Applied Computer Systems
ISSN 2255-8691 (online)
2026, vol. 31, no. 1, pp. 63–73
https://doi.org/10.2478/acss-2026-0006
https://content.sciendo.com

1862
RIGA TECHNICAL UNIVERSITY

Physics-Inspired Hamiltonian Particle Swarm Optimisation for Multi-Agent Movement

Oleh Sinkevych[1*], Bohdan Sokolovskii[2], Yaroslav Boyko[3], Igor Olenych[4]
[1–4]*Faculty of Electronics and Computer Technologies, Ivan Franko National University of Lviv, Lviv, Ukraine*

***Abstract* – This study puts forward a Hamiltonian-inspired modification of Particle Swarm Optimisation (PSO) algorithm. Since the standard PSO procedure does not take into account physical properties like particle masses, geometrical sizes, and energy consumption, it is not fully applicable as a navigational and coordination tool in real-world environments. In particular, generic PSO mechanism cannot stop the particles from collisions. To address these issues, we propose a new PSO formulation based on a Hamiltonian interpretation. This approach allows bringing together the kinetic and potential energy terms with the forces acting on the agents, as well as the derivation of agent's velocities and positions. The potential energy represents attraction toward both personal and global best positions in a spring-like manner. As a component of conservative forces derived from the potential energy term, we introduce a special repulsive potential function to prevent collisions among agents. The kinetic energy, which is derived via agent mass and momentum, determines the movement dynamics. To model the energy loss, we incorporate Rayleigh dissipation term that accounts for non-conservative forces. According to the proposed model, agent displacements are computed using the obtained velocity and momentum vectors. Additionally, we introduce individual and swarm energy efficiency metrics to study the agents' motion in a 2D testing environment. The presented approach enables stable, coordinated, and collision-free multi-agent motion within a physics-inspired optimisation framework.**

***Keywords* – Collision avoidance, particle swarm optimisation, pathfinding, physics-inspired algorithms, swarm intelligence.**

I. Introduction

Nature-based algorithms provide a solid foundation for modelling the motion of various kinds of swarm objects. These include abstract massless particles from optimisation algorithms, game bots, unmanned air and ground vehicles, and scientific exploration probes. Inspired by collective living beings such as birds, ants, and bees, these algorithms simulate swarm behaviour by defining rules of inter-agent and environment interactions. However, since these algorithms were primarily designed for numerical optimisation, they remain mostly abstract [1]. Their mathematical formulations typically do not take into account the physical characteristics of agents, such as size, mass, and energy. Besides, motion-related quantities like momentum and acceleration usually are beyond the model scopes [2]. For instance, while Particle Swarm Optimisation (PSO) algorithm incorporates some simplification of velocity and inertia, it does not fully count for physically thorough motion with the mass and forces [3].

Considering their effectiveness in numerical optimisation problems, i.e., finding the route to an optimal or suboptimal solution, many of nature-inspired approaches can be utilised as the guidance mechanism for real-world robotics [1]. Thus, combining their search capabilities as the navigation element with physically accurate motion principles allows for the development of new approaches for multi-agent robotic systems [4].

The concept of PSO, originally developed in the context of swarm behaviour observed in flocks of birds, despite its primary focus on function optimisation, incorporates a mechanism of inter-agent interaction [5]. This mechanism enables the coordination of agents' movements toward a common goal. The latter can be treated as the optimum of some predefined objective function. Such coordination is achieved through two key components: the cognitive component, which stands for the agent movement toward its own best-known position, and the social component, which stands for the attraction toward the global best swarm position [6]. These components may either stay in balance or shift in relative priority depending on the agents' positions [3].

Thereby, PSO can serve as a foundation for developing efficient algorithms for the coordination and navigation of multi-agent physical systems, such as groups of coordinated robots [7], [8]. Since the development of such algorithms requires substantial modification of the basic PSO framework, this paper proposes an approach to addressing the outlined problem from the view of physical principles.

In this paper, we introduce a physics-inspired reformulation of the PSO algorithm based on the concepts of Hamiltonian and Newtonian mechanics. It allows: 1) modelling the swarm motion from the starting position to the target area; 2) incorporating meaningful principles of energy and forces acting on agents; 3) avoiding collisions and 4) quantifying the agents' energy usage in PSO-like multi-agent motion. The developed approach bridges the gap between abstract swarm intelligence and real-world multi-agent motion. It provides the foundation for coordinated, stable, and collision-free robotic behaviour.

*Corresponding author's e-mail: oleh.sinkevych@lnu.edu.ua
Article received 2026-03-17; accepted 2026-04-08

©2026 Author(s). This is an open access article licensed under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0).

Since the adaptation of PSO principles to real-world multi-agent movement requires their revision, in the following sections the proposed methodology and motion model are considered in detail.

II. Literature Review

The Particle Swarm Optimisation, firstly introduced in a seminal paper [9], describes the motion of dimensionless particles towards the extremum of objective function one wants to minimise. Being inspired by the behaviour of bird flocks, this algorithm has undergone exhaustive research and a significant number of modifications [10]–[12]. Its applications to numerical optimisation, automation control [13], [14], machine learning [15], [16] and robotics [17], [18] exhibit notable efficiency and adaptability in solving diverse applied problems. Among these challenges, the use of PSO as a coordination and guidance mechanism for the single robotic systems and artificial swarms composed of multiple agents is of particular interest [19], [20].

In many scenarios, the movement of a group of robots is set based on a leader–follower strategy, which is reflected in publications [21]–[23]. Here, as well as in [24], [25], PSO is mainly used as an optimisation routine to calculate the optimal formation configuration or PID controller parameters and does not act as a direct motion strategy.

As agent motion requires collision-free operation, numerous studies have addressed this issue. For instance, the paper [26] proposes a PSO-based population learning method that trains a neural-network controller in an unsupervised manner, where the network outputs the left and right wheel velocities of the robots. More advanced approach, which utilises Q-learning, repulsion potential, and PSO to compute optimal trajectory for twin-robots, is presented in [27]. The papers [28], [29] present particle swarm optimisation for multi-robot/UAV path planning, where PSO generates collision-free waypoint sequences (later smoothed with Bézier curves in [28]) and is extended in [29] to coordinated 4D planning with inter-agent separation and synchronized arrival time. Another approach, but designed for single robot motion, is outlined in [30], where obstacle-aware movement is handled via a penalty function. For multi-robot systems, the AERPSO method [31] employs PSO directly as a cooperative guidance mechanism, which enables robots to disperse for environment exploration, avoid obstacles, and subsequently converge around detected targets. Another example of integrating the repulsive force into the PSO algorithm to avoid obstacles is given in [32], where the authors consider the collision-free motion of a single agent in a 2-dimensional map with simple obstacles.

Despite the interesting results in the reviewed publications, the most PSO approaches do not explain the physical meaning of the algorithm parameters and are applied in a more abstract sense to build a valid path-finding approach. In this field, there are a small number of publications aimed at the physical interpretation of PSO [33]. In [34], authors proposed a complex approach, which incorporates PSO, fuzzy logic, and Hamiltonian modelling to represent UAV dynamics. Also, the basics of physical-inspired PSO formulation is presented in [35], where the physical interpretation of the algorithm is outlined. In addition, the reinforcement learning method [36], which incorporates Hamiltonian structure on the policy representation for distributed multi-robot problems, is studied. Thermodynamic approach with PSO, which is proposed in [37] where personal best update is used for obstacle avoidance, is an excellent example of a physical modification that can be used to model multi-agent systems. Since currently there are relatively few experiments with physical interpretations of PSO, in this paper we explore the development of a physical approach that will be useful for researchers of physically inspired multi-agent systems.

