METHANE PLUME LOCALIZATION WITH ENHANCED SELF-BEST REDUCTION AND GAUSSIAN IMPROVED PARTICLE SWARM OPTIMIZATION (GiPSO)

LIEW JIA JUN

DISSERTATION SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN
COMPUTER SCIENCE (BY RESEARCH)

SCHOOL OF ENGINEERING AND TECHNOLOGY
SUNWAY UNIVERSITY
MALAYSIA

2024

Table of Content

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate : Liew Jia Jun

IC /Passport No : 940515 – 06 - 5007

Student ID : 12025094

Name of Degree : Masters Degree in Computer Science (By Research)

Title of Project Paper/Research Report/Dissertation/Thesis (“this Work”):

Methane Plume Localization With Enhanced Self-Best Reduction and Gaussian Improved Particle Swarm Optimization (GiPSO)

I do solemnly and sincerely declare that:

1. I am the sole author/writer of this Work;

2. This Work is original;

3. Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;

4. I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;

5. I hereby assign all and every right in the copyright to this Work to the Sunway University (SunU), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of Sunway University having been first had and obtained;

6. I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by Sunway University.

Liew Jia Jun
liewjiajun

Candidate’s Signature Date:27/3/2023

Subscribed and solemnly declared before,

Witness’s Signature Date:27/3/2023

Name : Prof. Yap Kian Meng
Designation : Main Supervisor

**Witness will be the supervisor*
(This form to be attached together with the final hardbound thesis)

METHANE PLUME LOCALIZATION
WITH ENHANCED SELF-BEST REDUCTION AND
GAUSSIAN IMPROVED PARTICLE SWARM OPTIMIZATION
(GiPSO)

ABSTRACT

Swarm intelligence is a branch of artificial intelligence that studies the collective behavior of groups of social animals such as birds, fish, and bees. It has been used to solve various dynamic problems, including gas leak detection in drone-based leak detection platforms. However, gas plume dispersion has dynamical characteristics often influenced by external environmental factors such as wind direction, wind speed, dispersion rate and gas density. To investigate the adaption of swarm intelligence with dynamic modification to further enhance its capability to optimize gas plume dispersion.

The research focuses on three questions to enhance the drone swarm optimization algorithm. These three questions steer the research in three separate domains, which helps the evaluations of the performances of our research. The research question, problems and objectives will be the research directed toward modifying Particle Swarm Optimization (PSO), namely Gaussian improved Particle Swarm Optimization (GiPSO).

Firstly, how can swarm intelligence aid in engaging dynamically challenging optimization problems such as gas plume dispersion? To investigate this, our research will investigate the adaptation of the Gaussian gas plume in the

simulation. Adapting the Gaussian gas plume model in the simulation provides the experiment with a realistic optimization problem for GiPSO to optimize in the simulation, where we can test the engagement of dynamically challenging optimization problems such as gas plume dispersions.

Secondly, our research questions how the gas Gaussian gas plume model can address the adaptation of swarm intelligence in drone-based gas leakage detection. To address swarm intelligence adaptation in drone-based gas leakage detection, we investigate the existing swarm intelligence capability in optimizing dynamical problems in gas plume detection. Our research employs Gaussian improved Particle Swarm Optimization (GiPSO), derived from modification implemented on Particle Swarm Optimization (PSO) with Z-axis coefficient clamping and Self-Best reduction mechanism. Z-axis coefficient Clamping provides safety and reduction of drone swarm controlled by GiPSO risk with the physical collision with petroleum refinery exhaust.

Finally, the third question of our research is how the gas leakage detection algorithm's performance can be improved when the drone population is low. This guides the research investigating how population growth can impact GiPSO in Optimising Dynamic Problems. To enhance the performance of

population study in GiPSO, the GiPSO self-best reduction mechanism allows GiPSO to re-disperse the swarm when the same particle has retained as the global best as it achieves the limitation controlled by the operator.

The highlight of our algorithm, GiPSO, exhibits improvement in optimizing the source of leakage in high precision Objective Function Value (OFV). As the experiment setup benchmark specification of DJI Phantom 4 available flight time, GiPSO shows improvement with high success in localizing the source of leakage with population performance peak with 14 particles used in the drone swarm. These further answers our third research question concerning the performance of GiPSO with low particle population.

Keywords:- Gas Plume, Swarm Intelligence, Artificial Intelligence, Drones

(465 Words)

ACKNOWLEDGEMENT

First and foremost, I want to thank my thesis advisor, Dr. Yap, whose expertise and guidance were invaluable to me throughout this process. I am grateful to Dr. Steven and Prof. Qiang Ni for their time and effort in assisting me in refining my research questions, developing my methodology, and shaping my argument. They pushed me to think critically and deeply about my research, and their feedback was always constructive, thoughtful, and insightful.

