Adaptive Swarm Coordination for Wildfire Control via Q-Learning Tuned PSO with Quantum-Inspired Coupling

Sonika Dhenuva Konda[1], **Grégoire Bergamo**[2]

Received March 8, 2026
Accepted September 24, 2026
Electronic access November 30, 2026

The nature of the fire being highly unpredictable and fast has made wildfire suppression rather difficult during the years. In addition to that, limited availability of the resources that can be utilized through aerial means complicates the situation further. Autonomous swarm of drones can be used to optimize and decentralize the firefighting and fire sensing processes. This paper explores three swarm methods. Vanilla Particle Swarm Optimization (S-PSO), Reinforcement Learning Driven Particle Swarm Optimization (RL-PSO), and RL Quantum Coupled Particle Swarm Optimization (RL-QCPSO). These approaches are compared in terminal-based simulations including terrain, wind effects, wildfire spread and drone characteristics and capabilities. Among the metrics analyzed here is the number of extinguished fires, number of trees that were saved, response time and efficiency. According to the results of this research, best results in terms of the performance metrics are provided by RL Quantum Coup. PSO. Its efficiency is characterized by increased number of trees surviving by 11. 5% relative to that of S-PSO. Although the RL-PSO makes the system more adaptable, it decreases its efficiency in the terms of speed of operations. On the contrary, while S-PSO is fast, it performs worse in dynamic environments. The findings of this research illustrate the possibilities and the potential of combining reinforcement learning and quantum inspired swarm intelligence techniques. The collaboration between these two enables the system to adapt in a dynamic environment and avoid inefficient search methods.

Key words: Swarm intelligence, particle swarm optimization, reinforcement learning, quantum inspired algorithm, drones

Introduction

Wildfires are a prevalent threat to the environment that can easily destroy many ecosystems and nearby communities. The most efficient way of wildfire management includes rapid detection and monitoring the surrounding environment and this requires a largely coordinated response system that can span across large and difficult terrain. Traditional firefighting methods rely heavily on human coordination and can struggle to respond quickly enough to fires that spread across dense forests. As a result, many researchers have started to explore use of autonomous systems, specifically drone swarms, to help with wildfire monitoring[1] and suppression.

One effective strategy to coordinate these large numbers of autonomous drones is called swarm intelligence. Swarm intelligence is a part of intelligence that takes inspiration from collective behavior in many natural systems[2] all over the world. This field studies how communities of intelligent agents cooperate to solve complicated problems. It looks at how groups of agents, like ants or birds, work together to solve complex problems. The intelligent agents operate under certain rules. However, their overall behavior results in complicated and smart structures. Swarm intelligence serves as a foundation for developing adaptive and learning algorithms within computer science. The goal is to create solutions by mimicking nature's way of working in large groups. Because they are not centralized, the systems tend to be inherently robust since they can function effectively despite the failure of several individual agents.

These attributes mentioned above make swarm intelligence make swarm particularly noticeable in cases where the application of distributed autonomous systems[3] is considered. Swarm coordinated drones can be used to span large geographical areas and share information without a controller[4]. Reinforcement learning has also been used to optimize UAV swarm pathways as a wildfire evolves[5]. Such skillsets are especially valuable in large scale disaster response scenarios, in which conditions are highly unstable and may change rapidly, furthermore limiting the communication infrastructure.

This study investigates how swarm-based optimization algorithms can be used to coordinate autonomous drones in wildfire suppression environments. In particular, the work evaluates swarm algorithms within a simulated wildfire model to analyze how decentralized coordination can influence fire containment and forest preservation.

[1] *Del Norte Highschool, San Diego, CA, United States*
[2] *Norwegian University of Science and Technology, Trondheim, Norway; Harvard University, Cambridge, MA, United States*

© The National High School Journal of Science 2026

Related Work

The concept of swarm intelligence was formally introduced by Gerardo Beni and Jing Wang[6] in 1989 during their research on distributed robotic systems known as Cellular Robotic Systems. Their work demonstrated that large groups of simple robots could perform complex tasks through local interactions[7] without centralized control. This research showed that complex global behavior could emerge from decentralized systems[6] forming the foundation for modern swarm robotics[8] and distributed multi-agent coordination.

Among the popular swarm-based optimization algorithms include Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The ACO algorithm was invented by Marco Dorigo[9] during the early 1990s and the theory behind the algorithm is based on the food foraging of ant colonies. During their food hunt, ants lay down pheromones on the routes they take to get to the food source[9]. When more and more ants travel through a particular route, the amount of pheromones becomes increasingly high, and this eventually leads to the ant colony converging towards the shortest route from their nest to the food sources.

