# POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Matematica


## Tesi di Laurea Magistrale

# Particle Swarm Optimization model: description of the movement of pedestrians in evacuation scenarios

**Relatore:**
Prof. Marco Scianna

**Candidata:**
Arianna Coppola

**Correlatrice:**
Prof.ssa Annachiara Colombi

**Anno Accademico 2024-2025**

# Summary

# 1 Introduction

## 1.1 Definition of the problem

Understanding and simulating the behavior of pedestrian crowds, particularly in emergency scenarios, presents a significant challenge. Human movement in these situations is highly dynamic and influenced by a variety of factors, including environmental conditions, individual goals, local interactions, and limited information. These interactions often lead to emergent collective behaviors, such as bottlenecks, herding, or congestion waves, which are difficult to predict using centralized or static models.

Historically, pedestrian dynamics have been studied only through empirical approaches that rely on observation and data collection. Although useful for identifying average behaviors or response trends, these methods lack predictive power, especially in complex or evolving environments. More recently, traditional empirical methods have increasingly been integrated and interfaced with computational approaches derived from applied physics, mathematics, and engineering.

Physics-based models typically treat individuals as passive particles with fixed goals and predetermined trajectories. As a result, they often struggle to capture the dynamic and locally informed decision-making that characterize real pedestrian behavior, particularly in emergencies, where goals and constraints change rapidly.

To overcome these limitations, there has been growing interest in agent-based and optimization-inspired models that treat pedestrians as autonomous agents capable of making decisions sensitive to context. In these models, pedestrian behavior can be viewed as a decentralized decision-making process in which individuals continuously evaluate their surroundings and adjust their movements accordingly. This perspective suggests that pedestrian dynamics, particularly in evacuation contexts, can be framed as an optimization problem: each individual seeks to minimize a personal cost function while navigating a constrained, dynamic environment filled with obstacles, other agents, and evolving dangers such as fire or smoke.

This mathematical formulation captures pedestrian behavior as a process of continuous adaptation. Each agent updates its position in response to changing conditions in an effort to minimize perceived cost. However, classical optimization methods are ill-suited to the decentralized, dynamic, and multi-agent nature of such problems.

To address this, the present work adopts and extends a Particle Swarm Optimization (PSO) framework.

## 1.2 Problem formalization

Under this framework, evacuation behavior can be modeled as a numerical optimization problem. In general, optimization refers to the process of minimizing (or maximizing) a function while satisfying some constraints on its variables.

A numerical optimization problem can be formally defined as $\mathcal{P} = (\mathbf{X}, \mathcal{F})$, where:

- $\mathbf{X}$ represents the continuous set of feasible solutions, commonly referred to as the search space;
- $\mathcal{F}(\mathbf{x}) : \mathbf{X} \to \mathbb{R}$ is the objective function, which assigns a cost or fitness value to each element $\mathbf{x} \in \mathbf{X}$.

The goal of optimization is to identify the set of optimal solutions $\mathbf{X}^*$, defined as:

$$\mathbf{X}^* = \arg\min_{\mathbf{x} \in \mathbf{X}} \mathcal{F}(\mathbf{x}) \subseteq \mathbf{X}. \tag{1}$$

This implies that a solution $\mathbf{x}^*$ belongs to $\mathbf{X}^*$ if and only if it minimizes the objective function $\mathcal{F}$. For simplicity, we assume that the search space $\mathbf{X}$, which is also the domain of $\mathcal{F}$, is a bounded region in the $d$-dimensional real space, specifically a hyperrectangular domain (box), such that $\mathbf{X} \subseteq \mathbb{R}^d$, with $d \in \{1, 2, 3\}$.

For the sake of completeness, in the case of maximization problems, $\mathbf{X}^*$ contains solutions that maximize $\mathcal{F}$. However, we hereafter consider the minimization one, as defined in Eq. (1).

## 1.3 Particle swarm optimization algorithm

There are three main categories of optimization methods: exact methods. heuristics and metaheuristics. Exact methods are algorithms that guarantee finding an optimal solution to a problem, given sufficient time and resources. These include techniques like Branch-and-Bound, Dynamic Programming, and Integer Programming. They are best suited for small to medium-sized problems due to their high computational complexity.

Heuristics are problem-specific strategies designed to find good, though not necessarily optimal, solutions in a reasonable amount of time. They are based on domain knowledge and predefined rules. Examples include Greedy algorithms and Nearest-Neighbor methods.

Metaheuristics, on the other hand, are high-level frameworks that guide underlying heuristics in exploring the solution space. They are problem-independent and are often inspired by natural or physical processes.

Common examples include Genetic Algorithms, Simulated Annealing, and Particle Swarm Optimization.

Particle Swarm Optimization (PSO) is a population-based stochastic optimization algorithm inspired by the collective intelligence of animal swarms, such as bird flocks and fish schools. Introduced by Kennedy and R. Eberhart, 1995, PSO has undergone significant refinements to address various optimization challenges. As a metaheuristic approach, PSO is particularly effective in solving complex, multidimensional optimization problems where traditional deterministic algorithms fall short. It operates on probabilistic transition rules, leveraging social cooperation and evolutionary behaviors to efficiently explore solution spaces. Unlike Genetic Algorithms and Differential Evolution, PSO does not require derivative information, making it suitable for both continuous and discrete optimization tasks, including constrained and unconstrained problems.

The PSO algorithm operates as a swarm-based search, where the position of each individual, known as particle, represents a potential solution in the $d$-dimensional search space. Each particle retains memory of both its own best position and the best position found by the swarm, updating its velocity according to a prescribed set of rules. This process continues until convergence (i.e., when all particles reach the same position), an optimal solution is found, or computational limits are reached. A flow diagram of the algorithm is shown in Fig. 1.

## 1.4 Canonical version of the PSO

In PSO-based methods, a swarm consists of $N$ point particles. Each of them moves within the search space $\mathbf{X}$, aiming to find the minimum (or the minima) of the objective function $\mathcal{F}$. In this respect, the positions of the swarm agents in $\mathbf{X}$ can be interpreted as possible solutions to the optimization problem. Here specifically, any particle $i = 1, ..., N$, for any iteration $t \in T \subset \mathbb{N}_+ \cup \{0\}$, is characterized by the following state:

$$\mathbf{s}_i(t) = (\mathbf{x}_i(t), \mathbf{v}_i(t), \mathbf{p}_i(t)) \in \mathbf{X} \times \mathbb{R}^d \times \mathbf{X} \tag{2}$$

where:

- $\mathbf{x}_i(t)$ is its current position;
- $\mathbf{v}_i(t)$ is its velocity;
- $\mathbf{p}_i(t)$ is a memory vector that store the best position found by the $i$-th particle so far.

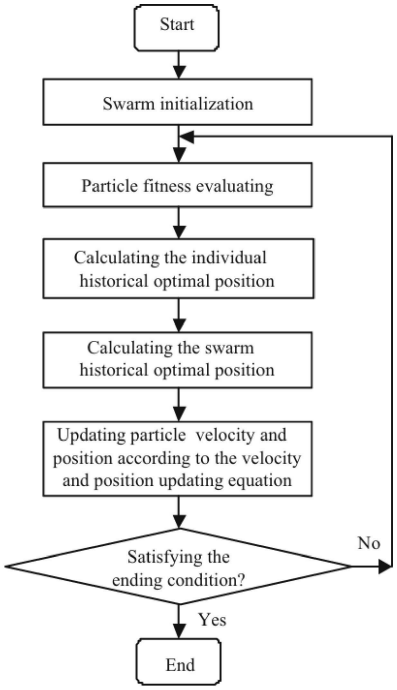


Figure 1: Flowchart of the PSO algorithm from Wang et al., 2018.

The sequence of states acquired over time by the $i$-the agent gives its history, and the sequence of positions gives its trajectory. The particle update mechanism in PSO integrates four key factors:

1. Inertia: the tendency of the particle to maintain its previous velocity;
2. Cognitive component: the influence of the particle's own past best position;
3. Social component: the effect of the best known position within the swarm;
4. Randomness.

Mathematically, the dynamics of the representative $i$-th particle follow these equations, for any time step $t \in T$:

$$\bar{\mathbf{x}}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{3}$$

$$\mathbf{v}_i(t{+}1) = \underbrace{m\mathbf{v}_i(t)}_{\mathbf{v}_i^{\text{in}}(t+1)\text{: inertia}} + \underbrace{c_1\mathbf{R}_1(t+1)(\mathbf{p}_i(t) - \mathbf{x}_i(t))}_{\mathbf{v}_i^{\text{cogn}}(t+1)\text{: cognitive component}} + \underbrace{c_2\mathbf{R}_2(t+1)(\mathbf{p}_g(t) - \mathbf{x}_i(t))}_{\mathbf{v}_i^{\text{soc}}(t+1)\text{: social component}} \tag{4}$$

where:

- $\mathbf{p}_g$ is the global best position actually found by the entire swarm;
- $m \in [0, 1]$, $c_1$ and $c_2$ are positive constants;
- $\mathbf{R}_1(t+1), \mathbf{R}_2(t+1) \in \mathbb{R}^{d \times d}$ are diagonal matrices with randomly distributed values, introducing stochastic variation into the movement.

The position is eventually corrected using a boundary operator $\overline{\partial \mathbf{X}}$, i.e.:

$$\mathbf{x}_i(t+1) = \begin{cases} \bar{\mathbf{x}}_i(t+1), & \text{if } \bar{\mathbf{x}}_i(t+1) \in \mathbf{X}; \\ \overline{\partial \mathbf{X}}(\bar{\mathbf{x}}_i(t+1)), & \text{otherwise.} \end{cases} \tag{5}$$

This ensures that the particles remain within valid search boundaries, as $\overline{\partial \mathbf{X}}$ maps a point $\bar{\mathbf{x}}_i \notin \mathbf{X}$ to one that falls within $\mathbf{X}$.

At each iteration, the memory vector of $i$-th agent is updated based on function evaluation:

$$\mathbf{p}_i(t+1) = \begin{cases} \mathbf{p}_i(t), & \text{if } \mathcal{F}(\mathbf{x}_i(t+1)) > \mathcal{F}(\mathbf{p}_i(t)); \\ \mathbf{x}_i(t+1), & \text{if } \mathcal{F}(\mathbf{x}_i(t+1)) \leq \mathcal{F}(\mathbf{p}_i(t)); \end{cases} \tag{6}$$

with initial condition $\mathbf{p}_i(0) = \mathbf{x}_i(0)$, and $\mathcal{F}$ denoting the objective function to be minimized. Then the following chain rule is verified:

$$\mathcal{F}(\mathbf{p}_i(0)) \geq \mathcal{F}(\mathbf{p}_i(1)) \geq ... \geq \mathcal{F}(\mathbf{p}_i(t)) \tag{7}$$

for any step $t \in T$.