Based on the conducted literature overview, a main unresolved limitation of existing algorithms is the lack of a unified, physically grounded framework that consistently integrates interaction potentials with agent dynamics (mass, energy consumption, and damping), which allows for the development of collision-free multi-agent motion models.

III. Hamiltonian-inspired Particle Swarm Optimisation for Agents' Motion

Reformulating the PSO algorithm to take into account physical concepts such as mass, momentum, acceleration, forces, and energy requires a clear review of the principles of agent interaction.

To achieve this, several stages of derivation are performed. First, a Hamiltonian function [38] is defined to describe the total mechanical energy of an agent as the sum of potential and kinetic energies. Next, the conservative forces acting on the agent are derived from the instantaneous potential to govern its motion: (a) attraction to the local best and global (swarm) positions and (b) inter-agent repulsion based on a potential function. To account for energy losses during motion, a Rayleigh dissipation function is introduced, through which non-conservative forces are defined. After that, the standard iterative PSO equations for calculating velocity and displacement vectors are reformulated based on the derived forces, agent mass, and momentum. To assess the effectiveness of the proposed approach, a metric is introduced that accounts for energy losses during motion relative to the approach to a given target.

A. Hamiltonian-inspired Energy Formulation

To adapt the particle swarm dynamics within a physically consistent framework, a Hamiltonian-inspired formulation of the multi-agent system is introduced. An agent is assumed to be a rigid object with a circular shape of radius r and m. The agent has a position $\mathbf{x}_i(t)$ at time t and moves in two-dimensional Euclidean space with velocity $\mathbf{v}_i(t)$.

Let $H_i(t)$ be the Hamiltonian function that encapsulates the total mechanical energy of an individual agent i

$$H_i(t) = K_i(t) + U_i(t). \tag{1}$$

For each agent $H_i(t)$ is defined as the sum of its kinetic energy $K_i(t)$, determined by the agent mass and velocity, and the potential energy $U_i(t)$. The latter is primary explained through

the idealized linear springs that represent agent's attraction to the personal best-known $\mathbf{x}_{i,\,\text{best}}(t)$ and the swarm's best-known $\mathbf{l}_{\text{best}}(t)$ positions. The decision about 'best' positions is made based on Euclidean proximity to the target.

$$d_i(t) = \left\| \mathbf{x}_i(t) - \mathbf{x}_{\text{target}} \right\|_2 . \tag{2}$$

Since the best-known positions are changed in time, the Hamiltonian function is time-dependent. Therefore, it does not correspond to a conservative mechanical system. In general, these positions are chosen based on the value of a given fitness function, which may include other useful metrics.

Here, the kinetic energy is expressed as

$$K_i(t) = \frac{\left\| \mathbf{p}_i(t) \right\|^2}{2m_i}, \tag{3}$$

where $\mathbf{p}_i = m_i \boldsymbol{v}_i$ is the momentum, m_i is the agent's mass and $\boldsymbol{v}_i$ is the agent's velocity.

The potential energy term is given by

$$U_i(t) = \frac{1}{2} c_1 \left\| \mathbf{x}_i(t) - \mathbf{x}_{i,\text{best}}(t) \right\|^2 + \frac{1}{2} c_2 \left\| \mathbf{x}_i(t) - \mathbf{l}_{\text{best}}(t) \right\|^2, \tag{4}$$

where c_1 and c_2 [kg/s^2] are the coefficients that act as spring constants, modelling the attraction to the best-known positions as forces from idealized linear springs, $\mathbf{x}_i = (x_1, x_2)$ is the agent's position in 2D space.

To prevent agents from collisions, we introduce the repulsion potential R_i,

$$R_i(t) = \sum_{j=1, i \neq j}^{N} \frac{c_3}{\left(\left\| \mathbf{x}_i(t) - \mathbf{x}_j(t) \right\|^2 + \varepsilon^2 \right)^{\frac{n}{2}}}, \tag{5}$$

where c_3 [$kg \cdot m^{2+n} / s^2$] is the scaling coefficient that defines the strength of interaction, the exponent $n > 0$ controls the steepness of the repulsive force, N is the total number of agents. A small parameter $\varepsilon > 0$ is introduced to avoid singular repulsive forces at short inter-agent distances. The proposed repulsion term stands for the short-range force that decays with distances between the agents.

By defining the total potential as $V_i(t) = U_i(t) + R_i(t)$, a force on the agent is derived as the negative gradient of total potential $V_i(t)$ as

$$\mathbf{F}_i(t) = -\nabla V_i(t), \tag{6}$$

where

$$\nabla V_i(t) = c_1 \left(\mathbf{x}_i(t) - \mathbf{x}_{i,\text{best}}(t) \right) + c_2 \left(\mathbf{x}_i(t) - \mathbf{l}_{\text{best}}(t) \right) - \\ - n c_3 \sum_{j=1, i \neq j}^{N} \frac{\mathbf{x}_i(t) - \mathbf{x}_j(t)}{\left(\left\| \mathbf{x}_i(t) - \mathbf{x}_j(t) \right\|^2 + \varepsilon^2 \right)^{\frac{n}{2}+1}}. \tag{7}$$

The resulting gradient incorporates "cognitive", "social" and repulsion forces acting on the i-th agent at time t. To model velocity-dependent energy dissipation for each agent, we define a Rayleigh dissipation function for i-th agent as

$$f_i(t) = \frac{1}{2} \gamma \left\| \mathbf{v}_i(t) \right\|^2, \tag{8}$$

where $\gamma > 0$ is the damping coefficient. The resulting non-conservative damping force that represents energy loss due to friction, air resistance or other velocity-dependent damping effects is given as

$$\mathbf{F}_i^{\text{damp}}(t) = -\frac{\partial f_i}{\partial \mathbf{v}_i(t)} = -\gamma \mathbf{v}_i(t). \tag{9}$$

Based on (6) and (9), the total force acting on agent i is defined as

$$\mathbf{F}_{i,\text{total}}(t) = \mathbf{F}_i(t) + \mathbf{F}_i^{\text{damp}}(t) + \xi_i(t), \tag{10}$$

where $\xi_i(t)$ is the zero-mean Gaussian noise with variance proportional to the time step.

B. Velocity Update Rule

In generic PSO implementation, the updates of velocity vectors and particles' positions at each time step are performed as follows:

$$\mathbf{v}_i(t + \Delta t) = w \mathbf{v}_i(t) + c_1 r_1 \left(\mathbf{x}_{i,\text{best}}(t) - \mathbf{x}_i(t) \right) + \\ + c_2 r_2 \left(\mathbf{l}_{\text{best}}(t) - \mathbf{x}_i(t) \right), \tag{11}$$

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t + \Delta t) \Delta t, \tag{12}$$

where w is the inertia factor that allows the agent to retain the portion of its velocity from the previous step, r_1, r_2 are random variables in the range [0, 1] that introduce stochasticity into the system. Unfortunately, these equations are simplified and do not fully capture the physical nature of agent motion.

Another important consideration is the order, in which velocities and positions are updated. For example, (a) agents may first compute their velocity vectors in parallel using the previous values and then simultaneously update their positions; (b) agents may update velocities and positions sequentially agent by agent. The choice of update strategy can significantly influence swarm behaviour and convergence. Parallel updates are computationally efficient but may introduce synchronisation errors due to the simultaneous use of outdated information. In contrast, sequential updates more accurately describe the immediate interaction with higher computational load.