PSO, introduced by James Kennedy and Russell Eberhart[10] in 1995, is another widely used swarm-based optimization technique inspired by the collective motion of bird flocks and fish schools. In PSO, each particle represents a potential solution within a search space. Particles update their positions[10] based on both their own previous best solution and the best solution discovered by other members of the swarm based on. With time, this cooperative method enables the swarm to approach high-quality solutions. Due to its simplicity and computational efficiency[11] PSO has been widely researched as a swarm-based optimization technique, with several modifications made to solve various optimization problems[12].

Swarm-based methods have also been applied to the coordination of autonomous aerial vehicles. Research by Amala Arokia Nathan et al.[4] demonstrated a drone swarm system that uses real-time particle swarm optimization combined with synthetic aperture sensing to navigate dense forest environments. Their approach was able to detect hidden targets in as little as 14 seconds while achieving visibility rates of up to 72%, outperforming traditional search strategies. These results highlight the potential of swarm coordination for search and rescue operations[4] in complex environments. Hybrid PSO methods have been used more recently in multi-UAV path planning, taking into consideration both path feasibility and coordination among multiple UAVs[13].

There has been increasing researches done on using autonomous drones in wildfire surveillance[1] and disaster. For example, research by Luis Merino and collaborators demonstrated an unmanned aerial vehicle system capable of automatically detecting and monitoring forest fires while collecting real-time environmental data. The system demonstrated that autonomous aerial vehicles could greatly assist in wildfire monitoring through constant surveillance and situational awareness for firefighting forces. These systems help in reducing dangers to human pilots as well as effective monitoring of wide territories.

Building on these developments, this study investigates how swarm optimization techniques can be used to coordinate autonomous drone swarms in wildfire suppression scenarios. By evaluating the performance of swarm-based algorithms in simulated wildfire environments, this research aims to better understand how decentralized coordination strategies can improve fire containment and environmental protection.

Research Questions and Hypothesis

This research aims to assess the impact of swarm based optimization algorithms on the efficiency of autonomous drone coordination in fighting fires. Despite the extensive research on swarm intelligence in robotics and optimization studies, the practical application of swarm intelligence in combating wildfires still needs further research. This includes the analysis of the impact of various swarm coordination techniques[14] on fire prevention and forest conservation. To explore this problem, the following research question is proposed: How do different swarm optimization strategies affect the performance of autonomous drone swarms in simulated wildfire suppression environments? More specifically, this study compares the performance of three swarm coordination approaches: Vanilla Particle Swarm Optimization (V-PSO), a reinforcement learning enhanced variant of PSO (RL-PSO), and a RL Quantum Coup. PSO model (RL-QCPSO). These algorithms are evaluated using wildfire simulation to measure metrics such as fire suppression effectiveness, number of fires extinguished, and forest preservation. Based on prior research on swarm optimization and distributed learning systems[3,15], the following hypothesis is proposed. A RL Quantum Coup. PSO model will demonstrate the highest overall performance among the tested algorithms by improving coordination between drones and reducing the likelihood of the swarm becoming trapped in suboptimal search patterns.

Methodology

This study implements a simulated Particle Swarm Optimization (PSO) based drone swarm for wildfire suppression within a two-dimensional grid environment. The effectiveness of three swarm optimization algorithms[7,10,15] is evaluated in terms of fire detection and extinguishing performance. The methodology is divided into four stages: (a) environmental modeling, (b) swarm design, (c) drone control via the algorithms, and (d) experimental evaluation.

© The National High School Journal of Science 2026

Environment Model

Simulation is carried out in the form of a 40 by 20 square grid, where each grid corresponds to a 10 by 10 meter square region in the simulation. The attributes stored in each grid cell include a presence of tree (0), presence of fire (5), burnt grid (4), and the presence of water (3). In essences, a numerical matric of the environment at any given period, or time step, is created. Elevation value is assigned to each cell to consider the slope effect on the fire transfer. Wind is simulated by a vector (v_x, v_y) that assumes values of either -1, 0, 1. Spread probability for fires depends on terrain slope, wind direction, and a stochastic process that puts the fire out. Fires are ignited in seven randomly chosen cells close to the center of the grid initially. Since the fires can continue spreading during the simulation period, there is no predetermined number of burning cells other than the number of fires, depending on the effectiveness of swarm in extinguishing the fire. Water refill grids are defined in either two or three clusters of adjacent cells, taking the place of trees at such grids. Probability that a fire spreads from burning grid to the adjacent cells is computed based on slope effect between the two grids, wind effect, and random extinguishing where there is 2% chance of fire being put out per burning grid per time step. As fire can spread from burning cells, there is no preset number of fires but the total number of cells on the grids is the maximum limit of the number of fires. Environment has been kept intentionally simple to have a controllable simulation setting to conduct algorithmic comparisons and not to create an entire realistic wildfire model.

$$P_{\text{spread}} = \min\big(\max\big(P_0 + \beta \times \Delta h + W, 0.01\big), 0.5\big) \quad \text{(Eqn 1)}$$

where $P_0 = 0.02$ is the spread probability and $\beta \times \Delta h$, where Δh is the elevation or height difference between the cell and its neighbor. Fires spread more easily uphill ($\Delta h > 0$) and while spreading more slowly downhill ($\Delta h < 0$) and also wind effect where $W = 0.1$ if the wind direction aligns exactly with the neighbor direction, otherwise, $W = 0$. The probability range is set between minimum of 1% and 50% to maintain stability. Once the fire in a cell is extinguished, that cell remains in a burnt cell for the rest of the simulation.