The best global position is determined as:

$$\mathbf{p}_g(t) = \underset{i=1,...,N}{\arg\min} \{\mathcal{F}(\mathbf{p}_i(t))\}, \forall t \in T, \tag{8}$$

starting from

$$\mathbf{p}_g(0) = \arg\min_i \{\mathcal{F}(\mathbf{p}_i(0))\} = \arg\min_i \{\mathcal{F}(\mathbf{x}_i(0))\}. \tag{9}$$

Thus, each particle not only leverages its own experiences but also integrates information from its group mates, enabling the swarm to explore the search space more efficiently.

The following section provides an analysis of the previously described version of the PSO and its parameters.

**Analysis of the velocity components**

- **Inertia component**
  The initial formulation of the PSO algorithm, called original PSO, did not involve an inertia term ($m = 1$). Although computationally simple, this approach was not very effective in optimization problems as it lacked mechanisms to effectively balance exploration and exploitation. To address this, an inertia weight $m \neq 1$ was introduced into the velocity update formula, significantly improving the algorithm performance with minimal additional complexity. This modification led to what is commonly called the canonical PSO.

  The introduction of the inertia weight can be mathematically interpreted by analyzing the effective forces acting on each particle. In particular, for any $t \in T$ and $i = 1, ..., N$, we define

  $$\mathbf{f}_i(t+1) := \mathbf{v}_i^{\text{cogn}}(t+1) + \mathbf{v}_i^{\text{soc}}(t+1) \tag{10}$$

  and refer to $\mathbf{f}_i$ as the resulting external force acting on the $i$-th particle, where the cognitive and social component reflect the pull towards the particle personal best and the global best positions, respectively. The resulting effective acceleration can be expressed as:

  $$\Delta\mathbf{v}_i(t+1) = \mathbf{v}_i(t+1) - \mathbf{v}_i(t) = \mathbf{f}_i(t+1) - (1-m)\mathbf{v}_i(t). \tag{11}$$

  In this formulation, the term $(1 - m)$ acts as a friction coefficient, effectively capturing the viscosity of the environment through which the particles move. Consequently, the inertia weight and the friction coefficient are dependent. Higher values of $m$, that is, lower viscosity and less friction, promote a more extensive exploration of the search space. On the other hand, smaller values of $m$ lead to a more dissipative system, promoting exploitation.

- **Cognitive and social components**
  In PSO, cognitive and social components represent internal and external influences on the trajectory of each particle. The cognitive term $\mathbf{v}_i^{\text{cogn}}$ reflects the self-confidence of the particle in its best position previously found, whereas the social term $\mathbf{v}_i^{\text{soc}}$ embodies the attraction towards the best solution found by the swarm. These components can be interpreted as attractive forces produced by springs with varying stiffness over time. Specifically, $(c_1\mathbf{R}_1)/2$ and $(c_2\mathbf{R}_2)/2$ determine the pull towards $\mathbf{p}_i$ and $\mathbf{p}_g$, respectively. In this analogy, $c_1/2$ and $c_2/2$ represent the average stiffness of these springs. Consequently, adjusting

these parameters, also known as learning factors, directly impacts the way particles react or agitate: higher values of $c_1$ and $c_2$ lead to stronger attractions, enhancing responsiveness, but risking instability. Thus, the balance between cognitive and social terms fundamentally shapes the ability of the swarm to explore the search space.

In general $c_1$ and $c_2$ are set to be constant in time and equal for all agents. Early implementations of PSO set both coefficients equal, typically $c_1 = c_2 = 2.0$. However, this choice often led to uncontrolled oscillations. Subsequent empirical studies sought better control over swarm dynamics by redefining the choice of $c_1$ and $c_2$. Clerc and Kennedy, 2002, suggested to fix both to approximately 1.49445 to ensure convergence. In contrast, Carlisle and Dozier, 2001, proposed to differentiate the two acceleration coefficients, recommending $c_1 = 2.8$ and $c_2 = 1.3$, emphasizing personal learning over social influence. This setup was later supported by Schutte and Groenwold, 2005, who observed that larger $c_1$ helped maintain diversity in the swarm and reduced the risk of premature convergence by discouraging particles from collapsing too quickly to a single solution, that is, to avoid stagnation in suboptimal solutions.

As with the inertia weight, various PSO variants have explored time-varying strategies for the acceleration coefficients. For example, Ratnaweera et al., 2004, suggested decreasing both $c_1$ and $c_2$ over time, while other works proposed complementary dynamics, such as decreasing $c_1$ while increasing $c_2$, to gradually shift the focus of the swarm from self-exploration to collective refinement. Adaptive approaches using evolutionary state information (Ide and Yasuda, 2005), genetic algorithms (Yu et al., 2005), fuzzy logic systems (Juang et al., 2011), differential evolution (Parsopoulos and Vrahatis, 2002), and reinforcement-based methods such as adaptive critic design (Doctor and Venayagamoorthy, 2005) have also been proposed.

From a sociological point of view, the components of the equation of the velocity update can be interpreted as follows: the inertial and cognitive terms embody individualistic behavior, reflecting the reliance of particle on its current trajectory and past experiences; the social term instead represents collective learning, arising from the exchange of information among group members. This interpretation underlines the hybrid nature of PSO, blending individual self-reliance with group-driven cooperation to achieve efficient global optimization.

- **Velocity initialization and clamping**
  Another critical aspect of PSO performance lies in the initialization and control of particle velocities. If all particles are initialized with zero velocity, their movement depends entirely on cognitive and social terms, limiting exploration. Therefore, proper initialization and regulation of velocity are crucial.

  One of the earliest and most widely used strategies to regulate particle velocity and maintain stable swarm behavior involves applying component-wise velocity clamping. For instance, Kennedy and R. Eberhart, 1995, introduced a velocity clamping technique, bounding each component of $\mathbf{v}_i$ within a predefined range $[-\mathrm{v}_{\max}, +\mathrm{v}_{\max}]$.

  More generally, it can be set the following rule: an upper limit on each dimension of the velocity of the particle, ensuring that

$$|(\mathbf{v}_i(t))_j| = \min\{|(\mathbf{v}_i(t))_j|, \mathrm{v}_{\max}\}, \tag{12}$$

  for any $t \in T$ and $j = 1, \ldots, d$, where $\mathrm{v}_{\max} \in \mathbb{R}_+$ is a predefined threshold.

  The choice of $\mathrm{v}_{\max}$ has a significant impact on the search behavior of the swarm. Large values of $\mathrm{v}_{\max}$ allow for global exploration, while smaller values of $\mathrm{v}_{\max}$ restrict movement, leading to localized exploitative behavior. However, choosing an appropriate value for $\mathrm{v}_{\max}$ is highly problem-specific, and there is no established heuristic to reliably determine it.

  To overcome this limitation, adaptive strategies have been proposed. For example, Fan, 2002, introduced a linearly decreasing $\mathrm{v}_{\max}$ over time to gradually shift the focus of the swarm from broad exploration in the early stages to fine exploitation later in the run. Similarly, Fourie and Groenwold, 2002, proposed to reduce $\mathrm{v}_{\max}$ dynamically, on the basis of the recent success of the swarm, creating a feedback mechanism to balance exploration and exploitation.

  The inertia weight $m$ can also serve as a velocity control parameter. When $m$ is large, the particle achieves higher velocities, thus enhancing the exploration capability. In contrast, smaller values of $m$ constrain the speed of the particle and encourage a more localized search. This can accelerate convergence, but the risk of premature stagnation in local optima increases.

  In parallel with the inertia-based approach, Clerc and Kennedy, 2002, proposed an alternative method for velocity regulation based on the

introduction of a constriction factor $\chi$, modifying the velocity update rule to

$$\mathbf{v}_i(t+1) = \chi[\mathbf{v}_i(t) + \tilde{c}_1\mathbf{R}_1(t+1)(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + \tilde{c}_2\mathbf{R}_2(t+1)(\mathbf{p}_g(t) - \mathbf{x}_i(t))]. \tag{13}$$

The appropriate parameter mapping (i.e. from $(\chi, \tilde{c}_1, \tilde{c}_2)$ to $(m, c_1, c_1)$) renders the constriction and inertia formulations equivalent. However, the constriction factor must be carefully tuned: excessive values of $\chi$ are shown to lead to slow convergence, but high solution accuracy, while too small values accelerate convergence at the expense of solution quality, potentially leading to swarm stagnation Clerc and Kennedy, 2002.

Empirical investigations R. C. Eberhart and Shi, 2000, also show that the combination of inertia weight or constriction factor with velocity clamping, limiting the maximum velocity of the particle, substantially improves the convergence rate and the quality of the solution.

- **Neighborhood topologies and information propagation**
  Particle interaction in PSO is governed by a communication topology, often modeled as a social network. Different topologies, such as fully connected (global best) and localized neighborhoods, shapes how information spreads through the swarm and thus affects the convergence behavior.

  The definition of the global memory vector $\mathbf{p}_g$ implies that there is a transmission of knowledge throughout the entire swarm, as $\mathbf{p}_g$ relies on the information shared by all particles. This option, which gives rise to the so-called global version of the method, has been shown to lead to fast convergence, but to increase the possibility of premature stagnation. To mitigate this limitation, local variants of the PSO were proposed in which each particle communicates only with a limited subset of the swarm. In these cases, each representative particle $i$, has its own "global" best solution vector that depends on the information received by the subgroup of agent it is actually communicated with, defined by the set $\mathcal{N}_i(t)$:

$$\mathbf{p}_{g,i}(t) = \underset{j \in \mathcal{N}_i(t) \cup \{i\}}{\arg\min} \{\mathcal{F}(\mathbf{p}_j(t))\}, \forall t \in T. \tag{14}$$

  The vector $\mathbf{p}_{g,i}(t)$ has then to replace its counterpart $\mathbf{p}_g$, both in the state of the $i$-th particle and in the velocity update rule.

  These local communication structures introduce multiple social attractors, which promotes greater population diversity. As a result,

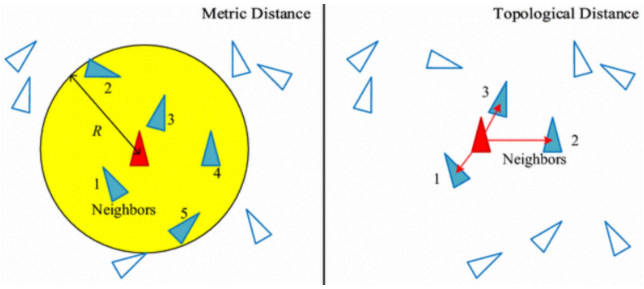


Figure 2: Neighborhoods defined by either the metric distance or topological distance, from Duan et al., 2018.

the speed of convergence decreases, but the possibility of finding global optima improves.