In this study, sequential updates are used, where forces are computed in a step-wise manner based on the current positions within the Hamiltonian-inspired framework. Sequential updating ensures that each agent "sees" the latest position of others. By setting Δt as the time interval between two

subsequent steps equals to a sufficiently small value, the system effectively approximates a continuous-time dynamics. In case of $\Delta t \to 0$, simulation approaches quasi-parallel updates of agents' velocities and positions. Fully parallel updates offer computational efficiency for a large number of agents. However, since all agents respond to slightly outdated states, this may introduce collisions and overlaps.

Based on these considerations, motion dynamics of the proposed approach is outlined as follows:

$$\mathbf{v}_i\left(t+\Delta t\right)=\mathbf{v}_i\left(t\right)+\mathbf{F}_{i,\text{total}}\left(t\right)\Delta t/m_i, \tag{13}$$

$$\mathbf{x}_i\left(t+\Delta t\right)=\mathbf{x}_i\left(t\right)+\frac{\mathbf{v}_i\left(t\right)+\mathbf{v}_i\left(t+\Delta t\right)}{2}\Delta t, \tag{14}$$

where (13) incorporates all forces act on agent i and updates the new velocity vector, (14) changes the agent position.

The presented model corresponds to a second-order mass-spring-damper system. This formulation describes motion under impedance control and artificial potential-field navigation. In this interpretation, attraction terms act as virtual springs toward desired positions. Rayleigh damping represents effective velocity-dependent dissipation. While the resulting system is non-conservative due to time dependency, damping, and stochastic perturbations, the Hamiltonian formulation provides a solid framework for force derivation and energy-aware performance evaluation.

C. Evaluation Metric

To study swarm motion, we introduce a swarm-level metric F that measures how efficiently the swarm moves toward the target. As the proposed model includes second-order equation and explicit energy dissipation, this metric evaluates how effectively the swarm's mechanical energy is translated into progress toward the goal.

Specifically, for each agent i the cumulative mechanical energy dissipated due to Rayleigh damping over the simulation horizon T is given by

$$E_i=\sum_{t=0}^{T-1}\gamma\left\|\mathbf{v}_i\left(t\right)\right\|^2\Delta t, \tag{15}$$

where γ is the damping coefficient. This term accounts for irreversible energy loss caused by friction and related resistive effects.

To evaluate task-related progress without assuming a predefined target-reaching time or a known terminal potential, we define the progress achieved by agent i as the maximum reduction in its Euclidean distance to the target along the trajectory

$$\Delta d_i=d_i\left(0\right)-\min_{0\le t\le T}d_i\left(t\right),\quad d_i\left(t\right)=\left\|\mathbf{x}_i\left(t\right)-\mathbf{x}_{\text{target}}\right\|_2. \tag{16}$$

The only distance-based fitness $d_i(t)$ is used to update best-known positions in (7), which pushes the swarm toward the target during motion.

Using these quantities, the per-agent energetic efficiency is defined as

$$F_i=\frac{E_i}{\Delta d_i+\delta},\ \delta>0, \tag{17}$$

where δ is a small constant to ensure numerical stability. This ratio has an interpretation as energy dissipated per unit of achieved progress, penalizing inefficient or oscillatory trajectories. A swarm-level measure is then obtained by averaging over all agents as

$$F=\frac{1}{N}\sum_{i=1}^{N}F_i. \tag{18}$$

D. Swarm Motion Algorithm

A combination of the steps covered in previous sub-sections allows us to formulate a complete algorithm for the motion of a swarm system.

Given the two-dimensional map defined as $\mathbf{W}=\{(x_1, x_2)\}$, where x_1, x_2 are coordinates in metric units, the initial position $\mathbf{x}_{\text{start}}$, target location $\mathbf{x}_{\text{target}}$, the swarm of N agents is randomly generated inside the circle area O_{start}, $O_{\text{start}}=\{\mathbf{x}|\ \|\mathbf{x}-\mathbf{x}_{\text{start}}\|\le R_s\}$, where R_s is the area's radius. All agents are circular bodies with radius r_i and unit mass m_i=1. They are initially placed in a non-collision manner. Based on the motion rules described in this section, N agents must coordinate their movement to reach a neighbourhood around the target without colliding with one another. To achieve this, the agents' motion is governed by a modified Hamiltonian-inspired PSO algorithm, where best-known positions are updated using the distance-based fitness $d_i(t)$. If, during the motion, two or more agents collide with each other, they are removed from the run. The simulation continues until all surviving agents cluster around the target within a certain radius R_{target}.

To set the ranges of parameters such as Δt, γ, c_1, c_2, c_3 and N, we consider the following assumptions: a) the discrete-time integration step Δt is chosen within the set {0.01, 0.05, 0.1}; b) the attraction coefficients c_1 and c_2, which corresponds to virtual linear spring forces, for unit-mass agents are set within the range [20, 50] that assumes the agent's response time on the order of approximately one second; c) the exponent n that controls the localization of repulsion is set within the range [2,5]; d) parameter c_3, which is chosen within the set {300, 500, 700, 900, 1100}, is relatively comparable with the attraction terms given the admissible force $f_{\max}=20$; e) the damping coefficient γ is chosen in the range [9, 16] to provide near-critically damped motion. The algorithm is outlined below.

Step 1. N agents with mass m_i, size r_i are initialized randomly at $\mathbf{x}_i(0)$ positions within the spawn radius R_s in non-collision manner. $\mathbf{x}_{i,\text{best}}(0)$ are set equal to $\mathbf{x}_i(0)$, $\mathbf{l}_{\text{best}}(0)$ is determined on the basis of target distance. Dissipation coefficient γ, attraction coefficients c_1, c_2, repulsion exponent n and weight c_3, integration step Δt, target position $\mathbf{x}_{\text{target}}$ are specified as well as the neighbourhood radius R_{target} around $\mathbf{x}_{\text{target}}$. The latter is used as the agent's hold-out boundary, i.e., as soon as the agent reaches this neighbourhood, it is stopped there and affects only

$\mathbf{l}_{best}$ swarm value. List $\mathbf{L}_{act}$ of active agents, which holds the non-collided particles, is initialized.

Step 2. For each time step t we perform the agent-by-agent updates from list $\mathbf{L}_{act}$:

Step 2.1. Get i-th agent's neighbours $\mathbf{N}_i$ (*which are all swarm agents in current simulation*);

Step 2.2. Calculate total force acting on i-th agent via (7), (9), and (10);

Step 2.3. Calculate agent's velocity $\boldsymbol{v}_i(t)$ and position $\mathbf{x}_i(t)$ vectors via (13, 14);

Step 2.4. Obtain distance $d_i(t)$ to the target by (16.2) and reduction distance Δd_i by (16.1);

Step 2.5. Based on $d_i(t)$ update personal best position $\mathbf{x}_{i,\text{best}}(t)$, and energy efficiency value F_i using (17);

Step 2.6. If $d_i(t) \le R_{\text{target}}$, then particle is marked as "stopped" and does not move anymore;

Step 2.7. Calculate the best global swarm $\mathbf{l}_{best}(t)$ position;

Step 2.8. If the i-th agent tunnels through or collides with other agents, all of them are removed from $\mathbf{L}_{act}$ list.

Step 3. If all remaining agents reach the target area, i.e., all their positions $\mathbf{x}_i(t)$ are within R_{target}, then the simulation loop is completed.