Drone Swarm Design

Drones are deployed at step 10 of 500, allowing initial fire propagation. Each drone moves at 10 m/s and carries three units of water. Extinguishing a burning cell requires 40 seconds, while refilling at a water source requires 30 seconds. Drone positions are mapped to the grid for environmental interaction, while PSO operates in continuous space. Drones monitor remaining water and track personal best[10] performance to inform swarm coordination.

Control Algorithms

Three swarm intelligence algorithms are implemented to coordinate drones in dynamic environments[16].Vanilla Particle Swarm Optimization represents the baseline, following the standard PSO formulation used for comparison with subsequent PSO improvements[17]. Reinforcement learning-enhanced PSO approaches have been investigated for adaptive UAV path planning in complex environments[18]. In this study, RL-PSO and RL-Quantum Coup. PSO extend the standard PSO framework to improve adaptability and performance in the simulation environment.

In Standard PSO, each particle represents a potential solution. Its position indicates the current solution, and its velocity determines how the position changes. Velocity[10] is updated according to

$$v_i(t+1) = \omega v_i(t) + \varphi_p r_p\big(pbest_i - x_i(t)\big) + \varphi_g r_g\big(gbest - x_i(t)\big) \quad \text{(Eqn 2)}$$

where ω is the inertia weight, φ_p and φ_g are are the cognitive and social coefficients, and r_p and r_g are random numbers from 0 to 1 that introduce variability to the swarm allowing them to explore. The position is then updated as in Equation 3:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad \text{(Eqn 3)}$$

Standard PSO balances exploration and exploitation[11,19], but may converge prematurely[20] in dynamic environments as its parameters are static.

Reinforcement Learning enhanced PSO allows swarm parameters to adapt in real time. Previous research has demonstrated that reinforcement learning can be used to adapt PSO parameters during optimization, allowing the algorithm to respond to changes in the search process[21].An RL agent observes swarm level metric such as particle diversity, convergence behavior[20], average distance to fire, remain drone water capacity and local terrain conditions. Based on this state, the RL agent dynamically adjust the PSO parameters $(\omega, \varphi_p, \varphi_g)$ for the next iteration in order to improve swarm search behavior[22] and coordination. The RL agent uses a discrete state-action representation for Q-learning. Swarm state is determined using two metrics: global best fitness improvement and swarm diversity[19]. Global best[10] improvement is calculated as the change in the swarm's best fitness value between iterations, while diversity is measured using the standard deviation of particle positions. Both metrics are discretized into three levels (low, medium, high), producing a $3 \times 3 \times 3$ state space with 27 total possible states. The action space consists of 12 discrete PSO parameter combinations formed from inertia weight values $\omega \in \{0.4, 0.7, 0.9\}$, cognitive coefficient values $\varphi_p \in \{1.0, 1.5, 2.0\}$, and social coefficient values $\varphi_g \in \{1.0, 1.5, 2.0\}$. In every iteration, the RL

agent chooses one set of parameter values action from the Q-table for fine-tuning the swarm's actions. In the implementation, Standard PSO uses fixed parameters $\omega = 0.7$, $\varphi_p = 1.5$, and $\varphi_g = 1.5$. For RL-PSO and RL Quantum Coup. PSO, the RL agent selects parameters from discrete ranges $\omega \in \{0.4, 0.7, 0.9\}$, $\varphi_p \in \{1.0, 1.5, 2.0\}$, and $\varphi_g \in \{1.0, 1.5, 2.0\}$. RL Quantum Coup. PSO additionally tunes the coupling parameter $\kappa \in \{0.0, 0.1, 0.3\}$. All particle velocities are initialized randomly within the range $[-1.5, 1.5]$. No explicit velocity clamping is applied. Instead, particle positions are clipped to remain within the simulation grid boundaries. The Q-Learning table is shown below

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a')\right) \quad \text{(Eqn 4)}$$

In Q-learning[23] update, r represents the reward, α is the learning rate, γ is the discount factor, s is the current swarm state, a is the selected parameter adjustment action, s' and is the resulting state after the action is applied. In this implementation, the reward is mainly based on improvement in the swarm's global best fitness value, along with a small diversity bonus that encourages exploration. The reward focuses more on improving swarm coordination and search behavior rather than directly maximizing tree survival or minimizing runtime. This allows the RL agent to adjust parameters automatically as wildfire conditions change and improve overall swarm performance.