More specifically, the set $\mathcal{N}_i$ can be defined as metric-based (i.e., defined by spatial distance) or topological (i.e., defined by a network structure) neighborhood. Although metric-based methods were explored in early studies, such as in Kennedy and R. Eberhart, 1995, and Heppner and Grenander, 1990, their computational cost and poor convergence properties led to a shift towards topological structures. The difference between these structures is shown in Fig. 2.

Topological networks, usually referred to as a social network, can be static (fixed connectivity) or dynamic (time-varying). Static examples include ring, star, and von Neumann structures, whereas dynamic networks adapt over time based on particle performance or stochastic rules.

In principle, the choice between open (excluding self) and closed (including self) neighborhoods affects local best selection as well. However, studies have indicated that this choice has only a minor influence on overall algorithm behavior (Kennedy and Mendes, 2002, Mendes et al., 2003). Furthermore, Suganthan, 1999, suggested a gradual transition from local to global connectivity to combine early-stage exploration benefits with late-stage convergence advantages.

## 1.5 Limitations of classical PSO

Thanks to its efficiency in solving complex optimization problems, PSO is widely used in machine learning, engineering, and computational intelligence. Advances in computational techniques and GPU technology have further expanded its applications across scientific and engineering domains. However,

ongoing research continues to refine PSO variants to improve robustness and efficiency.

### 1.5.1 Generic limitations

Despite its simplicity, minimal parameter tuning, and intuitive implementation, canonical PSO can suffer from stagnation and premature convergence. Consequently, to address these issues, numerous variants have been developed to improve performance and adaptability.

In its canonical form (see Eq. (4)), each particle updates its velocity based on a combination of inertia, its personal best position (cognitive component), and the global best of the swarm (social component). A key issue in this structure is the use of the global best position $\mathbf{p}_g$, which assumes full information sharing throughout the swarm and continuous attraction towards a fixed, static target.

Moreover, the stochastic elements $\mathbf{R}_1$ and $\mathbf{R}_2$, typically sampled from uniform distributions, are simplistic and do not represent structured or directional uncertainty. Another limitation lies in the inertial term $m\mathbf{v}_i(t)$, which implies a continuous and smooth motion. Although effective in static environments, this behavior limits the responsiveness of the algorithm in rapidly changing or noisy settings.

Lastly, the canonical PSO assumes fixed or slowly evolving parameters and lacks mechanisms for adaptive information or role differentiation between agents, factors that are often critical in complex, dynamic systems. In particular, the coefficients $c_1$ and $c_2$ in Eq. (4) lack a precise physical interpretation, as they merely act as empirical scaling factors for the cognitive and social components. This limits their explanatory power to modeling real-world phenomena. In contrast, in the proposed model introduced in the following chapter, the new coefficients are defined as weights with a clear physical meaning, providing a more interpretable link between the mathematical formulation and the underlying behavior.

### 1.5.2 Limitations in pedestrian evacuation scenarios

These limitations become especially critical when PSO is applied to a dynamic multi-agent system such as pedestrian evacuation modeling. In real-world evacuation scenarios, pedestrians operate with limited information and interact locally. As such, the assumption of a global attractor like $\mathbf{p}_g$ is therefore behaviorally unrealistic and may lead to premature convergence or synchronized motion that neglects spatial awareness and heterogeneity.

This limitation motivates the replacement of $\mathbf{p}_g$ with a local best $\mathbf{p}_{g,i}$,

calculated over a dynamically defined neighborhood $\mathcal{N}_i(t)$, which better reflects visibility constraints, proximity-based interactions, and personal space in crowded environments.

Additionally, in real environments, randomness is influenced by factors such as partial visibility, perceived danger, and crowd density. Thus, these terms should be redefined to encode bounded, directional uncertainty, and heterogeneous environmental awareness (Colombi et al., 2017).

The inertial component $m\mathbf{v}_i(t)$ also poses issues when modeling pedestrian behavior. In contrast with the smooth and continuous motion implied by inertia, pedestrians can stop, reassess, change direction, or start moving again, especially in response to local signals such as fire, smoke, or crowd movement. This behavior suggests that the inertial term should be minimized or removed, providing a more flexible and reactive model that is better suited to simulate sudden changes in motion. This approach is supported by first-order pedestrian models that eliminate inertia to better reflect stop-and-go dynamics.

Furthermore, the canonical PSO does not account for the dual role of walls and obstacles. In real evacuation scenarios, walls can pose danger by creating entrapment risks, but may also serve as navigational aids in low-visibility conditions due to smoke. Modeling this duality requires incorporating context-sensitive interactions between agents and environmental boundaries, where walls may exert repulsive or attractive influence depending on the situation.

Finally, while the canonical PSO assumes fixed or gradually evolving parameters, it lacks mechanisms for adaptive information transfer that reflect human communication. Recent alternative swarm methods, such as Swarm-Based Gradient Descent (SBGD) proposed by Lu et al., 2024, address this by introducing mass redistribution and communication-based adaptation. These models distinguish between heavy agents (stable leaders) and light agents (explorers), allowing more realistic dynamics where behavior evolves through both individual learning and inter-agent communication.

These limitations are especially pronounced in evacuation scenarios, where agents must constantly adapt to changing environments, moving obstacles, and varying goals. The assumptions of static topology, uniform behavior, and full information do not hold in these settings. Therefore, the canonical PSO needs to be extended with mechanisms such as visual and auditory local neighborhoods, context-aware stochasticity, and behaviorally grounded dynamics. The following chapter presents a modified PSO model that incorporates these extensions to more accurately simulate pedestrian evacuation behavior under realistic constraints.

# 2 Proposed model

In order to overcome the limitations of the classical Particle Swarm Optimization model and better adapt the algorithm to simulate pedestrian behavior, this chapter introduces a modified PSO framework. Each of the previously identified limitations of the classical model is now analyzed and re-examined in the context of behavioral realism, and the corresponding algorithmic modifications are proposed to align the model with the observed dynamics of human crowds.

## 2.1 Compact formulation of the modified PSO model

The modified PSO formulation can be expressed in a compact form that generalizes the classical PSO formulation while adapting it to the context of pedestrian dynamics. The proposed model integrates cognitive, social, and stochastic contributions into a single velocity update rule and separates the magnitude and direction of the velocity. This structure mirrors the canonical PSO but introduces a separation of speed and direction, making it more suitable for pedestrian dynamics.

For each agent $i$ at time $t \in T$, the update rule is written as:

$$\mathbf{v}_i(t+1) = \mathrm{v}_i(t+1)\widehat{\mathbf{w}_i(t+1)}, \tag{15}$$

where $\mathrm{v}_i(t+1)$ denotes the speed (that is a scalar quantity bounded above by the maximum admissible velocity $\mathrm{v}_{\max}$), and $\widehat{\mathbf{w}_i(t+1)} = \frac{\mathbf{w}_i(t+1)}{||\mathbf{w}_i(t+1)||}$ is the unit vector indicating the direction of motion. This separation ensures that the model respects physiological speed limits while preserving realistic orientation dynamics.

The direction $\mathbf{w}_i(t+1)$ results from a convex combination of behavioral stimuli:

$$\mathbf{w}_i(t+1) = \sum_{j \in \mathcal{J}_i} \alpha_i^j(t+1)\widehat{\mathbf{w}_i^j(t+1)}, \tag{16}$$

where $\mathcal{J}_i$ is the set of behavioral contributions and the coefficients $\alpha_i^j(t)$ the corresponding adaptive weights that satisfy, for any $t \in T$ and agent $i$:

$$\begin{cases} \alpha_i^j(t) \in [0,1], & \forall j \in \mathcal{J}_i; \\ \sum_{j \in \mathcal{J}_i} \alpha_i^j(t) = 1. \end{cases} \tag{17}$$

In order to maintain consistency with classical PSO variants, we assume that pedestrian movement is dictated by social and cognitive inputs, affected by

random aspects, i.e., $\mathcal{J}_i = \{\text{cogn, soc, rand}\}$ for any $i$. The update rule of the direction of the particle $i$ at time $t \in T$ therefore amounts to:

$$\mathbf{w}_i(t{+}1) = \alpha_i^{\text{cogn}}(t{+}1)\widehat{\mathbf{w}_i^{\text{cogn}}(t+1)} + \alpha_i^{\text{soc}}(t{+}1)\widehat{\mathbf{w}_i^{\text{soc}}(t+1)} + \alpha_i^{\text{rand}}(t{+}1)\widehat{\mathbf{w}_i^{\text{rand}}(t+1)}. \tag{18}$$

The coefficients $\alpha_i^j$ capture the relative importance of each behavioral input and can evolve over time depending on internal factors such as stress and urgency, and external conditions such as visibility and crowding. This convex combination ensures balanced contributions, preventing any single component from dominating the others non-proportionally.

## 2.2 Cognitive component

The cognitive term models the tendency of the agent to move toward its own goal or target. It is defined as:

$$\widehat{\mathbf{w}_i^{\text{cogn}}(t+1)} = \frac{\mathbf{p}_i(t) - \mathbf{x}_i(t)}{||\mathbf{p}_i(t) - \mathbf{x}_i(t)||}, \tag{19}$$

where $\mathbf{p}_i(t)$ is the personal best position associated with the agent $i$, for any $t \in T$.

In canonical PSO, $\mathbf{p}_i$ corresponds to the best position found so far by the particle in terms of the optimization objective. In the pedestrian context, this concept is reinterpreted as an agent-specific desired position, for example, an exit or a safe zone, reflecting the intentional, goal-directed nature of human motion. This reinterpretation grounds the cognitive component in behavioral realism, ensuring that each agent maintains a clear objective.

## 2.3 Social component

A structural limitation of the canonical PSO is the assumption of global interactions, where all particles are influenced by the same best position. Although suitable for abstract optimization tasks, this assumption is unrealistic in the context of pedestrian dynamics, particularly in the evacuation scenarios considered, where obstacles, partial visibility, and perceived danger can affect pedestrian behavior. To address this, neighborhoods are redefined as sets of agents that fall within perceptual ranges. Here, visual neighborhoods are considered.

