviving agents. The mean total energy consumption was 509.8 (range: 68.7–2531), while the average agent speed was 4.45, consistent with the results of the first experiment. The mean swarm cohesion was 3.75, the mean path efficiency was 0.90, and the mean energy efficiency was 72.05. The average convergence time across all simulations was 172.2 seconds. The parameter–metric correlation analysis revealed moderate associations only between the repulsive power n and total energy consumption (0.44), as well as between n and swarm cohesion (0.46).

The example of agents' movement for $\Delta t = 0.1$ is provided in fig. 2. Here one can see that all the agents survived as well as all of them reached the target area. Here, the convergence time is 89.8s, swarm cohesion is 3.2, total steps are 862, and path efficiency equals 0.95.

Based on the obtained results, we can draw three important conclusions:

a) Δt is the crucial parameter for preventing collisions/tunneling in a multi-agent simulation. Smaller Δt reduces overlap and tunneling because agents move smaller distances per step, making discrete updates a better approximation of continuous motion;

b) repulsive power n has a likely moderate, but tangible impact on agents' energy consumption and swarm cohesion, which is important to keep the swarm formation solid;

c) the proposed approach significantly reduces collisions and tunneling effects among particles (or agents) when their navigation and swarm behavior are driven by the PSO concept.

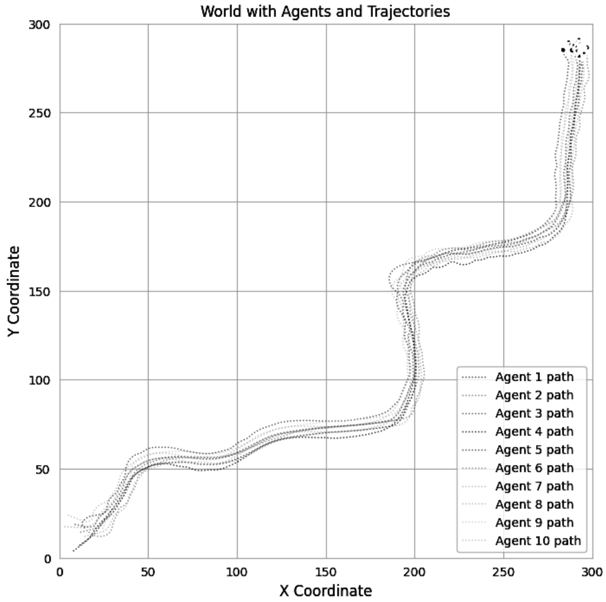


Fig. 2. Visualization of agents' movement, Δt = 0.1

V. CONCLUSION

This paper presented a novel approach to collision avoidance in the motion of multi-agent systems. The proposed method was based on the Particle Swarm Optimization (PSO) algorithm, which ensured efficient swarm navigation toward a target and incorporates mechanisms for inter-agent interaction. Since the standard PSO algorithm lacked inherent collision avoidance capabilities, the authors proposed its modification through the introduction of a potential function. The implementation of this function enabled the modeling of inter-agent repulsion. The repulsive force was governed by two parameters: a weighting coefficient and an exponent in the denominator of the potential function. The study examined the influence of various algorithm configurations on the swarm's motion toward the target and the number of potential collisions between agents, depending on the method's parameters. The obtained results demonstrated the effectiveness and potential of the proposed approach for addressing safe motion problems in multi-agent systems.

VI. CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

VII. DECLARATION ON GENERATIVE AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. *IEEE Access*, *10*, 10031–10061. https://doi.org/10.1109/access.2022.3142859.

[2] Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. *IEEE Access*, *10*, 10031–10061. DOI: https://doi.org/10.1109/access.2022.3142859.

[3] Hilli, A. A., Al-Ibadi, M., Alfadhel, A. M., Abdulshaheed, S. H., & Hadi, A. H. (2021). Optimal path finding in stochastic quasi-dynamic environments using particle swarm optimization. *Expert Systems With Applications*, *186*, 115706. DOI: https://doi.org/10.1016/j.eswa.2021.115706.