The reward structure evaluates whether the selected PSO parameter settings improved the overall performance of the swarm rather than directly rewarding individual drone actions. Positive rewards are given when the swarm successfully extinguishes fires, uses water efficiently, maintains good coverage[24] of active fire areas and successfully refills at water sources when needed. Other rewards are provided if the parameter tuning results in better swarm coordination and convergence. Penalties are imposed for behaviors that are inefficient in nature, such as drones flying close to one another, wasting water on non-fire cells, delayed action to the fire outbreaks, or fires spreading.

Rather than directly controlling drone actions, the RL agent learns which PSO parameter settings lead to better swarm behavior under changing wildfire conditions[16,25] The reward structure helps the agent improve the swarm's exploration, coordination, and convergence in real time by tuning the PSO algorithm itself[26,27].

RL Quantum Coup. PSO builds on RL PSO by adding a quantum inspired coupling mechanism[28].This mechanism enhances exploration and helps prevent premature convergence[29], allowing particles to not get stuck too early in a bad situation. Quantum-behaved particle swarm optimization has also been proposed as a quantum-inspired extension of PSO, introducing quantum behavior into the movement and search of particles[30].Particles still maintain positions and velocities as in standard PSO, while inertia weight ω and cognitive and social coefficients (φ_p and φ_g) are adaptively tuned by the RL agent based on the swarm's current state. In addition, a quantum coupling term is calculated for each particle, taking into account the relative positions of all other particles in the swarm. This term is weighted by a coupling coefficient k, which is also controlled by the RL agent, and added to the velocity update as shown in Equation 5

$$\begin{aligned} v_i(t+1) = \omega v_i(t) &+ \varphi_p r_p\left(\text{pbest}_i - x_i(t)\right) \\ &+ \varphi_g r_g\left(\text{gbest} - x_i(t)\right) \\ &+ \kappa \sum_{j \neq i} c_{ji}\left(x_j - x_i\right) \end{aligned} \quad \text{(Eqn 5)}$$

The quantum coupling term, $\kappa \sum_{j \neq i} c_{ij}(x_j - x_i)$, models the influence of neighboring particles[7,15] and adjusts each particle's position based on the locations of nearby drones. Here, c_{ij} represents normalized weights that control how strongly particle j affects particle i, while κ is a coupling coefficient dynamically tuned[27] by the RL agent. This enables the swarm to investigate the search space thoroughly and avoid getting trapped into local optima[28,29]. This phrase permits the particles to take into consideration the positions of their neighbors, thus making sure that the swarm remains cohesive and at the same time, enlarges the search space. Positions are updated as in Standard PSO, equation 3. The wildfire environment used in this study is a simplified 2D simulation designed to test swarm coordination and adaptive optimization behavior rather than perfectly model real wildfire physics. Fire spread probabilities, wind effects, elevation effects, and natural extinguishing behavior were selected to create a controllable and computationally efficient testing environment. Therefore, the results should be interpreted as comparisons of algorithm performance within this simulation rather than direct predictions of real-world wildfire suppression outcomes.

Conducting the Experiments

Simulations run for 500 steps with drones deployed after fires spread for 10 steps. The RL agent continuously observes swarm state and updates PSO parameters and quantum coupling in real time. Learning continued during the evaluation runs, meaning the RL agent updated its Q-table online throughout the simulations rather than using a separate pretrained policy. Drone performance is evaluated based on distance to fires, proximity to water sources, remaining water, and local elevation. Metrics include fire extinguished, water usage efficiency, and swarm convergence. Total fire extinguished refers to the cumulative number of burning cells put

© The National High School Journal of Science 2026

out by drones across the full simulation. Because fires can continue spreading during the run, this value can be higher than the initial seven seed fires.

A fixed swarm size of 20 drones is used in all experiments. The particles' starting positions are generated uniformly and randomly within the grid's limits while their starting velocities are generated randomly too. The runs are repeated several times to ensure reliability in the presence of stochastics. With each run, the simulation generates a new scenario with a new terrain height, wind pattern, water source locations and starting fires. Drone starting positions are also randomly assigned within the grid for every run. No fixed random seed is used, so each method is tested under independently generated conditions rather than identical starting setups. This reflects realistic wildfire scenarios where conditions are unpredictable but introduces variability that is reduced by averaging over 50 trials.