Before detailed analysis, it is important to define what kind of neighborhood will be used in this formulation. For evacuation modeling, metric neighborhoods seem more realistic, since in emergency situations

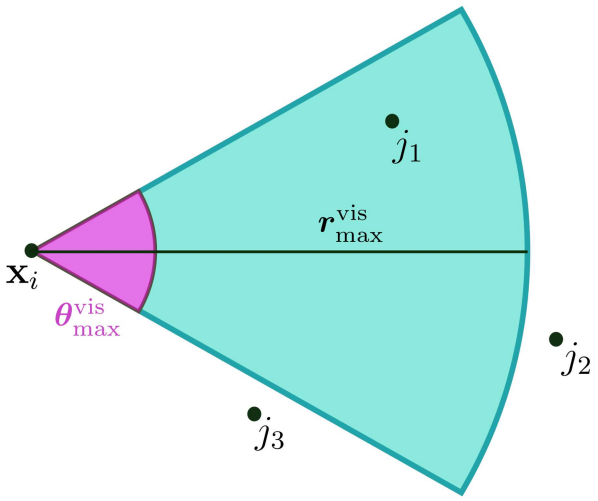


Figure 3: Visual field of particle $i$: $j_1 \in \mathcal{N}_i(t)$, $j_2$, $j_3 \notin \mathcal{N}_i(t)$.

pedestrians tend to react to spatial proximity, not fixed interaction ranks. Consequently, the following formulations adopt the metric approach.

The social component reflects the influence of other agents through a local neighborhood. Unlike the canonical PSO, which assumes global communication, here interactions are restricted to perceptual ranges. The visual neighborhood of agent $i$ at time $t \in T$ is defined as (Fig. 3):

$$\begin{aligned}\mathcal{N}_i(t) = \Bigg\{ & j = 1, ..., N, j \neq i : ||\mathbf{x}_j(t) - \mathbf{x}_i(t)|| \leq r_{\max}^{\text{vis}}, \\ & \arccos\left(\frac{(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \cdot \mathbf{w}_i(t)}{||\mathbf{x}_j(t) - \mathbf{x}_i(t)||}\right) \leq \frac{\theta_{\max}^{\text{vis}}}{2} \Bigg\},\end{aligned} \tag{20}$$

where $r_{\max}^{\text{vis}}$ is the maximum visual range (vision depth) of the individual and $\theta_{\max}^{\text{vis}}$ is the angle of vision. This definition identifies the visual neighborhood as a circular sector centered at the position $\mathbf{x}_i(t)$, with radius $r_{\max}^{\text{vis}}$ and angle $\theta_{\max}^{\text{vis}}$.

The social influence is then based on the local best position $\mathbf{p}_{g,i}$ among these neighbors:

$$\mathbf{p}_{g,i}(t) = \underset{j \in \mathcal{N}_i(t) \cup \{i\}}{\arg\min} \{\mathcal{F}(\mathbf{p}_j(t))\}, \tag{21}$$

yielding the normalized social direction:

$$\widehat{\mathbf{w}_i^{\text{soc}}(t+1)} = \frac{\mathbf{p}_{g,i}(t) - \mathbf{x}_i(t)}{||\mathbf{p}_{g,i}(t) - \mathbf{x}_i(t)||}. \tag{22}$$

This formulation captures perception limits and the directional dependence of vision, leading to realistic emergent effects such as clustering, lane formation, or clogging at exits. It also ensures that interactions remain local and bounded, in contrast to the unrealistic global awareness assumed in the canonical PSO.

## 2.4 Random component

In the canonical formulation of the algorithm, stochasticity is introduced by the random diagonal matrices $\mathbf{R}_1$ and $\mathbf{R}_2$, whose components are typically sampled from a uniform distribution $\mathcal{U}(0,1)$. Although intended to promote exploration, the resulting randomness is isotropic, uniform across all agents, and insensitive to environmental variability.

In pedestrian dynamics, uncertainty is rarely uniform: it is often directional, context-dependent, and heterogeneous, influenced by visibility, crowding, and perceived danger Colombi et al., 2017. To address this, stochasticity is introduced in Eq. (4) as an explicit velocity component, independent of cognitive and social terms:

$$\widehat{\mathbf{w}_i^{\text{rand}}(t+1)} = \frac{\mathbf{r}_i(t+1)}{||\mathbf{r}_i(t+1)||}, \tag{23}$$

where $\mathbf{r}_i(t)$ is a bounded random vector that can encode context-specific variability.

In a first version of the model the random component will be a vector of diffusive type, i.e.

$$\mathbf{r}_i(t+1) = (\cos(\eta_i(t+1)), \sin(\eta_i(t+1)) \tag{24}$$

with $\eta_i \sim U[0, 2\pi), \forall i = 1, ..., N$ and $t \in T$. This formulation allows agents to explore alternative directions and prevents premature alignment. Moreover, it introduces heterogeneity among agents, ensuring that their trajectories remain diverse even under similar conditions.

## 2.5 Advantages over the canonical PSO

The modified formulation differs from the classical PSO in the following key ways:

1. Absence of inertia and explicit random component: the model avoids the rigid momentum effects of the canonical algorithm and incorporates a flexible exploration mechanism. This improves adaptability and reflects the reactive nature of pedestrians, who may stop, reorient, or react suddenly to local stimuli such as crowd congestion, alarms, or smoke;

2. Decoupling of velocity magnitude and direction: the separation of speed from orientation ensures that agents respect physiological velocity limits while maintaining flexibility in their directional choices;

3. Local rather than global interactions: unlike the canonical PSO, where all agents are influenced by the same global best, the modified model restricts interactions to perceptual neighborhoods. This makes the dynamics behaviorally plausible, since pedestrians have limited information and act based on local visibility and proximity, not global knowledge of the environment.

Together, these modifications provide a framework that preserves the strengths of PSO while adapting it to the specific requirements of evacuation dynamics, where reactivity, heterogeneity, and perceptual constraints are critical.

# 3 Numerical results

To assess the performance of the proposed PSO algorithm, it was applied to two standard benchmark functions: the paraboloid (or sphere) function and the Ackley function.

## 3.1 Search space and test functions

All simulations were conducted within a two dimensional square search space $\mathbf{X} = [-L, L]^2$, where the position vector of the particle $i$ is defined as:

$$\mathbf{x}_i(t) = [x_{1,i}(t), x_{2,i}(t)]' \in \mathbf{X}, \forall t \in T. \tag{25}$$

A two-dimensional setup was selected to simplify the visualization of swarm trajectories and reduce the computational cost, while still being adequate to demonstrate the effectiveness of the proposed modification.

To examine the performance of the algorithm across different fitness landscapes, two standard benchmark functions were selected, the sphere and the Ackley functions, representing respectively a unimodal and a multimodal optimization problem.

### 3.1.1 Sphere Function

The sphere function is a classic unimodal function, commonly used to test the convergence efficiency and precision of optimization algorithms. In the two-dimensional case, it is defined as:

$$\mathcal{F}(x_1, x_2) = x_1^2 + x_2^2. \tag{26}$$

This function has a unique global minimum at $\mathbf{x}^* = (0, 0)$, with $\mathcal{F}(\mathbf{x}^*) = 0$, as shown in Fig. 4. Its smooth and convex nature makes it particularly suitable for evaluating the exploitation capacity of the algorithm.

### 3.1.2 Ackley function

The Ackley function is a complex multimodal function employed to test the ability of the algorithm to avoid stagnation in local minima and to maintain a proper balance between exploration and exploitation.

In two dimensions, it is defined as:

$$\mathcal{F}(x_1, x_2) = -20 \exp\left\{-0.2\sqrt{\frac{x_1^2 + x_2^2}{2}}\right\} - \exp\left\{\frac{\cos(2\pi x_1) + \cos(2\pi x_2)}{2}\right\} + 20 + e. \tag{27}$$

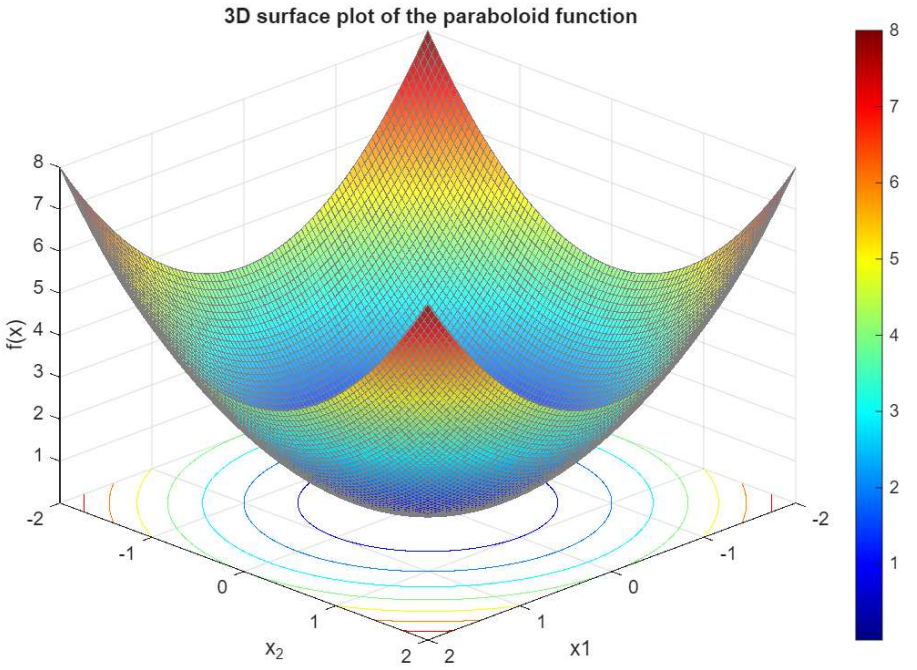


Figure 4: Plot of the sphere function defined in Eq. (26).

This function is characterized by a global minimum in $\mathbf{x}^* = (0, 0)$, with $\mathcal{F}(\mathbf{x}^*) = 0$, and numerous local minima distributed uniformly throughout the domain, as shown in Fig. 5.

This landscape presents a significant challenge for optimization algorithms, which can easily become trapped in suboptimal regions.

### 3.1.3 Boundary conditions

A critical aspect in the simulations concerns the management of particles that attempt to leave the search space $\mathbf{X}$. A reflective boundary condition was implemented to ensure that all particles remain within the domain.