IV. Numerical Experiments and Results

To study the proposed physics-inspired algorithm, we conducted the numerical simulations for different groups of parameters. The implementation and evaluation of the algorithm's efficiency and performance were carried out in Python 3 programming language using open-source libraries.

To describe and analyse the results of simulations, the following metrics are tracked alongside the average dissipated cumulative mechanical energy and swarm-level averaged energetic efficiency F: a) overlapped and broken agents (overlaps count and fraction, time of the first overlap); and b) swarm cohesion over time, which is defined as

$$C(t) = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_i(t) - \bar{\mathbf{x}}(t)\right\|, \tag{19}$$

where $\bar{\mathbf{x}}(t)$ is the swarm's centre of masses; c) safety margin tracker which is based on

$$g_{ij}(t) = \left\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\right\| - \left(r_i + r_j\right), \tag{20}$$

$$g_{\min}(t) = \min_{i<j} g_{ij}(t). \tag{21}$$

First, for the different number $N = [10, 20, 30]$ of agents with a size of $r_i = 1$ we set attraction coefficients $c_1 = c_2 = 20$, dissipation coefficient $\gamma = 10$, the maximum admissible force $f_{\max} = 20$ to be fixed. For all setups, the size of map $\mathbf{W}$ is 200 × 200 units, $\mathbf{x}_{\text{start}} = (15, 15)$, $\mathbf{x}_{\text{target}} = (190, 185)$. Then, we investigate the effect of the time step $\Delta t = [0.01, 0.05, 0.1]$, which governs the distance agents travel in each iteration. For each value of Δt, we perform $K = 5$ independent runs with identical initial agent configurations. Within each run, we systematically vary the repulsion weight $c_3 = [300, 500, 700, 900, 1100]$ and repulsion coefficient $n = [2, 3]$ to examine their effect on inter-agent repulsion behaviour. The results of such experiments are presented in the following sections.

A. Motion Dynamics for N=10 Agents

The first part of the experiments concerns a configuration of 10 agents. After five runs, the aggregated results are shown in Table I. Here, E_{med} is the median value of E across all runs, gap_{med} is the median value of the minimum gap between agents and coh_{med} is the median swarm cohesion within the runs, $\bar{v}_{\text{med}}$ is the median average velocity magnitude, and n_{steps} is the number of steps.

As can be seen from Table I, increasing the time step Δt leads to a large reduction in the total number of simulation steps n_{steps}, which are perfectly correlated with the number by which the steps increase.

For a fixed value of Δt, with the increase of repulsion via c_3 median value E_{med} tends to decrease. As the repulsion between agents increases, their velocities tend to decrease, as confirmed by the values in column $\bar{v}_{\text{med}}$. The median velocities remain similar due to the limitation of maximum allowed force $f_{\max}$.

The values of gap_{med} continue to satisfy the trend in the data: stronger repulsion increases the distance between agents, and as n increases, the safety gap decreases. These effects are also directly reflected in the behaviour of the median swarm cohesion coh_{med}, which increases with c_3 due to enhanced swarm expansion. Moreover, for all the tested values of Δt and c_3, using a higher repulsion exponent $n = 3$ results in substantially improved cohesion compared to $n = 2$. The median values of F_{med} lie in a narrow range across all configurations, taking values approximately between 17.0 and 18.5. For fixed Δt and n, variations in c_3 induce only modest changes in F_{med}, with no consistent monotonic trend.

To choose the best combination, one can consider various selection criteria. For example, based on the safety-prioritized selection criterion, the configuration with $\Delta t = 0.05$, $c_3 = 1100$, and $n = 2$ is the most optimal setting. This is due to collision-free behaviour, the largest median safety gap, and the lowest median energy among the collision-free configurations.

Another approach for optimal configuration is to use the Pareto optimisation (Fig. 1). Here, we minimise E_{med} and maximise gap_{med}. The selected optimality criteria allow us to select the most suitable parameter configurations, which are mostly responsible for the repulsion effect.

The agents' trajectories on the map resulting from the simulation with $\Delta t = 0.05$, $c_3 = 1100$, and $n = 2$ are shown in Fig. 2. Here, no collisions or tunnelling effects occurred, and all agents successfully reached the target area in 3302 steps.

TABLE I
SIMULATION RESULTS FOR $N = 10$ BY Δt, c_3, AND n

		E_{med}		gap_{med}		coh_{med}		$\bar{v}_{\text{med}}$		n_{steps}	
Δt	c_3	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$
0.01	300	4307	4259	1.56	1.03	6.89	4.71	1.56	1.6	16832	16513
0.01	500	4252	4259	1.79	1.43	8.03	5.33	1.54	1.59	16944	16448
0.01	700	4257	4250	2.01	1.64	8.98	5.79	1.53	1.59	17022	16524
0.01	900	4243	4262	2.31	1.94	9.66	6.14	1.51	1.58	17594	16638
0.01	1100	4207	4262	2.24	2.03	10.15	6.45	1.48	1.58	17858	16829
0.05	300	4167	4244	1.39	1.01	6.94	4.79	1.57	1.62	3342	3135
0.05	500	4088	4191	1.79	1.37	8.27	5.42	1.56	1.61	3233	3147
0.05	700	4050	4152	2.39	1.56	9.25	5.88	1.55	1.61	3223	3153
0.05	900	4033	4096	2.67	1.67	10.04	6.26	1.54	1.6	3288	3141
0.05	1100	4024	4072	3.04	1.21	10.69	6.58	1.53	1.54	3227	3178
0.1	300	4349	4423	0.95	0.69	7.23	4.95	1.59	1.67	1663	1574
0.1	500	4169	4424	1.76	1.2	8.53	5.51	1.59	1.63	1562	1597
0.1	700	4115	4404	1.63	0.99	9.38	5.91	1.55	1.63	1647	1631
0.1	900	4118	4387	2.59	1.16	9.97	6.27	1.54	1.61	1629	1613
0.1	1100	4155	4301	2.27	1.19	10.54	6.6	1.52	1.56	1688	1640

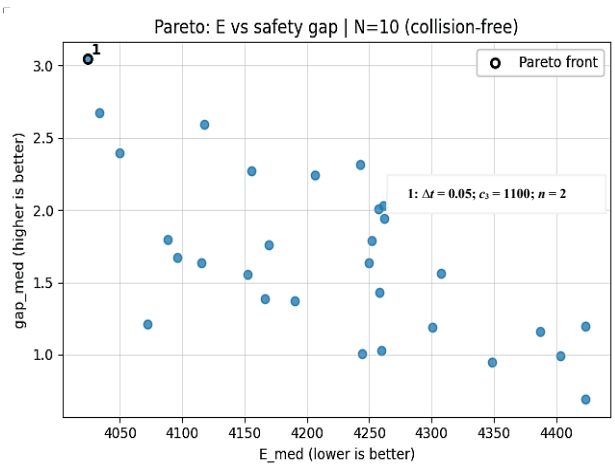


Fig. 1. Pareto front: energy efficiency E_{med} vs. safety gap gap_{med}.

B. Motion Dynamics for N = 20 Agents

The next experiments were conducted for configuration of 20 agents. Five independent runs produced results provided in Table II. With the increase of Δt, the number of steps n_{steps} decreases synchronously, which is similar to the results with ten agents.

For fixed Δt the increase in c_3 reduces high-velocity avoidance bursts in dense swarms (larger N), so E_{med} becomes nearly monotone even when median speed $\bar{v}_{\text{med}}$ is not.