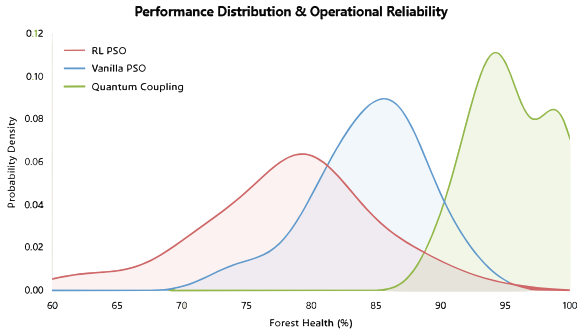


Figure. 1 Kernel Density showing the percentage of healthy trees left after 50 simulation runs for three algorithms: Vanilla PSO, Reinforcement Learning PSO, and RL Quantum Coup. PSO. This shows how well each algorithm protects the forest and how much the results vary between runs.

Results

This experiment compared three swarm algorithms, Standard PSO, RL PSO, and RL Quantum Coup. PSO to determine the impact of coordination strategies on fire suppression in the simulated environment. All algorithms were run 50 times each under the same conditions to make sure that the differences observed were due to algorithm efficiency.

Performance was measured using a range of performance indicators: average number of simulation steps, fires left unattended, number of fires extinguished, running time and percentage of trees survived. One simulation step corresponds to one cycle of the swarm behavior and environmental dynamics change. Operation time was estimated separately in seconds using drone travel distance, drone speed, water refill duration (30 seconds), and fire extinguishing duration (40 seconds). Tree survival was measured as the percentage of healthy trees. Table 1 summarizes average results across all trials.

RL Quantum Coup. PSO achieved the best fire suppression and tree preservation, leaving the fewest fires unextinguished (2.80) and maintaining the highest tree survival (95.37%). Standard PSO completed simulations in the fewest steps (145.06) and shortest time (817.85 seconds) but extinguished fewer fires (19.90) and left more fires remaining (3.34). RL PSO extinguished more fires than Standard PSO (36.90) but required the most steps and time (175.62 steps, 1557.73 seconds) and had slightly lower tree survival (77.48%).

Probability density plots of tree survival (Figure 1) show consistent patterns across trials. Vanilla PSO centered near 82%, RL PSO near 77%, and RL Quantum Coup. PSO is near 95% with slightly wider variation. Vanilla deviations and 95% confidence intervals further support these observations, indicating the superior and consistent performance of RL Quantum Coup. PSO.

Confidence interval analysis supported these observations. 95% confidence intervals were calculated using the parametric formula $\bar{x} \pm 1.96 \times s/\sqrt{n}$ where $\bar{x}$ is the sample mean, s is the sample standard deviation, and $n = 50$ trials. Confidence interval analysis supported these observations. Vanilla PSO produced a 95 percent confidence interval of [82.87, 85.01]. RL PSO produced a wider interval of [74.84, 80.12], reflecting greater variation across trials. RL Quantum Coup. PSO achieved the highest interval range, [94.32, 96.42], with values consistently higher than those of the other two algorithms. These statistical results further support the improved tree preservation performance of RL Quantum Coup. PSO. To further evaluate whether the differences between algorithms were statistically significant, a one way ANOVA test was performed on the healthy tree percentages across all three algorithms. The ANOVA produced an F statistic of 157 with a p value of 23.28×10^{-27} indicating statistically significant differences between algorithm performances. Cohen's d effect size analysis ($d = 2.915$) also suggested a very large practical difference between methods. Tukey HSD post hoc comparisons showed that RL Quantum Coup. PSO improved healthy tree preservation by approximately 13.43% over Vanilla PSO and 18.7% over RL PSO. These statistical results further support the conclusion that RL Quantum Coup. PSO achieved the strongest and most consistent wildfire suppression performance within the simulated environment.

The paths of individual drones are shown in Figure 2. Every line in color indicates the route taken by each drone from its initial point indicated in black circles to its final point indicated in black squares. Blue squares are water refill points while red triangles are places where there are active fires. The green circles indicate places where the fires have been extin-

© The National High School Journal of Science 2026

Table 1 Performance Comparison of PSO Based Algorithms in Wildfire Mitigation

	Metrics of the 3 Algorithms					
	Avg. Simulation Steps	***Avg. Fires Remaining***	***Avg. Drone Extinguish***	***Avg Time (in seconds)***	***Avg. % of Healthy Trees***	***Avg. % of Dead Trees***
Standard PSO	145.06	3.34	19.90	817.85	83.94%	16.06%
RL PSO	175.62	5.52	36.90	1557.73	77.48%	22.52%
RL-Quantum Coup. PSO	158.92	2.80	39.52	1459.92	95.37%	4.63%

a. Results represent average performance across 50 trials for three algorithms, Standard PSO, RL PSO, and RL-Quantum Coup. PSO, evaluated on wildfire containment metrics including steps run, fires remaining, drone extinguishing, runtime, and forest health outcomes.