As defined in Eq. (3), the candidate new position is computed as:

$$\bar{\mathbf{x}}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{28}$$

and then corrected according to Eq. (5). In the reflective scheme, $\partial\mathbf{X}$ maps any point $\bar{\mathbf{x}}_i \notin \mathbf{X}$ back into the feasible region by reflection across the violated boundary. This is achieved as follows:

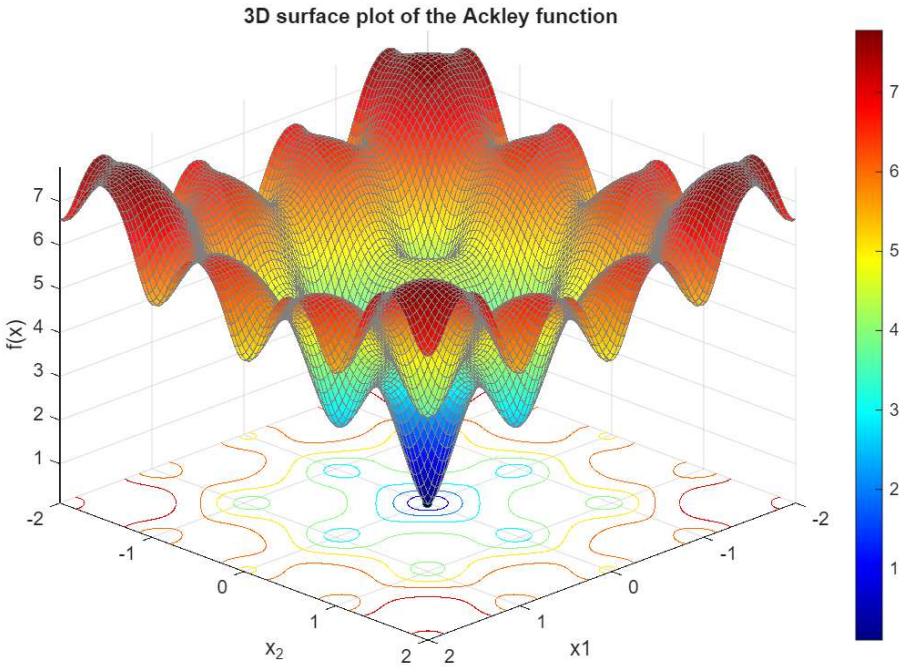


Figure 5: Plot of the Ackley function defined in Eq. (27).

1. Compute the overshoot distance of the particle. For example, if the particle trespasses the right boundary

$$d = \bar{\mathbf{x}}_i(1) - L, \text{ if } \bar{\mathbf{x}}_i(1) > L. \tag{29}$$

2. Reflect the position back into the domain by the same distance, e.g.

$$\bar{\mathbf{x}}_i(1) = L - d, \tag{30}$$

   while maintaining the other component.

3. Invert the corresponding component of the direction vector (that is $\mathbf{w}_i(1) = -\mathbf{w}_i(1)$ in the above example), simulating a physical reflection of the visual field.

The described procedure is repeated properly over the entire boundary until $\bar{\mathbf{x}}_i \in \mathbf{X}$ and then $\mathbf{x}_i(t+1) = \bar{\mathbf{x}}_i(t+1)$.

## 3.2 Classification of the results

The performance analysis focuses on a homogeneous swarm in which all particles have the same level of exploration and exploitation behavior, which is constant in time, i.e. $\alpha_i^{\text{cogn}} = \alpha^{\text{cogn}}$, $\alpha_i^{\text{soc}} = \alpha^{\text{soc}}$, and $\alpha_i^{\text{rand}} = \alpha^{\text{rand}}, \forall i = 1, ..., N$. The aim of our first analysis is to identify the set of weighting parameters $\boldsymbol{\alpha} = (\alpha^{\text{cogn}}, \alpha^{\text{soc}}, \alpha^{\text{rand}})$ that produces the best convergence behavior and solution quality.

Given the stochastic nature of the algorithm, each considered configuration of $\boldsymbol{\alpha}$ was tested over $R$ independent runs, each limited to a maximum of $T_{\max}$ iterations.

For every simulation run $r = 1, ..., R$, the swarm of $N$ particles was initialized as follows:

- initial positions: randomly distributed within the domain $\mathbf{X}$, with $x_{1,i}(0), x_{2,i}(0) \sim U[-2, 2]$;
- initial velocities: random directions, i.e. randomly distributed in $[0, 2\pi)$, with a constant magnitude v = 0.01 across all particles and runs.

The results were evaluated using two types of metrics: success rate indicators and swarm learning analysis.

**Success and convergence rates** The success of convergence is assessed at the final iteration $T_{\max}$, based on the distance of all particles from the true global minimum $\mathbf{x}^*$, within a tolerance $tol$.

For each simulation $r = 1, ..., R$, two performance indicators are computed:

1. Success rate ($SR$):

$$SR = \frac{1}{R} \sum_{r=1}^{R} S_r, \tag{31}$$

where

$$S_r = \mathbb{1}\big( \min_{i=1,...,N} ||\mathbf{p}_i(T_{\max}) - \mathbf{x}^*|| < tol \big)_r, \tag{32}$$

and $\mathbb{1}$ denotes the indicator function.

2. Convergence rate ($CR$):

$$CR = \frac{1}{R} \sum_{r=1}^{R} C_r, \tag{33}$$

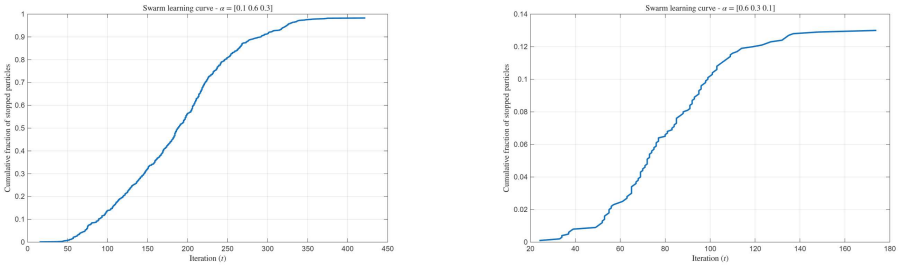


Figure 6: Learning curves of the Ackley function for different sets of $\boldsymbol{\alpha}$. The plots show the cumulative fraction of stopped particles as a function of the iteration $t$.

where

$$C_r = \mathbb{1}\big(\max_{i=1,\dots,N} ||\mathbf{p}_i(T_{\max}) - \mathbf{x}^*|| < tol\big)_r. \tag{34}$$

The success rate measures the fraction of runs in which at least one particle reaches the global minimum, while the convergence rate quantifies the fraction of runs in which all particles converge within the tolerance.

**Swarm learning and stopping analysis** The proposed modification introduces a particle-stopping mechanism, which requires additional performance metrics related to the learning dynamics of the swarm. A status vector of size $N$ is initialized to zero. When the personal best position of one particle $\mathbf{p}_i$ lies within the tolerance of the global minimum (i.e. $||\mathbf{p}_i - \mathbf{x}^*|| < tol$) at a given time $\bar{t} \in T$, its corresponding entry is set to 1. From that moment on, the velocity of the particle is set to zero:

$$\mathbf{v}_i(t) = \mathbf{0},\ \forall t > \bar{t}. \tag{35}$$

This mechanism models the behavior of an agent that, upon reaching the optimum, stops moving and communicates its optimal position to the rest of the swarm. Consequently, its position remains fixed at the optimum for all subsequent iterations.

The stopping time $t_{\text{stop}}$ of each agent is recorded, and the cumulative fraction of stopped particles is plotted as a function of iteration $t$, as shown in Fig. 6. This curve, referred to as the swarm learning curve, provides insight into the collective learning speed and efficiency for different $\boldsymbol{\alpha}$ configurations.

As illustrated in the plots, in the first case almost all particles stop and reach the minimum of the Ackley function, while in the second case only about 13% of the particles stop, indicating incomplete convergence.

| Parameter | Description | Value |
|---|---|---|
| $N$ | Swarm size | 100 |
| $T_{\max}$ | Maximum number of iterations | 1000 |
| $R$ | Number of repetitions | 10 |
| $L$ | Domain size $\mathbf{X}^* = [-L, L]^2$ | 2 m |
| $\theta_{\text{vis}}$ | Visual angle of the particles | 1.48 rad |
| $r_{\text{vis}}$ | Visual radius (depth) of the particles | 1 m |
| v | Constant particle speed | 0.01 m $\text{s}^{-1}$ |
| tol | Convergence tolerance | 0.01 m |

Table 1: Summary of the parameters used in the simulations.

## 3.3 Parameter settings

All simulations were performed using $N = 100$ particles, uniformly initialized within the squared domain, with $R = 10$ repetitions per each choice of $\boldsymbol{\alpha}$ to ensure statistical robustness. The maximum number of iterations was set to $T_{\max} = 1000$. A summary of parameters is reported in Tab. 1.

## 3.4 Variation of the $\alpha$ weights

The main goal of this analysis is to determine the optimal combination of the weight parameters $\boldsymbol{\alpha} = (\alpha^{\text{cogn}}, \alpha^{\text{soc}}, \alpha^{\text{rand}})$, which govern the cognitive, social, and random components of the modified PSO dynamics. A total of 18 unique $\boldsymbol{\alpha}$ sets were tested. Performance was evaluated using the previously defined $SR$ and $CR$ metrics, for both the sphere and the Ackley functions.

The aggregated results are summarized in Tab. 2, allowing for a direct comparison between the two problems. The results obtained for the sphere and Ackley functions reveal consistent trends, but also highlight the different demands of unimodal and multimodal optimization. The sphere function primarily tests exploitation capability, while the Ackley function requires a more delicate balance between exploration and exploitation due to its many local minima.

**Edge-case configurations** The results from the "edge-case" parameter sets serve to validate the implementation of the model.

- Cognitive-only ($\boldsymbol{\alpha} = (1, 0, 0)$): in this configuration, the direction of the velocity reduces to $\mathbf{w}_i = 1 \cdot \mathbf{w}_i^{\text{cogn}}, \forall i = 1, ..., N$. At the first iteration, the personal best of each particle coincides with its current positions, i.e. $\mathbf{p}_i(0) = \mathbf{x}_i(0)$, leading to $\mathbf{w}_i^{\text{cogn}}(1) = 0$, $\forall i = 1, ..., N$. Consequently, the simulation shows that all particles never move from the initial

| Values of $\boldsymbol{\alpha}$ | | | $\mathcal{F}-$Sphere | | Ackley | |
|---|---|---|---|---|---|---|
| $\alpha^{\text{cogn}}$ | $\alpha^{\text{soc}}$ | $\alpha^{\text{rand}}$ | $SR^S$ | $CR^S$ | $SR^A$ | $CR^A$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0.6 | 0 | 0.6 | 0 |
| 0 | 0 | 1 | 0.5 | 0 | 0.3 | 0 |
| 0 | 1/2 | 1/2 | 1 | 1 | 1 | 0 |
| 1/2 | 0 | 1/2 | 1 | 0 | 1 | 0 |
| 1/2 | 1/2 | 0 | 0.6 | 0 | 0.4 | 0 |
| 1/3 | 1/3 | 1/3 | 1 | 1 | 1 | 0 |
| 0.6 | 0.2 | 0.2 | 1 | 1 | 1 | 0 |
| 0.2 | 0.6 | 0.2 | 1 | 1 | 1 | 0.5 |
| 0.2 | 0.2 | 0.6 | 1 | 0 | 1 | 0 |
| 0.6 | 0.3 | 0.1 | 1 | 1 | 1 | 0 |
| 0.6 | 0.1 | 0.3 | 1 | 1 | 1 | 0 |
| 0.1 | 0.6 | 0.3 | 1 | 1 | 1 | 0.3 |
| 0.3 | 0.6 | 0.1 | 1 | 1 | 1 | 0.9 |
| 0.3 | 0.1 | 0.6 | 1 | 0 | 1 | 0 |
| 0.1 | 0.3 | 0.6 | 1 | 0 | 1 | 0 |
| 0.05 | 0.6 | 0.35 | 1 | 1 | 1 | 0.4 |
| 0.05 | 0.5 | 0.45 | 1 | 1 | 1 | 0 |

Table 2: Statistical success and convergence rates for the 18 tested $\boldsymbol{\alpha}$ sets, compared across sphere ($SR^S$, $CR^S$) and Ackley ($SR^A$, $CR^A$) functions.

position $\mathbf{x}_i(0)$, resulting in the total absence of full convergence for both the tested functions, (i.e. $CR^S = 0$, $CR^A = 0$), while a nonzero success rate ($SR > 0$) may occur only by chance if a particle is initialized exactly at the global minimum.