In addition, I'd like to thank the staff and the School of Engineering and Technology for their assistance throughout my degree program. The department's resources and opportunities were invaluable in assisting me in developing my research skills and deepening my knowledge of Artificial Intelligence.

Furthermore, I'd like to extend my gratitude to Sunway University for the full scholarship in funding the research. The scholarship has provided me with state-of-the-art equipment for providing idealistic results throughout the research.

Thank you for all of your help and advice throughout this process. I am grateful for the opportunity to work with such talented and dedicated people, and I look forward to expanding on the skills and knowledge I have gained through my studies.

Sincerely,

Liew Jia Jun

LIST OF FIGURES

LIST OF EQUATIONS

List of Abbreviations

1. PSO = Particle Swam Optimization
2. ABC = Artificial Bee Colony
3. ACO = Ant Colony Optimization
4. Gbest = Global Best
5. OFV = Objective Function Value
6. GiPSO = Gaussian improved Particle Swarm Optimization
7. Q1 = Quarter 1
8. Q2 = Quarter 2
9. Q3 = Quarter 3

LIST OF TABLES

1. INTRODUCTION

Current gas leakage detection solutions can be mainly categorised into four approaches: infrastructure-mounted sensors, pipeline leakage detection micro-robots, personal handheld devices, and leakage detection drones. However, these solutions need to be more effective. Mounted sensors [1] [2] [3], as well as handheld sensory devices such as infrared methane sensors [4] [5], can aid in overcoming the limited effectiveness and outreach of gas leakage detection solutions. For example, a leakage in inaccessible pipelines can be solved by using remote micro-robots [6] [7] [8] to carry out inter-pipeline inspections to determine the severity of the leakage. These technologies are beneficial for infrastructure leakage and detection that are inaccessible to workers. Nonetheless, these platforms will not be efficient when covering large areas as more sensors and personnel are required to detect and localise the gas leakage source in the shortest time possible.

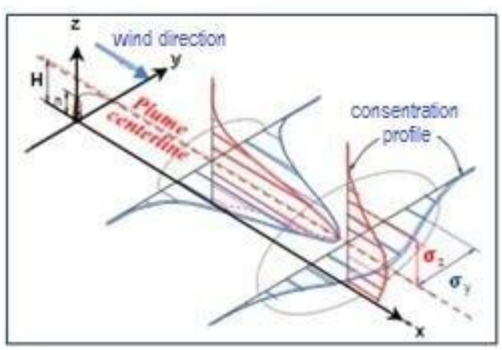

Figure 1-1 Three - Dimensional Gaussian Gas Plume Model

To enhance existing gas detection solutions, gaining an understanding of the dynamical characteristics of gas plume dispersion is crucial. Figure 1-1 shows the variables that impact a gas plume's dispersion, where essential factors such as wind speed and leakage stack height will change the time-averaged plume centreline and the time-averaged plume boundary [9]. The overall height of the plume centreline, H, is defined by the combination $h + \Delta h$, where h defines the stack height of the leakage source and Δh as the plume rises in elevation. σy and σz represent the coefficients by which the gas plume will expand according to normal distributions in both the y-axis and z-axis. The length of the plume, which is depicted as X in Figure 1-1, is determined by the wind speed factor that will affect the direction of X. According to Gaussian plume dispersion models, the concentration of pollution disperses from a source by spreading outwards along the centreline, C, of the plume under the influence of normal statistical distributions, σy and σz, in both horizontal and vertical matter.

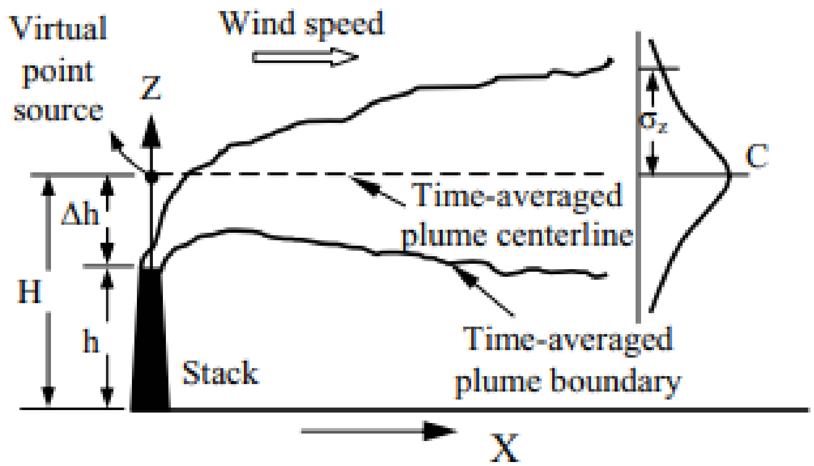

Figure 1-2Gaussian Gas Plume Model[9]

The Gaussian gas plume model corresponds to the natural phenomena that occur in plants' pollination process. This process is affiliated with how bees localise flower sources in the wild [10]. Bees, such as Bombus Impatiens and Eastern Bumblebees, sense the traces of pollen carrying negatively charged static, where the bees remain positively charged throughout the flight, as shown in Figure 2. When the flower releases streams of pollen that flow along the direction of the wind, bees will pick up and track the pollen traces, which eventually lead to the flower. The swarm is separated into two groups throughout the foraging stage: the scout and employed bees. Scout bees are tasked to survey for surrounding food sources, whereas employed bees forage for the best food source available.