Table 2 Vanilla Deviation of PSO Based Algorithms Across Metrics

	Standard Dev. of Each Metric					
	Simulation Steps	***Fires Remaining***	***Drone Extinguish***	***Time (in seconds)***	***% of Healthy Trees***	***% of Dead Trees***
Vanilla PSO	3.57	3.34	3.46	3.97	3.43	2.99
RL PSO	3.14	3.72	3.37	3.42	2.87	3.01
RL-Quantum Coup. PSO	2.98	3.43	3.12	3.81	3.24	2.79

a. Standard deviation values computed over 50 trials for Vanilla PSO, RL PSO, and RL-Quantum Coup. PSO across key wildfire containment metrics: steps run, fires remaining, drones extinguishing, runtime, and forest health outcomes

Table 3 95% Confidence Intervals Across All Performance Metrics

	95% Confidence Intervals of Each Metric					
	Simulation Steps	***Fires Remaining***	***Drone Extinguish***	***Time (in seconds)***	***% of Healthy Trees***	***% of Dead Trees***
Vanilla PSO	[143.33, 146.79]	[2.94, 3.74]	[19.22, 20.58]	[814.30, 821.40]	[82.87, 85.01]	[14.99, 17.13]
RL PSO	[172.19, 179.05]	[4.72, 6.32]	[34.82, 38.98]	[1545.16, 1570.30]	[74.84, 80.12]	[19.88, 25.16]
RL-Quantum Coup. PSO	[156.66, 161.18]	[2.50, 3.10]	[38.19, 40.85]	[1452.58, 1467.26]	[94.32, 96.42]	[3.58, 5.68]

a. 95% confidence intervals (95% CI) for key performance metrics across all algorithms in the wildfire simulation. Values are computed using $\bar{x} \pm 1.96 \times \frac{s}{\sqrt{n}}$ where n=50n = 50n=50 trials. Metrics include simulation steps, fires remaining, drone extinguishing actions, runtime, and forest outcomes (healthy and dead trees).

guished by drones while the pink circles are places that have burnt down naturally. The paths are concentrated near water sources and fire clusters, showing that drones adjust their routes based on environmental conditions. The paths show that drones adjust their movement based on fire locations and available resources, with RL Quantum Coup. PSO showing more structured and consistent coverage of active areas compared to the other methods. In the RL Quantum Coup. PSO model, κ represents the overall coupling strength between drones, while c_{ij} represents the adaptive interaction weight between individual particles based on distance and fitness similarity.

© The National High School Journal of Science 2026

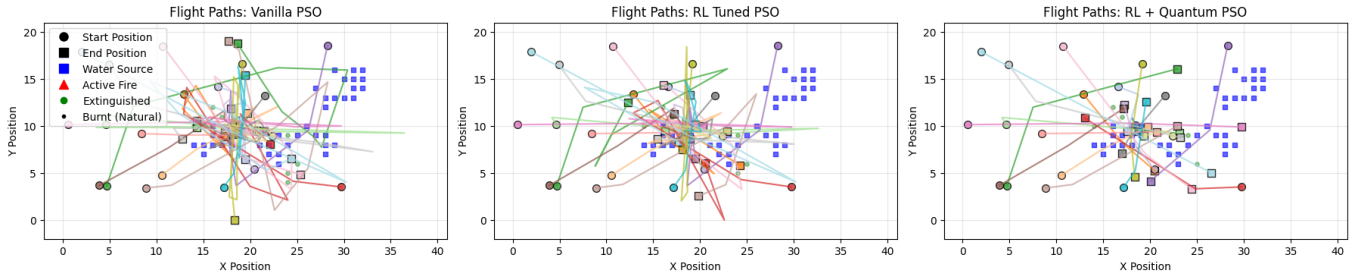


Figure. 2 Flight paths of drones during the wildfire simulations using Vanilla PSO, RL PSO, and RL Quantum Coup. PSO. Colored lines show individual drone paths, black circles mark starting points, black squares mark end points, blue squares show water refill spots, red triangles are active fires, green circles show fires successfully put out by drones, and pink circles show fires that burned naturally.

To better understand the qualitative trajectories shown in Figure 2, the analysis looks at how drones spread across fire areas and water sources over time. Although no formal swarm metrics such as path length or coverage fraction were calculated in this study, the visual results still help show clear differences in how each algorithm explores and covers the environment.

Discussion

PSO shows significant differences in approaches used to solve the problem of wildfires.