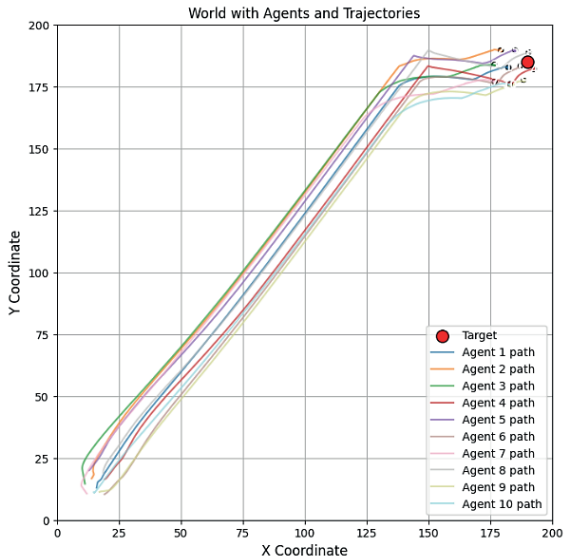


Fig. 2. Agents' motion for $N = 10$, $\Delta t = 0.05$, $c_3 = 1100$, $n = 2$.

The increase in values of gap_{med} demonstrates the effect of stronger repulsion c_3, while the increase in n reduces its impact. Although the general trend in the increase of the value of gap_{med} is preserved, it can be seen that a larger number of agents introduces a perturbation that violates the clear monotonicity. In contrast, the cohesion values coh_{med} do not exhibit such sensitivity and are aligned with the behaviour of ten agents.

For $N = 20$, the median efficiency values F_{med} remain within a relatively narrow range of approximately 17.9–19.5 across all tested configurations. It indicates that variations in c_3 and n produce only moderate changes in energetic efficiency due to the effect of increased repulsion, which mainly redistributes motion geometry and dissipation patterns without substantially altering the overall energy–progress balance of the swarm.

Across all parameter combinations, the distributions of E_{med} have mean ± standard deviation equal to 4217 ± 113 for $N = 10$ and 4439 ± 114 for $N = 20$, with median values of 4244 and 4443, respectively. This corresponds to an increase of approximately 5 % in the mean between $N = 10$ and $N = 20$, while the variance remains nearly unchanged.

Averaged over all values of Δt and c_3, the mean values of gap_{med} for $n = 2$ decrease from 2.02 ($N = 10$) to 1.09 ($N = 20$), and for $n = 3$ from 1.34 to 0.52, corresponding to reductions of approximately 46 % and 62 %, respectively.

The mean values of coh_{med} increase with swarm size: for $n = 2$ from 8.97 ($N = 10$) to 13.62 ($N = 20$), and for $n = 3$ from 5.77 to 8.72. Increasing the repulsion exponent from $n = 2$ to $n = 3$ for $N = 20$ reduces average swarm spread by approximately 36 %, indicating improved cohesion. The median average speed $\bar{v}_{med}$ remains nearly constant due to the force limit f_{max}, with mean values 1.54 ($n = 2$) and 1.60 ($n = 3$) for $N = 10$, and 1.47 and 1.62 for $N = 20$.

Because of the denser swarm, we observed a higher fraction of collided agents. For $c_3 = 300$ in all runs, the fraction of broken agents reached 10 %, which signifies weak strength of repulsion for such configuration. Based on the Pareto approach, there is a single optimal setup for $N = 20$: $\Delta t = 0.1$, $c_3 = 1100$, $n = 2$. The depiction of motion for this configuration is presented in Fig. 3, where all agents have reached the target area in 1856 steps.

TABLE II

SIMULATION RESULTS FOR $N = 20$ BY Δt, c_3, AND n

		E_{med}		gap_{med}		coh_{med}		$\bar{v}_{med}$		n_{steps}	
Δt	c_3	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$
0.01	300	4431	4493	0.58	0.25	10.90	7.12	1.50	1.57	16975	16101
0.01	500	4388	4479	0.88	0.50	12.68	7.99	1.50	1.59	16742	15932
0.01	700	4339	4447	1.10	0.72	13.93	8.60	1.47	1.53	17136	16668
0.01	900	4322	4418	1.19	0.40	14.91	9.20	1.47	1.59	17297	15835
0.01	1100	4314	4409	1.64	0.73	15.96	9.62	1.50	1.63	16815	15859
0.05	300	4567	4540	0.46	0.17	10.92	7.33	1.51	1.63	3485	3154
0.05	500	4438	4534	0.91	0.51	12.79	8.28	1.51	1.62	3302	3215
0.05	700	4353	4616	1.09	0.61	14.03	8.90	1.49	1.57	3491	3295
0.05	900	4274	4497	1.03	0.91	14.75	9.22	1.37	1.64	3652	3116
0.05	1100	4211	4524	1.34	0.95	15.63	9.92	1.45	1.65	3545	3136
0.1	300	4551	4652	0.43	0.19	11.19	7.40	1.58	1.69	1677	1497
0.1	500	4423	4504	1.09	0.23	12.59	8.73	1.40	1.63	1815	1537
0.1	700	4323	4547	1.30	0.37	13.59	9.32	1.44	1.67	1730	1516
0.1	900	4309	4511	1.55	0.56	14.65	9.75	1.37	1.64	1819	1526
0.1	1100	4210	4557	1.71	0.68	15.75	10.15	1.45	1.64	1683	1597

C. Motion Dynamics for N = 30 Agents

The last five independent experiments were completed for $N = 30$ agents (Table III). An obvious effect is provided here: as the integration step Δt increases, the number of iterations n_{steps} of the swarm movement decreases. Moreover, experiments for different N showed that between ten and thirty agents during movement, no significant change in the number of iterations is observed.

The increase of c_3 for a fixed Δt generally corresponds to a reduction of the median energy dissipation E_{med} averaged over the swarm for $N = 30$. Across all parameter combinations, the distributions of E_{med} have mean ± standard deviation equal to 4410±117 for $N = 30$, with a median value of 4425.

This corresponds to a decrease of approximately 0.7 % in the mean between $N = 20$ and $N = 30$, indicating that the energy dissipation distributions remain numerically very close for both swarm sizes. Compared with $N = 10$, the mean value for $N = 30$ is higher by approximately 4.6 %, which confirms the small increase of total dissipation with swarm size.

The median safety gap continues to decrease with increasing swarm size and repulsion exponent. Averaged over all values of Δt and c_3, the mean values of gap_{med} for $n = 2$ decrease from 1.09 for $N = 20$ to 0.95 for $N = 30$, while for $n = 3$ they decrease from 0.52 to 0.37, respectively.

This corresponds to an additional reduction of approximately 12 % for $n = 2$ and 28 % for $n = 3$ between $N = 20$ and $N = 30$. Relative to $N = 10$, the total reduction in the mean gap reaches approximately 53 % for $n = 2$ and 72 % for $n = 3$, which confirms that larger swarms and higher n produce consistently smaller and less variable inter-agent safety gaps.

For $N = 30$, the mean values of coh_{med} equal 16.01 for $n = 2$ and 10.74 for $n = 3$. Increasing the repulsion exponent from $n = 2$ to $n = 3$ reduces the average swarm spread by approximately 33 %. This indicates improved cohesion for higher repulsion. Compared with $N = 20$, the mean cohesion increases from 13.62 to 16.01 for $n = 2$ and from 8.72 to 10.73 for $n = 3$, which confirms the growth of swarm spatial extent. Relative to $N = 10$, the mean cohesion increases by approximately 44 % for $n = 2$ and 47 % for $n = 3$. This demonstrates the consistent scaling of swarm dispersion with system size.