Figure 1-3Bumble Bee and Flower Charges [10]

Likewise, this bio-inspired approach can be implemented on a drone-based detection platform. Drones have limited flight time due to their battery capacity; thus, this bio-spired foraging method can help to optimise the flight time constraint of a drone platform. A multi-drone platform [11] [12] may achieve better results. Drone swarm adaptation from bee swarm may offer a new method to improve upon the robot swarm approach used in detection solutions for larger detection areas. Incidents caused by gas leakage have been increasing, whereby the consequences of such incidents are rather severe and deadly. These incidents can be prevented through early detection, and preventive measures can be taken to reduce the likelihood of severe incidents occurring.

The sub-chapter is further subdivided in the Chapter 1 introduction section as 1.1, Research Questions, 1.2, Research Problem, 1.3, Research Objectives, and 1.4 Scope of Research. The focus of Section 1.1, Research Question, is on how the dynamical behaviour of a gas plume dispersion can be adapted to swarm optimisation algorithms as an optimisation problem. What methodology can we use to introduce gas plume dispersion in the simulated experiment to create a realistic situation for simulations to produce realistic results in the real world. The final research question emphasises the proposed solution's contribution to optimising the source of leakage as well as how the proposed solution approaches a dynamic environment.

The first question in section 1.1, Research Problem, focuses on integrating the dynamical behaviour of the gas plume dispersion into the swarm intelligence to improve its performance in optimising the gas plume in a dynamic environment. The research problems also highlight the swarm population's contribution and investigation towards implementing the proposed solutions. As a result, the research will have a structure for gathering results as well as benchmarking of the algorithm to further analyse the proposed solution's performance.

Section 1.2 Research Problem emphasizes on the problem for swarm intelligence to carry out optimization in three-dimensional environment as well ability to perform in optimization of dynamic challenge such as the gas plume dispersion. In this section, highlights such as design of the new optimization problem are being discussed in order to provide a realistic environment for the algorithm to further produce realistic results.

Section 1.3 Research Goal The section emphasises the research's goal of assisting in the resolution of dynamic environment issues using an evolutionary computing approach. The discussion in this section will revolve around the research focus and direction of tackling dynamic optimization of gas plume dispersion using a modified swarm intelligence algorithm.

1.1. RESEARCH QUESTIONS

Aside from the main objective of utilising an unmanned aerial vehicle with swarm intelligence integration for gas leakage detection, this research also aims to answer questions that will aid in further research on issues which revolves around the direction of optimization of group robots in gas plume dispersion optimization. The research questions are as follows:

1. How will swarm intelligence aid in the engagement of dynamically challenging optimisation problems such as gas plume dispersion?

Swarm intelligence is often engaged in static and spherical optimisation problems. However, swarm intelligence can be used to optimise best solutions, provided that it can adapt its search methods with progressive best candidates in static conditions. Hence, applying such logic can help to improve the swarm optimisation performance for detecting the source of gas leakage. The result of the swarm intelligence modification with dynamic factors from the Gaussian gas plume model can help the swarm to adapt to the gas plume behaviour. Consequently, the swarm's search and optimisation capability for gas plume detection can be improved.

2. How can the Gaussian gas plume model address the adaptation of swarm intelligence in drone-based gas leakage detection?

The Gaussian gas plume model represents the behaviour of real-world gas plume expansion with factors such as emission rate, pollutant density, and wind direction. As the factors vary with time, the optimisation problem for swarm intelligence changes from time to time. Adapting realistic optimisation problems to swarm intelligence in a simulation will help in the study of the swarm's performance in a real-world scenario, hence further justifying the performance of the swarm intelligence in collaboration with a drone-based gas leakage detection platform.

3. How can the gas leakage detection algorithm's performance be improved when the drone population is low?

The adaptation of swarm intelligence into a drone-based gas leakage detection platform helps the drones navigate in an organised manner with result-based velocity generation. As such, using the current global best candidate as a reference to generate the drone's next movement velocity will help the swarm to converge towards the source of leakage, as the global best candidate will be indicated by the highest sensory gas reading.