- Social-only ($\boldsymbol{\alpha} = (0, 1, 0)$): the swarm is influenced exclusively by the global best position. The model successfully reaches the optimum ($SR^S = SR^A = 0.6$) for both functions, but fails to achieve full swarm convergence ($CR^S = CR^A = 0$), as once the global best is found, particles continue to be attracted toward it without any stabilizing cognitive component, resulting in oscillations and dispersion around different regions of the search space.

- Random-only ($\boldsymbol{\alpha} = (0, 0, 1)$): the motion degenerates into a pure random walk, as particles lack both memory and social interactions. This configuration performs poorly, failing to converge in either test, resulting in $CR^S = 0$, $CR^A = 0$ in both scenarios. Occasionally, a few particles may reach the minimum by chance, yielding in partial success

($SR^S = 0.5$ and $SR^A = 0.3$).

**Performance on the sphere function** The results for the sphere function highlight the sensitivity of full swarm convergence, even for a simple unimodal landscape. As shown in Tab. 2, at least one particle reaches the global minimum in all the configurations tested ($SR^S > 0, \forall \boldsymbol{\alpha}$), confirming that the algorithm consistently identifies the optimal region of the search space regardless of the set of parameters chosen. This behavior indicates a strong intrinsic stability of the swarm dynamics.

However, complete convergence of the swarm ($CR^S = 1$) is achieved only for a subset of the tested set of $\boldsymbol{\alpha}$. Failures typically occur when social or random weights are set to zero ($\alpha^{\text{soc}} = 0$ or $\alpha^{\text{rand}} = 0$), implying that both collective interaction and a stochastic component are essential for the full alignment of all agents. On the other hand, when the random weight becomes dominant ($\alpha^{\text{rand}} = 0.6$), the added noise prevents the particles from settling precisely at the minimum, leading to oscillatory behavior around the optimum.

Overall, the results indicate that for smooth unimodal problems, the best performance arises from a dominant social term combined with a moderate random perturbation.

**Performance on the Ackley function** The Ackley function represents a much more challenging multimodal scenario, where the swarm must balance exploration and exploitation to avoid being trapped in local minima.

Despite the increased complexity, every configuration leads to at least one particle reaching the global minimum (i.e. $SR^A > 0, \forall \boldsymbol{\alpha}$). This is a remarkable result, particularly considering the limited communication network among agents in this model.

However, global swarm convergence ($CR^A = 1$) occurs only in a few specific cases, all characterized by a strong social influence ($\alpha^{\text{soc}} \approx 0.6$) and a low cognitive weight ($\alpha^{\text{cogn}} \leq 0.3$), combined with a moderate random contribution ($0.1 \leq \alpha^{\text{rand}} < 0.4$). Configurations lacking randomness ($\alpha^{\text{rand}} = 0$) do not converge, as particles remain trapped in local minima without sufficient stochastic energy to escape, resulting in stagnation. This is also confirmed by the low success rate obtained for $\boldsymbol{\alpha} = (0.5, 0.5, 0)$ (i.e. $SR^A = 0.2$), indicating that the absence of the random component prevents particles from escaping from local minima.

The stagnation effect is illustrated in Fig. 7, showing the final positions of all particles after $T_{\max}$ iterations. The background of the plot represents the contour lines of the Ackley function, with the global minimum at $(0, 0)$

marked by a black star. The particles (blue dots) are visibly clustered around multiple local minima, failing to converge to the true global optimum. Adding randomness restores exploratory capability, allowing the swarm to escape local minima and converge toward the global optimum.

These results emphasize that successful optimization on multimodal landscapes requires a well-balanced interplay between social cohesion and controlled randomness, while excessive self-reliance on personal bests impedes collective progress.

**General observations and optimal configuration** Comparing the results obtained for both functions, a clear pattern emerges:

- the absence of either the social or random component ($\alpha^{\text{soc}} = 0$ or $\alpha^{\text{rand}} = 0$) leads to failure in both test cases;
- the sphere function penalizes excessive randomness, whereas the Ackley function requires it to overcome stagnation in local minima.

Among all tested, the set $\boldsymbol{\alpha} = (0.3, 0.6, 0.1)$ emerges as the most robust configuration across both tests. It achieves perfect convergence on the unimodal function ($CR^S = 1$) and the highest convergence rate on the complex multimodal Ackley function ($CR^A = 0.9$).

This optimal configuration validates the theoretical principles underlying the modified PSO model: low cognitive bias, strong social consensus, and moderate random exploration provide the most effective balance to solve complex multimodal problems without sacrificing precision on simple ones.

While Tab. 2 provides detailed quantitative results, visualizing the performance of the swarm within the $\boldsymbol{\alpha}$ space offers a more intuitive understanding of the relationships between parameters.

Fig. 8 presents a 3D scatter plot in which each of the 18 tested $\boldsymbol{\alpha}$ sets is represented as a point according to its $(\alpha^{\text{cogn}}, \alpha^{\text{soc}}, \alpha^{\text{rand}})$ components. The performance is encoded using the following color map:

- white dots: no particles have reached the global minimum of the function, i.e. $SR = 0$;
- light blue dots: at least one particle has reached the global minimum, but without full convergence, i.e $SR > 0$ and $CR < 1$;
- blue dots: all particles have successfully converged to the true minimum of the function, i.e. $CR = 1$.

This visualization highlights how the balance between cognitive, social, and random weights directly affects the ability of the swarm to solve the optimization problems.

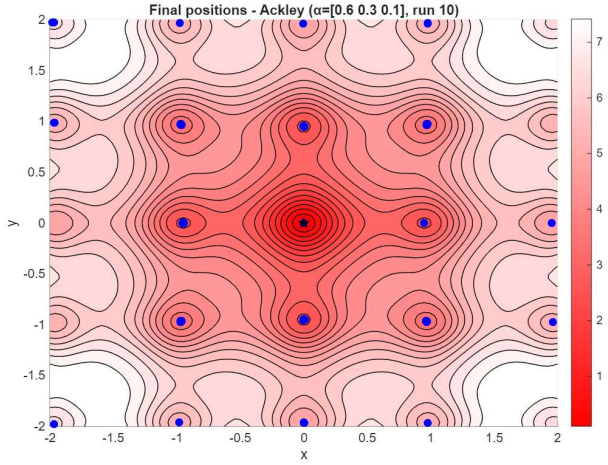


Figure 7: Example of stagnation in Ackley function.

## 3.5 Note on the influence of velocity and domain scale

In addition to the $\boldsymbol{\alpha}$ weights analysis, a complementary set of experiments was conducted to evaluate the sensitivity of the model to the speed v. Keeping all the other parameters identical to the previous simulations, the swarm was tested with three additional velocities: $\mathrm{v} \in \{0.05, 0.1, 0.5\}$. The corresponding success and convergence rates for both the sphere and the Ackley functions are reported respectively in Tab. 3 and Tab. 4, together with the results already examined for $\mathrm{v} = 0.01$.

Overall, the results indicate that the velocity magnitude strongly influences swarm convergence. For both functions, at least one particle always succeeds in reaching the global minimum ($SR > 0, \forall \mathrm{v}$), confirming the robustness of the exploratory mechanism of the model. However, full swarm convergence ($CR = 1$) is highly sensitive to the chosen speed.

For the sphere function, convergence progressively deteriorates as v increases. The best results are obtained for $\mathrm{v} = 0.01$, where most configurations achieve $CR^S = 1$. At intermediate velocities ($\mathrm{v} = 0.05, \mathrm{v} = 0.1$) particles tend to overshoot the minimum, leading to oscillatory stabilization, while at the highest speed tested, convergence becomes unstable for nearly all parameter sets. This behavior reflects the balance between movement resolution and attraction dynamics: excessively fast particles lack the precision required to settle within the narrow tolerance region of the minimum.

For the Ackley function, the overall trend is similar, but with stronger

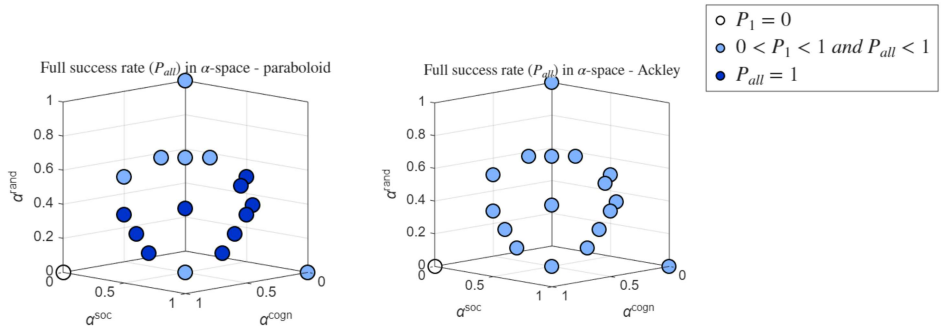


Figure 8: 3D representation of the tested $\boldsymbol{\alpha}$ parameter sets in the $(\alpha^{\text{cogn}}, \alpha^{\text{soc}}, \alpha^{\text{rand}})$ space. On the left the 3D plot for the sphere function, on the right for the Ackley function.

sensitivity to v. While the success rate remains high for most parameter sets, reflecting the observations done above, the convergence rate is consistently zero for all tested velocities. Moreover, increasing v does not enhance exploration: larger velocities cause particles to skip over narrow attraction basins and oscillate between local minima, preventing stable convergence.

To further investigate the role of velocity scaling relative to the search space, an additional test was performed on the Ackley function with an expanded domain $\mathbf{X} = [-10, 10]^2$ and a significantly higher particle velocity $\mathrm{v} = 1$. In this configuration, an apparent increase in the success rate was observed., due to the high velocity-to-domain ratio ($\mathrm{v}/L = 0.1$, compared to $\mathrm{v}/L = 0.005$ in the main experiment).