TABLE III

SIMULATION RESULTS FOR $N = 30$ BY Δt, c_3, AND n

		E_{med}		gap_{med}		coh_{med}		$\bar{v}_{\text{med}}$		n_{steps}	
Δt	c_3	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$	$n = 2$	$n = 3$
0.01	300	4427	4514	0.43	0.09	13.23	8.96	1.46	1.62	17205	15787
0.01	500	4339	4411	0.88	0.38	15.12	10.04	1.44	1.56	17326	15945
0.01	700	4408	4471	0.80	0.35	16.70	10.96	1.49	1.51	16913	16232
0.01	900	4291	4423	1.08	0.40	17.54	11.37	1.43	1.59	17204	15990
0.01	1100	4201	4361	1.16	0.41	18.19	12.03	1.41	1.58	17557	15543
0.05	300	4470	4565	0.49	0.19	13.56	9.12	1.57	1.64	3190	3044
0.05	500	4385	4442	0.86	0.34	15.18	9.92	1.48	1.63	3352	3086
0.05	700	4354	4509	0.79	0.48	16.15	11.06	1.38	1.56	3613	3197
0.05	900	4226	4496	1.29	0.52	17.13	11.40	1.36	1.50	3623	3370
0.05	1100	4195	4464	1.93	0.45	17.57	12.00	1.29	1.56	3844	3195
0.1	300	4440	4628	0.27	0.16	13.50	9.51	1.60	1.63	1583	1538
0.1	500	4376	4610	0.55	0.32	15.24	10.34	1.51	1.60	1643	1546
0.1	700	4320	4541	1.09	0.49	16.27	10.85	1.43	1.66	1729	1522
0.1	900	4194	4532	1.46	0.46	17.04	11.45	1.37	1.36	1803	1553
0.1	1100	4263	4449	1.23	0.57	17.74	12.16	1.36	1.62	1820	1542

The distribution of the normalized energy function F_{med} is approximately equal to experiments with $N = 10$ and $N = 20$, and remains within a narrow range [18, 19.5]. It indicates that energetic efficiency is invariant with respect to swarm size. This effect is explained by the relative rectilinear movement of the swarm to the target, the absence of obstacles, and the strong cohesion of the agents.

With increasing the number of agents, the number of collisions also increases. For $N = 30$ majority of experiment with $c_3 = 300$ resulted in collisions of 14 % agents. This effect indicates that within dense swarm, which is moving towards the target area, repulsion strength of $c_3 = 300$ is not enough to preserve collision-free motion.

Pareto criteria suggest that the best configurations for $N = 30$ are $\Delta t = 0.1$, $c_3 = 900$, $n = 2$ and $\Delta t = 0.05$, $c_3 = 1100$, $n = 2$.

All three experiments for different N show that considering motion safety and minimum energy, configurations with $n = 2$ and high repulsion strength $c_3 = 900$ and $c_3 = 1100$ are the most preferable. Motion trajectories for $N = 30$ and $\Delta t = 0.05$, $c_3 = 1100$, $n = 2$ are presented in Fig. 4, where all the agents have reached the target area in 3619 steps.

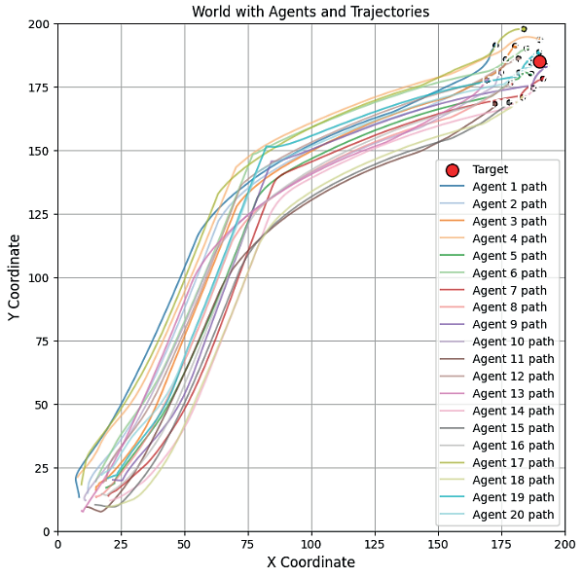


Fig. 3. Agents motion for $N = 20$, $\Delta t = 0.1$, $c_3 = 1100$, $n = 2$.

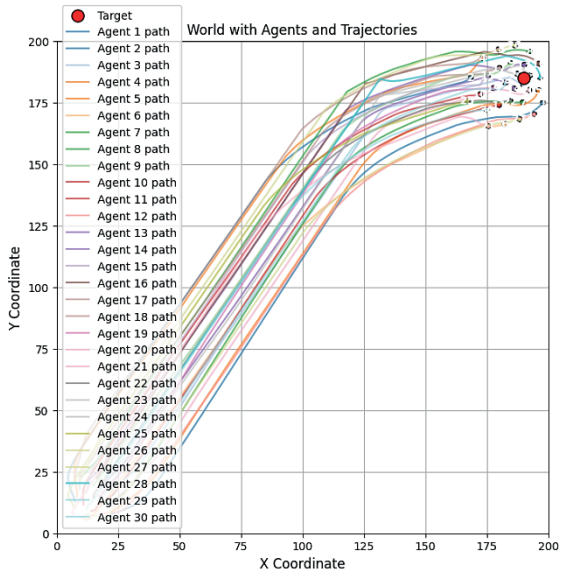


Fig. 4. Agents' motion for $N = 30$, $\Delta t = 0.05$, $c_3 = 1100$, $n = 2$.

D. Comparison with PSO Approach

To compare the proposed approach with standard PSO, we need to agree on key parameters between the methods. The inertia coefficient w is derived from the damping dynamics

$$\frac{\partial \mathbf{v}}{\partial t} = -\gamma \mathbf{v}, \tag{22}$$

$$\mathbf{v}(t+\Delta t) = e^{-\frac{\gamma}{m}\Delta t}\mathbf{v}(t) \Rightarrow \; w = e^{-\frac{\gamma}{m}\Delta t}. \tag{23}$$

Since in our algorithm the force acting on the agent is limited to fmax, to incorporate this behaviour into standard PSO we bound the velocity change per step:

$$\left\|\mathbf{v}(t+\Delta t) - \mathbf{v}(t)\right\| \le \frac{f_{\max}\Delta t}{m_i}. \tag{24}$$

All fifteen experiments for $\Delta t = 0.01$ and $\Delta t = 0.1$ were conducted for $N = 30$, $c_1 = c_2 = 20$, $\gamma = 10$, $r_1 = r_2 = 1$. As it is seen in Table IV, for $\Delta t = 0.01$ all agents collided with each other during 15 simulations, which were stopped on average at 451 steps. Median cohesion value is comparably low, which is caused by absence of repulsion. Similar behaviour is observed for $\Delta t = 0.1$, i.e., all the agents collided, but in a smaller number of averaged steps due to a relatively larger integration step.

The results of these experiments demonstrate the fact that without including the repulsive potential in the standard formulation, it is impossible to avoid collisions between agents even by reducing the integration step.

Other comparative experiments were conducted to evaluate the proposed approach (noise term was set to zero) against the absence of repulsion term (see Table V).

Here, the average number of steps when all the agents collided, are comparatively larger, which indicates an interesting outcome. Incorporating the second-order formulation against the first-order equation extends the life of agents in the simulation. When $\Delta t = 0.01$, the number of steps differs in more than 100 in favour of the proposed approach. Although there is no significant difference in the number of steps for larger $\Delta t = 0.1$, the resulting effects are explained by more accurate physical modelling of the swarm's movement.