[4] Lin, S., Liu, A., Wang, J., & Kong, X. (2023). An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse. *Journal of Computational Science*, *67*, 101938. https://doi.org/10.1016/j.jocs.2022.101938

[5] Lin, C., Yu, D., & Lin, S. (2024). Allocation strategy optimization using repulsion-enhanced quantum particle swarm optimization for multi-auv systems. *Journal of Marine Science and Engineering*, *12*(12), 2270. https://doi.org/10.3390/jmse12122270

[6] Zhai, C., & Cao, Z. (2020). GPU-accelerated repulsive particle swarm optimization for parameter retrieval of single sphere from angularly distributed scattered light. *Optik*, *220*, 165222. https://doi.org/10.1016/j.ijleo.2020.165222

[7] Zhu, S., Zhou, S., Shang, J., Wang, L., & Qiang, B. (2020). A multiion particle swarm optimization algorithm based on repellent and attraction forces. *Concurrency and Computation: Practice and Experience*. DOI: https://doi.org/10.1002/cpe.5979.

[8] Liu, Q., Li, J., Ren, H., & Pang, W. (2022). All particles driving particle swarm optimization: Superior particles pulling plus inferior particles pushing. *Knowledge-Based Systems*, 108849. DOI: https://doi.org/10.1016/j.knosys.2022.108849.

[9] Benmachiche, A., Derdour, M., Kahil, M. S., Ghanem, M. C., & Deriche, M. (2025). Adaptive hybrid PSO–APF algorithm for advanced path planning in next-generation autonomous robots. *Sensors*, *25*(18), 5742. DOI: https://doi.org/10.3390/s25185742.

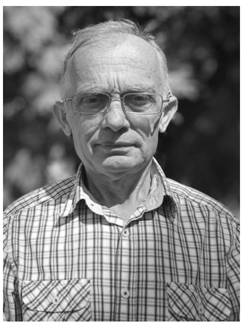
[10] Paez, D., Romero, J. P., Noriega, B., Cardona, G. A., & Calderon, J. M. (2021). Distributed particle swarm optimization for multi-robot system in search and rescue operations. *IFAC-PapersOnLine*, *54*(4), 1–6. DOI: https://doi.org/10.1016/j.ifacol.2021.10.001.

[11] Parker, L., Butterworth, J., & Luo, S. (2019, 17 July). *Fly safe: Aerial swarm robotics using force field particle swarm optimisation*. https://arxiv.org/. DOI: https://arxiv.org/pdf/1907.07647.

[12] Meng, Q., Chen, K., & Qu, Q. (2024). PPSwarm: Multi-uav path planning based on hybrid PSO in complex scenarios. *Drones*, *8*(5), 192. DOI: https://doi.org/10.3390/drones8050192.

[13] Wan, Z., Gan, L., Zhang, L., & Shu, Y. (2025). Multi-ship collision avoidance decision-making based on cooperative game and particle swarm optimization algorithm. *Ocean Engineering*, *341*, 122537. DOI: https://doi.org/10.1016/j.oceaneng.2025.122537.

Oleh Sinkevych, PhD, was born in Lviv, Ukraine, in 1988. Starting from 2023, he has been working as Associate Professor at the Faculty of Electronics and Computer Technologies of Ivan Franko National University of Lviv. His research interests encompass machine learning, natural language processing, swarm AI algorithms, numerical optimization, and metaheuristics.



Bohdan Sokolovskyy, PhD, was born in Kulykiv, Lviv region, Ukraine, in 1950. Since 2014, he has been working as Associate Professor at the Faculty of Electronics and Computer Technologies of Ivan Franko National University of Lviv. His research interests encompass computer modeling of nonuniform and methods of stochastic optimization.

Yaroslav Boyko, PhD, was born in Lviv region, Ukraine, in 1967. Since 2017, he has been working as Associate Professor at the Faculty of Electronics and Computer Technologies of Ivan Franko National University of Lviv. His research interests are Internet of Things and Fog/Edge computing., and metaheuristics.

Zenyk Matchyshyn was born in Lviv, Ukraine. Graduated from Ivan Franko National University of Lviv, faculty of applied mathematics and informatics. He is currently chief technology officer in Software Products Engineering, Capgemini Engineering. His research interests are Generative AI, Machine Learning and Agentic AI

Franko National University of Lviv. His research interests encompass computer modeling of nonuniform and methods of stochastic optimization.