At such speed, the particles take steps that are too large to perceive or be captured by the local minima of the Ackley function. As a result, the algorithm effectively "jumps" over the small-scale traps, perceiving only the large-scale global basin of attraction. This behavior artificially simplifies the optimization problem, leading to apparently easier convergence that does not reflect true exploratory capability.

These finding confirms that small velocities (e.g. $\mathrm{v} = 0.01$) are best suited for assessing the true dynamic balance between exploration and exploitation, while larger velocities can distort the search process by suppressing the effect of multimodality. The choice of v seems to be made in proportion to the spacial scale of the domain and the expected complexity of the objective function.

| Values of $\boldsymbol{\alpha}$ | | | $v_1 = 0.01$ | | $v_2 = 0.05$ | | $v_3 = 0.1$ | | $v_4 = 0.5$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha^{\text{cogn}}$ | $\alpha^{\text{soc}}$ | $\alpha^{\text{rand}}$ | $SR$ | $CR$ | $SR$ | $CR$ | $SR$ | $CR$ | $SR$ | $CR$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0.6 | 0 | 1 | 0 | 1 | 0 | 0.6 | 0 |
| 0 | 0 | 1 | 0.5 | 0 | 1 | 0 | 0.7 | 0 | 0.8 | 0 |
| 0 | 1/2 | 1/2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1/2 | 0 | 1/2 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1/2 | 1/2 | 0 | 0.6 | 0 | 1 | 0 | 1 | 0 | 0.1 | 0 |
| 1/3 | 1/3 | 1/3 | 1 | 1 | 1 | 0.9 | 1 | 0 | 0.4 | 0 |
| 0.6 | 0.2 | 0.2 | 1 | 1 | 1 | 0 | 1 | 0 | 0.8 | 0 |
| 0.2 | 0.6 | 0.2 | 1 | 1 | 1 | 0 | 1 | 0 | 0.2 | 0 |
| 0.2 | 0.2 | 0.6 | 1 | 0 | 1 | 1 | 1 | 0.1 | 1 | 0 |
| 0.6 | 0.3 | 0.1 | 1 | 1 | 1 | 0 | 1 | 0 | 0.4 | 0 |
| 0.6 | 0.1 | 0.3 | 1 | 1 | 1 | 0.3 | 1 | 0 | 0.6 | 0 |
| 0.1 | 0.6 | 0.3 | 1 | 1 | 1 | 0.2 | 1 | 0 | 0.1 | 0 |
| 0.3 | 0.6 | 0.1 | 1 | 1 | 1 | 0 | 1 | 0 | 0.6 | 0 |
| 0.3 | 0.1 | 0.6 | 1 | 0 | 1 | 1 | 1 | 0.2 | 1 | 0 |
| 0.1 | 0.3 | 0.6 | 1 | 0 | 1 | 1 | 1 | 0.2 | 1 | 0 |
| 0.05 | 0.6 | 0.35 | 1 | 1 | 1 | 0.9 | 1 | 0 | 0.2 | 0 |
| 0.05 | 0.5 | 0.45 | 1 | 1 | 1 | 1 | 1 | 0 | 0.8 | 0 |

Table 3: Success and convergence rates for the sphere function, compared across the 18 tested $\boldsymbol{\alpha}$ sets and different velocities.

## 3.6 Simulation on a real search space

To further validate the proposed model beyond the benchmark functions, a simulation was conducted in a simplified real-world environment: Piazza Vittorio Veneto in Turin. The search space is modeled as a rectangular domain $\mathbf{X} = [-L_1, L_1] \times [-L_2, L_2]$, representing the accessible area of the square.

**Domain and exit configuration** The environment includes six exit points: two main exits (Via Po and the access to the Vittorio Emanuele I bridge) and four secondary exits located along the two lateral sides of the square (see Fig. 9).

In an evacuation scenario, only the main exits ensure safe outflow. Secondary exits, although physically accessible, are too narrow and structurally inadequate to be considered viable evacuation points. Despite this, pedestrians may still be drawn toward them if they approach closely enough to visually detect them.

| Values of $\boldsymbol{\alpha}$ | | | $v_1 = 0.01$ | | $v_2 = 0.05$ | | $v_3 = 0.1$ | | $v_4 = 0.5$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha^{\text{cogn}}$ | $\alpha^{\text{soc}}$ | $\alpha^{\text{rand}}$ | $SR$ | $CR$ | $SR$ | $CR$ | $SR$ | $CR$ | $SR$ | $CR$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0.6 | 0 | 0.9 | 0 | 0.9 | 0 | 0.5 | 0 |
| 0 | 0 | 1 | 0.3 | 0 | 0.9 | 0 | 0.9 | 0 | 0.9 | 0 |
| 0 | 1/2 | 1/2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1/2 | 0 | 1/2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1/2 | 1/2 | 0 | 0.4 | 0 | 0.3 | 0 | 0.8 | 0 | 0.1 | 0 |
| 1/3 | 1/3 | 1/3 | 1 | 0 | 1 | 0 | 1 | 0 | 0.6 | 0 |
| 0.6 | 0.2 | 0.2 | 1 | 0 | 1 | 0 | 1 | 0 | 0.8 | 0 |
| 0.2 | 0.6 | 0.2 | 1 | 0.5 | 1 | 0 | 1 | 0 | 0.4 | 0 |
| 0.2 | 0.2 | 0.6 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0.6 | 0.3 | 0.1 | 1 | 0 | 1 | 0 | 1 | 0 | 0.1 | 0 |
| 0.6 | 0.1 | 0.3 | 1 | 0 | 1 | 0 | 0.9 | 0 | 0.6 | 0 |
| 0.1 | 0.6 | 0.3 | 1 | 0.3 | 1 | 0.2 | 1 | 0 | 0.1 | 0 |
| 0.3 | 0.6 | 0.1 | 1 | 0.9 | 0.9 | 0 | 1 | 0 | 0.8 | 0 |
| 0.3 | 0.1 | 0.6 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0.1 | 0.3 | 0.6 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0.05 | 0.6 | 0.35 | 1 | 0.4 | 1 | 0.2 | 1 | 0 | 0.2 | 0 |
| 0.05 | 0.5 | 0.45 | 1 | 0 | 1 | 0 | 1 | 0 | 0.8 | 0 |

Table 4: Success and convergence rates for the Ackley function, compared across the 18 tested $\boldsymbol{\alpha}$ sets and different velocities.

To correctly capture the different nature of the two types of exits, the objective function was designed to feature two global minima (corresponding to the main exits) and four local minima (associated with the secondary exits). Importantly, each local minimum becomes detectable only when the agent is sufficiently close to the corresponding secondary exit. This models the architectural occlusions created by the porticoes, which prevent pedestrians from visually identifying the side exits until they approach them closely.

**Objective function** The objective function acts as a cost function, guiding agents toward safe exits while penalizing movements toward secondary ones. It combines a global cost component based on the distance from the main exits with a local attraction term representing the temporary influence of the secondary exits. It is defined as:

$$\mathcal{F}(\mathbf{x}) = \max\left(0,\ \mathcal{F}_{\text{main}}(\mathbf{x}) - C(\mathbf{x})\right). \tag{36}$$

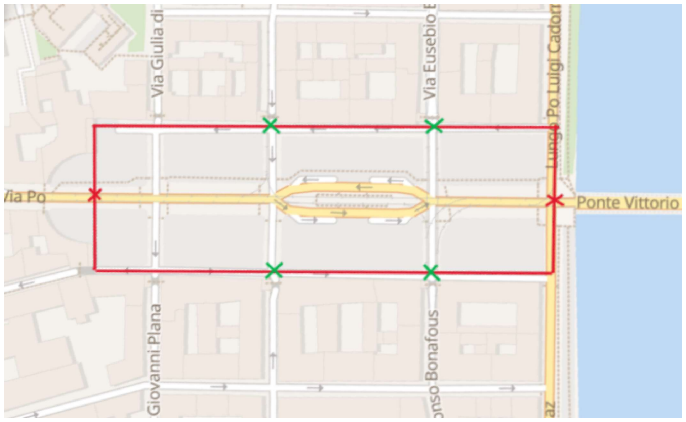


Figure 9: Map of the simulated environment showing the positions of the main (red) and secondary (green) exits, together with the rectangular domain $\mathbf{X}$.

The first term,

$$\mathcal{F}_{\text{main}}(\mathbf{x}) = \min(||\mathbf{x} - \mathbf{d}_1||,\ ||\mathbf{x} - \mathbf{d}_2||), \tag{37}$$

represents the distance to the nearest main exit $\mathbf{d}_1$ or $\mathbf{d}_2$. The second term,

$$C(\mathbf{x}) = \max(0,\ A - \frac{A}{\sigma^2} \cdot d_{\text{min}}^2), \tag{38}$$

models the local attractions cone generated by the closest secondary exit. Its formulation incorporates two key aspects:

1. visibility constraint: the cone becomes positive only when the agent is at a distance $d_{\text{min}} < \sigma$, meaning that secondary exits influence the cost only when they become visible;

2. depth modulation: the attraction increases as the agent approaches the exit, reaching depth $A$ at its center. This reflects the natural tendency of pedestrians to be increasingly attracted to what appears to be a viable exit once it enters their field of view.

The resulting function is characterized by two global minima in $\mathbf{x}_1^* = (-180, 0)$ and $\mathbf{x}_2^* = (180, 0)$, corresponding to the two main exits located on opposite sides of the square, and four local minima: $\mathbf{x}_1 = (-15, -55), \mathbf{x}_2 = (-15, 55), \mathbf{x}_3 = (95, -55)$ and $\mathbf{x}_4 = (95, 55)$.

The goal of the swarm is to reach one of the global minima, where $\mathcal{F}(\mathbf{x}^*) = 0$, representing safe evacuation.

| Parameter | Description | Value |
|---|---|---|
| $N$ | Swarm size | 100, 200, 300 |
| $T_{\max}$ | Maximum number of iterations | 1000 |
| $R$ | Number of repetitions | 10 |
| $L_1, L_2$ | Domain size $\mathbf{X}^* = [-L_1, L_1] \times [-L_2, L_2]$ | 180,55 m |
| $\theta_{\text{vis}}$ | Visual angle of the particles | 1.48 rad |
| $r_{\text{vis}}$ | Visual radius (depth) of the particles | 50 m |
| v | Constant particle speed | 1.5 m $s^{-1}$ |
| tol | Convergence tolerance | 1.5 m |
| $A$ | Deepness of the cone | 30 m |
| $\sigma$ | Radius of influence of the secondary doors | 30m |

Table 5: Summary of the parameters used in the real-case scenario simulations.