TABLE IV
PSO RESULTS FOR $N = 20$

	F_{med}	E_{med}	coh_{med}	n_{steps}
$\Delta t = 0.01$	26	57	4.3	451
$\Delta t = 0.1$	38	110	4.3	40

TABLE V
HAMILTONIAN-INSPIRED PSO RESULTS FOR $N = 20$

	F_{med}	E_{med}	coh_{med}	n_{steps}
$\Delta t = 0.01$	24	68	4.1	585
$\Delta t = 0.1$	27	69	4.8	42

The results of these comparisons indicate that without including the repulsion term, PSO-based approaches for modelling the motion of a multi-agent system do not ensure collision-free swarm movement.

V. CONCLUSIONS

This study has introduced a Hamiltonian-inspired reformulation of the particle swarm optimisation algorithm for coordinated multi-agent motion in continuous space. By incorporating agent mass, momentum, attractive and repulsive potentials, and Rayleigh dissipation, the proposed model defines physically consistent swarm system with the energy interpretation and collision avoidance.

The numerical experiments have demonstrated that the inclusion of distance-based repulsive potential ensures stable and collision-free motion within specific parameter regimes. In particular, dense swarms require sufficiently strong repulsion coefficients and smaller exponent n to prevent overlaps and tunnelling effects. Across all tested swarm sizes $N = 10$–30,

configurations with high repulsion strength provided the most reliable configuration. To demonstrate the proposed approach, we have used the set of fixed parameters c_1, c_2, γ, which affect the agents' trajectories. Nevertheless, in all the cases agents reach the target area.

The proposed energy-based efficiency metric and safety indicators have allowed for the evaluation of swarm performance. The provided results show that normalized energetic efficiency stays nearly invariant with respect to swarm size. This can be explained by the swarm's cohesion and synchronous movement towards the target, during which the repulsive force causes only minor oscillations of the trajectory. Safety gap and cohesion increase with increasing agent density. The latter can be fixed by intensifying repulsion strength c_3. These outcomes confirm that repulsion scaling is the dominant factor for maintaining collision-free motion in dense swarms. The proposed metrics enable multi-criteria (e.g., based on Pareto optimisation) selection of parameters using safety, cohesion, or energy-oriented objectives. It should also be noted that the influence of the γ energy dissipation coefficient requires further research, as it directly affects the dynamics of agent motion.

Although the study has considered only motion without obstacles, the developed Hamiltonian-inspired formulation provides an approach for developing swarm navigation with potential fields around obstacles. The proposed approach establishes a physically interpretable bridge between classical PSO coordination mechanisms and realistic multi-agent robotic motion.

Future research will focus on model development to include the direct comparison of first- and second-order formulation with repulsion terms, Lagrangian-inspired model, dynamic obstacles, different agent masses and sizes, and three-dimensional motion. Experimental validation on embedded systems and comparison with other physics-based swarm techniques will be conducted in further research.

REFERENCES

[1] A. Nitti, M. D. de Tullio, I. Federico, and G. Carbone, "A collective intelligence model for swarm robotics applications," *Nat. Commun.,* vol. 16, Jul. 2025, Art. no. 6572. https://doi.org/10.1038/s41467-025-61985-7

[2] A. Ayali and G. A. Kaminka, "The hybrid bio-robotic swarm as a powerful tool for collective motion research: a perspective," *Frontiers in Neurorobotics*, vol. 17, Jul. 2023. https://doi.org/10.3389/fnbot.2023.1215085

[3] A. P. Engelbrecht, *Computational Intelligence. An Introduction,* 2nd ed. John Wiley & Sons, Ltd., 2007. https://dai.fmph.uniba.sk/courses/ICI/engelbrecht.comp-intel-intro.07.pdf

[4] G. G. Rigatos, "Multi-robot motion planning using swarm intelligence," *Int. J. Adv. Robotic Syst.*, vol. 5, no. 2, Jan. 2008, Art. no. 23. https://doi.org/10.5772/5601

[5] E. H. Cui, Z. Zhang, C. J. Chen, and W. K. Wong, "Applications of nature-inspired metaheuristic algorithms for tackling optimisation problems across disciplines," *Scientific. Rep.*, vol. 14, Apr. 2024, Art. no. 9403. https://doi.org/10.1038/s41598-024-56670-6

[6] X.-S. Yang, "Nature-inspired optimisation algorithms: Challenges and open problems," *J. Comput. Sci.*, vol. 46, Oct. 2020, Art. no. 101104. https://doi.org/10.1016/j.jocs.2020.101104

[7] G. Rossides, B. Metcalfe, and A. Hunter, "Particle swarm Optimisation – An adaptation for the control of robotic swarms," *Robotics*, vol. 10, no. 2, Apr. 2021, Art. no. 58. https://doi.org/10.3390/robotics10020058

[8] M. G. M. Hamami and Z. H. Ismail, "A systematic review on particle swarm optimisation towards target search in the swarm robotics domain," *Arch. Comput. Methods Eng.*, Oct. 2022. https://doi.org/10.1007/s11831-022-09819-3

[9] J. Kennedy and R. Eberhart, "Particle swarm optimisation," in *ICNN'95 – International Conference on Neural Networks*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948. https://doi.org/10.1109/icnn.1995.488968

[10] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimisation," in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. Springer Berlin Heidelberg, 1998, pp. 591–600. https://doi.org/10.1007/BFb0040810

[11] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimisation: Theory, analysis, and application," *Swarm and Evolutionary Computation*, vol. 63, June 2021, Art. no. 100868. https://doi.org/10.1016/j.swevo.2021.100868

[12] A. G. Gad, "Particle swarm optimisation algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, pp. 2531–2561, Apr. 2022. https://doi.org/10.1007/s11831-021-09694-4

[13] A. Gurko and Y. Petrenko, "A PSO-based controller tuning for a laser technical vision system," in *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)*, Kharkiv, Ukraine, Oct. 2022, pp. 1–5. https://doi.org/10.1109/KhPIWeek57572.2022.9916393

[14] M. G. Skarpetis, N. D. Kouvakas, F. N. Koumboulis, and M. Tsoukalas, "PSO-based robust control of SISO systems with application to a hydraulic inverted pendulum," *Eng.*, vol. 6, no. 7, Jul. 2025, Art. no. 146. https://doi.org/10.3390/eng6070146

[15] E. Cengil, "The power of machine learning methods and PSO in air quality prediction," *Applied Sciences*, vol. 15, no. 5, Feb. 2025, Art. no. 2546. https://doi.org/10.3390/app15052546

[16] D. E. Ghersi, N. E. Mougari, K. Loubar, M. Amoura, and U. Desideri, "A machine learning and Particle Swarm Optimisation approach for desiccant wheel modeling and performance prediction," *Applied Thermal Engineering*, vol. 284, Jan. 2026, Art. no. 129084. https://doi.org/10.1016/j.applthermaleng.2025.129084

[17] Y. Zhang, N. Li, Y. Chen, and Y. Liu, "A mobile robot path planning method based on dynamic multipopulation particle swarm optimisation," *Journal of Robotics*, Dec. 2024. https://doi.org/10.1155/joro/5018491

[18] B. Bratina *et al*., "Mobile robot localization based on the PSO algorithm with local minima avoiding the fitness function," *Sensors*, vol. 25, no. 20, Oct. 2025, Art. no. 6283. https://doi.org/10.3390/s25206283