Vanilla PSO had the highest speed and used the least amount of operations, performing 145.06 operations and requiring 817.85 seconds for each iteration. On the other hand, it turned out to be the worst fire extinguishing method among the three as it extinguished only 19.90 fires in average and left 3.34 wildfires without extinguishment. The survival rate of the trees was pretty high as it was 83.94 percent. At the same time, confidence intervals were quite narrow. RL PSO showed another pattern of performance. It took the largest number of operations to complete one iteration (175.62) and required the largest operational time of all algorithms (1557.73 seconds), while extinguished the largest number of wildfires (36.90). However, left some of the wildfires without extinguishment (5.52). Relatively narrow confidence intervals show consistent performance throughout each run of RL PSO. One may suggest that using reinforcement learning gives ability to drones to adapt to the changing situations. But while using reinforcement learning, the approach resulted in extinguishing the biggest number of fires than others, more wildfires remained active during the time of simulation, and survival of trees dropped compared to other approaches. It can be concluded that using reinforcement learning, the drones were working with the fires only when they spread out instead of stopping it from spreading at the first stage. In addition, it is probable that regular parameter tuning performed by the RL algorithm caused more spreading of the drones. As a consequence, drones extinguished more fires, but wildfires were spreading to new areas of the forest.

RL Quantum Coup. PSO had the best results in this test. It extinguished the largest number of fires (39.52), left the smallest number of wildfires without extinguishing (2.80), and provided the highest survival rate of the trees (95.37 percent). Quite close probability plots and constant confidence intervals of approximately 1.761 prove reliable performance of the algorithm. It seems that quantum coupling helped particles to find the solution and converge faster.

The run time (1459.92 seconds) was the biggest comparing to Vanilla PSO but was shorter comparing to RL PSO.

Still, this algorithm has limitations. First of all, the wildfires environment used for this particular experiment is not a full-fledged simulation. Rather it is two-dimensional and is aimed at testing swarms behavior and their ability to implement adaptive optimization. In this regard, the following factors were chosen, fire spread probability, wind influence, altitude influence and natural extinguishing. Furthermore, it is easier to implement drone behavior in simulation, because the velocity of drone movements, its extinguishing time and time of refills is not limited by any battery restrictions, communication delays and other effects related to weather changes. It means that the results can be considered only relatively to other algorithms within the simulation and cannot be viewed as predictive modeling of fire suppression in reality. Moreover, it should be noted that computation time for thirty drones simulation implemented in this research work was manageable but bigger number of members of a swarm might require much more time. Thus, it needs to be explored, how would computation time increase in terms of increase of a swarm members' quantity.

The findings make it evident the usefulness of reinforcement learning for the swarm. However, this work makes com-

parison of PSO approaches only and does not consider the other non swarm approaches. Therefore, the results can show advantages of one approach over others but not advantage of PSO compared to all of the other approaches that might be used for firefighting in a situation of wildfire development. It can be an area for further studies.

Thus, implementation of reinforcement learning allows us to have adaptive[31] individual decision-making while quantum inspiration provides global coordination of a flock, reducing unnecessary movements and ensuring the systematic coverage[24]. It allowed obtaining good results in terms of fire suppression and saving trees. The areas for further exploration can include optimization of computation time and implementation of a 3D simulation, taking into account time delays in communications, drone crashes and battery constraints. An ablation study for both reinforcement learning and quantum inspiration was not performed in this study.

Conclusion

Thus, the present paper shows the impact of applying reinforcement learning algorithm as well as quantum inspired coordination in order to increase efficiency of fire suppression by drones in the simulation setting. While the standard PSO is rather quick but not flexible enough, and the reinforced learning one is more flexible but less efficient than usual. So, the hybrid version of RL Quantum Coup PSO has proved its advantages regarding flexibility and convergence providing the more efficient results of fire suppression.

Combining decision-making approaches with swarm coordination brings about the greater efficiency of wildfire suppression as well as trees preservation. This reflects the possibility of using reinforcement learning as well as quantum inspired coordination in order to increase the efficiency of fire suppression in a changing simulated setting. At the same time, this simulation includes very simple 2D setting of wildfires propagation and drones operation.

Acknowledgements

I would like to thank Grégoire Bergamo of Harvard University for his valuable guidance and mentorship throughout this research project.

References

1 L. Merino, F. Caballero, J. Martínez-de-Dios, J. Ferruz, A. Ollero. An unmanned aircraft system for automatic forest fire monitoring and measurement. *IEEE Journal of Intelligent & Robotic Systems*, 2012, DOI:10.1007/s10846-011-9604-2. https://DOI.org/10.1007/s10846-011-9604-2.

2 J. Kennedy. *Swarm intelligence*. Morgan Kaufmann Publishers, 2001.

3 E. Asri, Z. H. Zhu. An introductory review of swarm technology for spacecraft on-orbit servicing. *International Journal of Mechanical System Dynamics*, Vol. 4, No. 1, pp. 3–21, 2024.

4 A. Arokia Nathan, R. J., I. Kurmi, O. Bimber. Drone swarm strategy for the detection and tracking of occluded targets in complex environments. *Communications Engineering*, Vol. 2, Article 55, 2023, DOI:10.1038/s44172-023-00104-0. https://DOI.org/10.1038/s44172-023-00104-0.