**Evaluation metrics** To assess the performance of the swarm in the realistic evacuation scenario, the same convergence metrics used for the benchmark functions were employed:

- success rate ($SR$): fraction of repetitions in which at least one particle reaches a global minimum;
- convergence rate ($CR$): fraction of repetitions in which all particles reach a global minimum.

In addition, the time of full convergence $T_{\text{all}}$ is computed by collecting the stopping times of all particles within each simulation and averaging them, following the same analysis procedure as before.

**Parameter settings** The simulation was performed using only the $\boldsymbol{\alpha}$ configuration that yielded the best performance across all previous tests, i.e. $\boldsymbol{\alpha} = (0.3, 0.6, 0.1)$. Three different population sizes $N$ where tested, with agents uniformly initialized in the domain and $R = 10$ repetitions per each choice of $N$. Also in this case, the maximum number of iterations was set to $T_{\max} = 1000$. A summary of the parameters is reported in Tab. 5.

**Results analysis** The results are summarized in Tab. 6.

For every population size tested, $SR = 1$ and $CR = 1$, i.e. in every repetition all particles reach one of the global minima. This means that no particle remains trapped in any of the four local minima, despite their presence and their attraction cones. The swarm consistently identifies the two safe exits, demonstrating that the optimal parameter configuration

| Population size $N$ | $SR$ | $CR$ | Time of full convergence |
|---|---|---|---|
| 100 | 1 | 1 | 155 |
| 200 | 1 | 1 | 143 |
| 300 | 1 | 1 | 147 |

Table 6: Success rate ($SR$), convergence rate ($CR$), and average time of full convergence ($T_{\text{all}}$) for the real-case objective function, reported for the three tested population sizes $N$.

$\boldsymbol{\alpha} = (0.3, 0.6, 0.1)$ generalizes well from synthetic multimodal functions to a realistic spatially structured scenario.

Convergence times remain stable across different swarm sizes. The values of $T_{\text{all}}$ are extremely close for all population sizes, indicating that:

- increasing the number of particles $N$ does not significantly alter the speed of convergence;
- convergence is dominated by the strong attraction toward the main exits;
- secondary exits create temporary perturbations, but do not induce stagnation.

Although the objective function includes four local minima, none of the simulations ended with particles trapped in these points. Some particles may temporarily enter the attraction cones of secondary exits, reflecting perceptual limitations, but the strong social component encourages the to re-align with the overall swarm direction and follow the collective movement toward a main exit.

A representative plot of the final particle distribution in Fig. 10 illustrates that all agents consistently accumulate at the two global minima, visually confirming the quantitative results.

**Final observations** The real-case scenario confirms the robustness and scalability of the proposed model. Despite the presence of misleading local minima and visibility constraints, the swarm consistently converges to the safe exits with perfect success and convergence rates across all population sizes. The stability of convergence times and the absence of stagnation near secondary exits highlight the effectiveness of the balance between individual, social, and environmental components encoded in the parameter set $\boldsymbol{\alpha} = (0.3, 0.6, 0.1)$.

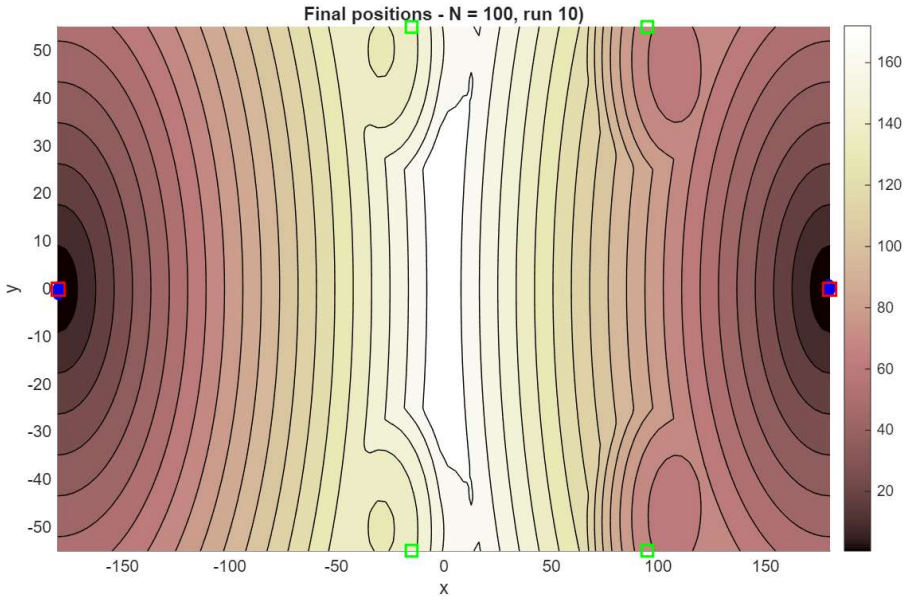


Figure 10: Final particle distribution at the end of a representative simulation with $N = 100$: all agents converge to the two global minima, while no agent remains trapped in a local minimum.

These results demonstrate that the modified PSO algorithm successfully captures evacuation-like behaviors in simplified spatial domains and performs reliably when transitioning from synthetic benchmark functions to realistic environments.

# 4 Conclusions

This thesis presented a modified Particle Swarm Optimization (PSO) model tailored to simulate pedestrian behavior in evacuation scenarios. The canonical PSO framework was extended by incorporating a random component into the velocity update, introducing perceptual constraints through a visual neighborhood, and adapting the interaction rules to reflect features of human crowd motion. These modifications were designed to overcome the limitations of the canonical algorithm when applied to realistic and spatially structured environments.

The performance of the proposed model was assessed on two benchmark functions, a simple unimodal function (the sphere) and the highly multimodal Ackley landscape, and on a real-scenario objective function representing Piazza Vittorio Veneto. Across all tests, the model exhibited robust and stable behavior. In particular, the results obtained with the Ackley function highlight the ability of the model to avoid local minima and maintain a balance between exploration and exploitation, a property that is essential in evacuation settings. The real-scenario simulation further confirmed the effectiveness of the model: for every population size tested, all agents successfully reached one of the two global minima, demonstrating reliable convergence and strong resilience to misleading local minima generated by secondary exits.

The outcomes of this work indicate that the proposed modifications of the PSO are suitable for modeling pedestrian evacuation, while ensuring good scalability and consistent convergence properties.

Several directions remain open for future research. A natural extension could consist in the introduction of obstacles into the domain to more accurately reflect real urban geometry and to test the ability of the model to navigate constrained environments. Additional perceptual neighborhoods, such as auditory ones, could be incorporated to capture richer forms of human awareness. Finally, the objective function could be expanded to account for environmental hazards such as smoke, reduced visibility, or dynamic changes in the accessibility of exits.

Overall, the results seem to indicate that the modified PSO framework constitutes a flexible tool to simulate pedestrian dynamics in evacuation contexts, offering both methodological insights and practical potential for real-world applications.

# References


Carlisle, Anthony and Gerry Dozier (2001). "Tracking changing extrema with particle swarm optimizer". In: *Auburn Univ., Auburn, AL, Tech. Rep. CSSE01-08*.

Clerc, Maurice and James Kennedy (2002). "The particle swarm-explosion, stability, and convergence in a multidimensional complex space". In: *IEEE transactions on Evolutionary Computation* 6.1, pp. 58–73.

Colombi, Annachiara, Marco Scianna, and Alessandro Alaia (2017). "A discrete mathematical model for the dynamics of a crowd of gazing pedestrians with and without an evolving environmental awareness". In: *Computational and Applied Mathematics* 36, pp. 1113–1141.

Doctor, Sheetal and Ganesh K Venayagamoorthy (2005). "Improving the performance of particle swarm optimization using adaptive critics designs". In: *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.* IEEE, pp. 393–396.

Duan, Haibin, Daifeng Zhang, Yuhui Shi, and Yimin Deng (Dec. 2018). "Close formation flight of swarm unmanned aerial vehicles via metric-distance brain storm optimization". In: *Memetic Computing* 10. DOI: 10.1007/s12293-018-0251-z.

Eberhart, Russell C and Yuhui Shi (2000). "Comparing inertia weights and constriction factors in particle swarm optimization". In: *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*. Vol. 1. IEEE, pp. 84–88.

Fan, Huiyuan (2002). "A modification to particle swarm optimization algorithm". In: *Engineering Computations* 19.8, pp. 970–989.

Fourie, PC and Albert A Groenwold (2002). "The particle swarm optimization algorithm in size and shape optimization". In: *Structural and Multidisciplinary Optimization* 23.4, pp. 259–267.

Heppner, Frank and Ulf Grenander (1990). "The ubiquity of chaos". In: *Sl]: Amer Assn for the Advancement*, p. 247.

Ide, Azuma and Keiichiro Yasuda (2005). "A basic study of adaptive particle swarm optimization". In: *Electrical Engineering in Japan* 151.3, pp. 41–49.

Juang, Yau-Tarng, Shen-Lung Tung, and Hung-Chih Chiu (2011). "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions". In: *Information Sciences* 181.20, pp. 4539–4549.

Kennedy, James and Russell Eberhart (1995). "Particle swarm optimization". In: *Proceedings of ICNN'95-international conference on neural networks*. Vol. 4. ieee, pp. 1942–1948.

Kennedy, James and Rui Mendes (2002). "Population structure and particle swarm performance". In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*. Vol. 2. IEEE, pp. 1671–1676.

Lu, Jingcheng, Eitan Tadmor, and Anil Zenginoğlu (2024). "Swarm-based gradient descent method for non-convex optimization". In: *Communications of the American Mathematical Society* 4.17, pp. 787–822.

Mendes, Rui, James Kennedy, and José Neves (2003). "Watch thy neighbor or how the swarm can learn from its environment". In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*. IEEE, pp. 88–94.

Parsopoulos, Konstantinos E and Michael N. Vrahatis (2002). "Recent approaches to global optimization problems through particle swarm optimization". In: *Natural computing* 1, pp. 235–306.

Ratnaweera, Asanga, Saman K Halgamuge, and Harry C Watson (2004). "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients". In: *IEEE Transactions on evolutionary computation* 8.3, pp. 240–255.

Schutte, Jaco F and Albert A Groenwold (2005). "A study of global optimization using particle swarms". In: *Journal of global optimization* 31, pp. 93–108.

Suganthan, Ponnuthurai N (1999). "Particle swarm optimiser with neighbourhood operator". In: *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*. Vol. 3. IEEE, pp. 1958–1962.

Wang, Dongshu, Dapei Tan, and Lei Liu (2018). "Particle swarm optimization algorithm: an overview". In: *Soft computing* 22.2, pp. 387–408.

Yu, Huanjun, Liping Zhang, DZ Chen, XF Song, and SX Hu (2005). "Estimation of model parameters using composite particle swarm optimization". In: *Journal of Chemical Engineering of Chinese Universities* 19.5, p. 675.