[19] M. B. Aremu, G. Ahmed, S. Elferik, and A. W. A. Sarif, "Autonomous mobile robot path planning techniques – A review: metaheuristic and cognitive techniques," *Robotics*, vol. 15, no. 1, Jan. 2026, Art. no. 23. https://doi.org/10.3390/robotics15010023

[20] J. Ni, Y. Zhao, Z. Zhang, C. Ke, and S. X. Yang, "A survey on theories and applications for multi-robot cooperative hunting," *Robotics and Autonomous Systems*, vol. 197, Mar. 2026, Art. no. 105296. https://doi.org/10.1016/j.robot.2025.105296

[21] N. M. Kwok, V. T. Ngo, and Q. P. Ha. "PSO-based cooperative control of multiple mobile robots in parameter-tuned formations," in *2007 IEEE International Conference on Automation Science and Engineering*, Scottsdale, AZ, USA, Sep. 2007, pp. 332–337. https://doi.org/10.1109/COASE.2007.4341716

[22] X. Wang, D. Yang, and S. Chen, "Particle swarm optimisation based leader-follower cooperative control in multi-agent systems," *Applied Soft Computing*, vol. 151, Jan. 2024, Art. no. 111130. https://doi.org/10.1016/j.asoc.2023.111130

[23] N. Gomez *et al.*, "Leader-follower behavior in multi-agent systems for search and rescue based on PSO approach," in *SoutheastCon 2022*, Mobile, AL, USA, Mar.–Apr. 2022, pp. 413–420. https://doi.org/10.1109/SoutheastCon48659.2022.9764133

[24] O. Boutalbi, F. Seghir, A. Boutalbi, and L. Guerra, "A PSO-based global path planning approach for mobile robots," in *2024 12th International Conference on Systems and Control (ICSC)*, Batna, Algeria, Nov. 2024, pp. 354–359. https://doi.org/10.1109/ICSC63929.2024.10928795

[25] E. A. Aner, M. I. Awad, and O. M. Shehata, "Performance evaluation of PSO-PID and PSO-FLC for continuum robot's developed modeling and control," *Scientific Reports*, vol. 14, Jan. 2024. https://doi.org/10.1038/s41598-023-50551-0

[26] A. Ayari and S. Bouamama, "A new multiple robot path planning algorithm: dynamic distributed particle swarm optimisation," *Robotics and Biomimetics*, vol. 4, Nov. 2017, Art. no. 8. https://doi.org/10.1186/s40638-017-0062-6

[27] B. Sahu, P. K. Das, and M. R. Kabat, "Multi-robot cooperation and path planning for stick transporting using improved Q-learning and democratic robotics PSO," *Journal of Computational Science*, vol. 60, Apr. 2022, Art. no. 101637. https://doi.org/10.1016/j.jocs.2022.101637
[28] Y.-L. Poy *et al*., "Enhanced particle swarm optimisation for multi-robot path planning with Bézier curve smoothing," *Robotics*, vol. 13, no. 10, Sep. 2024, Art. no. 141. https://doi.org/10.3390/robotics13100141
[29] Y. Liu, X. Zhang, Y. Zhang, and X. Guan, "Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach," *Chinese Journal of Aeronautics*, vol. 32, no. 6, pp. 1504–1519, June 2019. https://doi.org/10.1016/j.cja.2019.03.026
[30] K. K. Pandey *et al*., "Trajectory planning and collision control of a mobile robot: A penalty-based PSO approach," *Mathematical Problems in Engineering*, Jan. 2023. https://doi.org/10.1155/2023/1040461
[31] V. Garg, A. Shukla, and R. Tiwari, "AERPSO – An adaptive exploration robotic PSO based cooperative algorithm for multiple target searching," *Expert Systems with Applications*, vol. 209, Dec. 2022, Art. no. 118245. https://doi.org/10.1016/j.eswa.2022.118245
[32] A. Benmachiche, M. Derdour, M. S. Kahil, M. C. Ghanem, and M. Deriche, "Adaptive hybrid PSO-APF algorithm for advanced path planning in next-generation autonomous robots," *Sensors*, vol. 25, no. 18, Sep. 2025, Art. no. 5742. https://doi.org/10.3390/s25185742
[33] S. M. Mikki and A. A. Kishk, *Particle Swarm Optimisation: A Physics-Based Approach*. Springer International Publishing, 2008. https://doi.org/10.1007/978-3-031-01704-9
[34] E. Can, "Energy-aware adaptive altitude aontrol of UAVs via fuzzy-PSO optimisation within a port-hamiltonian framework under icing and sensor noise," *International Journal of Aeronautical and Space Sciences*, pp. 2552–2568, Nov. 2025. https://doi.org/10.1007/s42405-025-01087-2
[35] O. Sinkevych, B. Sokolovskyy, Y. Boyko, and Z. Matchyshyn, "Physics-informed particle swarm optimisation for collision-aware swarm navigation," *Advances in Cyber-Physical Systems*, vol. 10, no. 2, pp. 197–201, Nov. 2025. https://doi.org/10.23939/acps2025.02.197
[36] E. Sebastián, T. Duong, N. Atanasov, E. Montijano, and C. Sagues,"Physics-informed multiagent reinforcement learning for distributed multirobot problems," *IEEE Transactions on Robotics*, vol. 41, pp. 4499–4517, June 2025. https://doi.org/10.1109/tro.2025.3582836
[37] K. Bojappa and J. Lee, "Thermodynamic particle swarm optimisation for multi-agent system in unknown environment," *IFAC-PapersOnLine*, vol. 59, no. 30, pp. 461–466, 2025. https://doi.org/10.1016/j.ifacol.2025.12.280
[38] J. R. Mohallem, *Lagrangian and Hamiltonian Mechanics*. Cham, Switzerland: Springer International Publishing, May 2025.

Oleh Sinkevych received a PhD degree in Computer Science in 2023 at Ivan Franko National University of Lviv, Ukraine. Since 2023, he has been working as an 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 optimisation, and metaheuristics.
E-mail: oleh.sinkevych@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0001-9834-2700

Bohdan Sokolovskii received a Candidate in Physical and Mathematical Sciences degree in 1987 at Ivan Franko National University of Lviv, Ukraine. Since 2014, he has been working as an Associate Professor at the Faculty of Electronics and Computer Technologies of Ivan Franko National University of Lviv. His research interests include electronics, semiconductor physics, swarm AI algorithms, mathematical modelling, and metaheuristics.
E-mail: bohdan.sokolovskyy@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0002-2561-3255

Yaroslav Boyko received a Candidate in Physical and Mathematical Sciences degree in 2017 at Ivan Franko National University of Lviv, Ukraine. Since 2017, he has been working as an Associate Professor at the Faculty of Electronics and Computer Technologies of Ivan Franko National University of Lviv. His research interests include embedded programming, machine learning, swarm AI, and physical modelling.
E-mail: yaroslav.boyko@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0001-5967-6345

Igor Olenych received the degree of Candidate in Physical and Mathematical Sciences from the Institute for Physical Optics, Ukraine in 2010 and the degree of Doctor of Sciences in Physics and Mathematics from Ivan Franko Lviv National University, Ukraine in 2020. Since 2021, he has been working as the Head of Radioelectronic and Computer Systems Department of the Faculty of Electronics and Computer Technologies of Ivan Franko National University of Lviv. His research interests include nanoelectronics, nanostructures and nanosystems, fuzzy modelling and applied physics.
E-mail: igor.olenych@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0002-6642-0222