5 C. Diaz Vilor, A. C. Lozano, H. Jafarkhani. A reinforcement learning approach for wildfire tracking with UAV swarms. *IEEE Transactions on Wireless Communications*, Vol. 24, No. 4, pp. 2766–2782, 2025, DOI:10.1109/TWC.2024.3524324. https://DOI.org/10.1109/TWC.2024.3524324.

6 G. Beni, J. Wang. Swarm intelligence in cellular robotic systems. *Robots and Biological Systems: Towards a New Bionics?*, Vol. 102, 1993, DOI:10.1007/978-3-642-58069-7_38. https://DOI.org/10.1007/978-3-642-58069-7_38.

7 M. Mannone, V. Seidita, A. Chella. Modeling and designing a robotic swarm: A quantum computing approach. *Swarm and Evolutionary Computation*, 2023, DOI:10.1016/j.swevo.2023.101297. https://DOI.org/10.1016/j.swevo.2023.101297.

8 A. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley, 2005.

9 M. Dorigo. *Optimization, Learning and Natural Algorithms*. Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

10 J. Kennedy, R. Eberhart. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942–1948, 1995, DOI:10.1109/ICNN.1995.488968. https://DOI.org/10.1109/ICNN.1995.488968.

11 R. Poli, J. Kennedy, T. Blackwell. Particle swarm optimization: An overview. *Swarm Intelligence*, Vol. 1, 2007.

12 Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, R. G. Harley. Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 2, pp. 171–195, 2008.

13 Q. Meng, K. Chen, Q. Qu. PPSwarm: Multi UAV path planning based on hybrid PSO in complex scenarios. *Drones*, Vol. 8, No. 5, Article 192, 2024, DOI:10.3390/drones8050192. https://DOI.org/10.3390/drones8050192.

14 F. van den Bergh, A. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 225–239, 2004.

15 A. Thomas, A. Farinelli, et al. Swarm robotics and quantum computing in simulated environments. In A. Farinelli, et al., Eds., *AIRO@AIIA 2023*, CEUR Workshop Proceedings, Vol. 3686, 2024.

16 S. Yang. Non-stationary problem optimization using the primal-dual particle swarm optimizer. *IEEE Congress on Evolutionary Computation*, 2003.

17 D. Bratton, J. Kennedy. Defining a standard for particle swarm optimization. *IEEE Swarm Intelligence Symposium*, 2007.

18 W. Li, Y. Xiong, Q. Xiong. Reinforcement learning guided particle swarm optimization for multi-objective UAV path planning. *Symmetry*, Vol. 17, No. 8, Article 1292, 2025, DOI:10.3390/sym17081292. https://DOI.org/10.3390/sym17081292.

19 Y. Shi, R. Eberhart. A modified particle swarm optimizer. *IEEE ICEC*, 1998.

20 M. Clerc. Discrete particle swarm optimization, illustrated by the traveling salesman problem. In *New Optimization Techniques in Engineering*, Springer, 2004.

21 Yin, S., Jin, M., Lu, H., et al. (2023). "Reinforcement-learning-based parameter adaptation method for particle swarm optimization." *Complex & Intelligent Systems*, 9, 5585–5609.

22 X. Hu, R. Eberhart. Adaptive particle swarm optimization: Detection and

© The National High School Journal of Science 2026

response to dynamic systems. *Congress on Evolutionary Computation*, 2002.

23 R. Sutton, A. Barto. *Reinforcement Learning: An Introduction*. 2nd Edition, MIT Press, 2018.

24 F. Tzoumas, A. Diamantopoulos, F. Sapiński, A. Kalpoutzakis. Wildfire detection in large-scale environments using force-based control for swarms of UAVs. *Swarm Intelligence*, Vol. 16, pp. 123–143, 2022, DOI:10.1007/s11721-022-00218-9. https://DOI.org/10.1007/s11721-022-00218-9.

25 S. Blackwell, J. Branke. Multiswarm optimization in dynamic environments. *Applications of Evolutionary Computing*, Springer, 2004.

26 H. Wang et al. Reinforcement learning based adaptive particle swarm optimization. *Expert Systems with Applications*, 2022.

27 S. Zhang et al. Hybrid reinforcement learning and PSO for dynamic optimization. *Applied Intelligence*, 2023.

28 J. Sun et al. Particle swarm optimization with particles having quantum behavior. *Congress on Evolutionary Computation*, 2004.

29 M. Clerc, J. Kennedy. The particle swarm: Explosion, stability, and convergence in a multidimensional complex space. *IEEE TEC*, 2002.

30 J. Sun, W. Fang, X. Wu. Quantum-behaved particle swarm optimization. *Evolutionary Computation*, 2004.

31 A. Ratnaweera, S. Halgamuge, H. Watson. Self-organizing hierarchical PSO with time-varying acceleration coefficients. *IEEE TEC*, 2004.

© The National High School Journal of Science 2026