1.2. RESEARCH PROBLEM

From Section 1.2, there is a need to focus on issues related to swarm intelligence and drone-based gas leakage detection platform in optimising dynamic challenges such as gas plume and gas leakage. Specifically, the dynamical optimisation problems of real-world gas behaviour can be addressed by adapting an efficient detection platform to detect gas from higher altitudes. Hence, the problem statements of this research, along with the corresponding motivations, are as follows:

1. To investigate the adaptation of the Gaussian gas plume model in simulation to facilitate realistic optimization problem.

The Gaussian gas plume model facilitates the simulation of cloud-like optimisation problems with realistic plume behaviour, thus rending the optimisation problems to a realistic scenario of how a gas plume expands with dynamical wind behaviour. Hence, the time taken, which is determined by the algorithm, will facilitate the comparison of the results with real-world results with the adaptation of swarm intelligence and drone-based gas leakage detection platform.

2. To investigate existing swarm intelligence capability in optimising dynamical problems in gas plume detection.

The behaviour of gas plumes is unpredictable and dynamic. Such behaviour suggests that the early phase of the gas plume expansion may not have the same plume expansion characteristic as the late phases of the gas plume. As such, the optimisation problem formed can change drastically, affecting the optimisation time required to localise the source of leakage. Traditional methodologies for swarm intelligence are used to optimise static problems. Adapting the dynamic characteristic of the gas plume model can help the swarm intelligence to optimise dynamical problems, providing results that are relevant to realistic problems during experimentation investigations.

3. To investigate the drone population's impact on swarm intelligence in optimising dynamic problems.

Elements such as the growth in drone population suggest that larger coverage regions can be achieved simultaneously. A higher drone population will help to maximise the swarm's search capability compared to a lower drone population. Hence, the newly modified algorithm can help to reduce the population of the drone swarm to further improve the efficiency and productivity of the detection and optimisation of dynamic problems such as gas plume problems.

1.3. RESEARCH OBJECTIVES

This research investigates the benefits of dynamic swarm intelligence (SI) adaptation in drone-based gas leakage detection platforms for oil and gas industries. With the ever-evolving dynamical optimisation challenge, the algorithm must adapt to the nature of the gas plume's dynamic behaviour. The research objectives are as follows:

1. To develop modification to swarm algorithms to adapt to the dynamical challenges related to the gas plume characteristics.

To adapt to the dynamical characteristic of a gas plume expansion, the algorithm must first be able to function for a swarm of drones. Swarm intelligence controls the velocity generation for the drones' new movement based on the global best individual. Hence, to improve the performance of swarm intelligence for a group of drones, employing concepts such as z-axis clamp and global best control can further enhance the swarm control of the movement velocity.

2. To represent the actual gas plume behaviour as an optimisation problem in a simulated environment.

An optimisation problem based on the Gaussian gas plume model will be considered to represent the dynamic challenges related to wind factors. In other words, the optimisation problem in the simulated environment will represent realistic parameters and characteristics of real-world gas plume behaviour. To further enhance the realism of the results, parameters will be derived, and performance will be benchmarked based on marketable solutions such as drone simulation parameters. With the modification of the swarm intelligence for swarm control and its adaptation to drone platforms to optimise realistic problems, realistic results for real-world problems can be acquired in a simulated environment.

3. To employ dynamical control for swarm algorithms in controlling the movement velocity of the drone swarm.

A crucial element that affects the drone swarm's performances and search capabilities at a particular time is the population size of the swarm. Utilising a threshold for the prevention of early local optima can improve the search performance of the swarm with iterative progressive improvements. Both the swarm crowd control and swarm size mechanisms can help in the reduction of the optimisation time, thus improving the drone swarm's capability to perform an efficient search in the shortest time possible.

1.4. SCOPE OF RESEARCH

With the identified research objectives and problems, the research scope will involve evaluating the algorithm's performances in a realistic 3D simulation environment. To evaluate the swarm intelligence algorithm's performance, modelling of the plume optimisation is required. The model should be based on how the plume can behave in a natural environment depending on its environmental factors and parameters, such as the density, as the gas density can dictate how the plume can expand.

In the experiment, the modelling of the gas plume will follow the Gaussian gas plume model's metaheuristic methods in collaboration with the point cloud functions in CoppeliaSim to simulate the gas plume as the objective source of leakage.

Once the gas plume optimisation problem is defined, a benchmark model for drones is required to create a time limitation with a realistic battery flight time as the threshold in localising the source of leakage. In this scenario, DJI Phantom 4 Pro is utilised due to its commercial availability and cost-effectiveness compared to the majority of the industrial drones available in the market. According to its technical specification, the available flight time for Phantom 4 Pro is about 30 minutes.

With the Phantom 4 Pro GPS horizontal error range specification of ±1.5 metres, the objective function value (OFV) will be separated into three error range quartiles away from the radius of the actual source of leakage, which are:

1. Q1 = 1.125 metres error range
2. Q2 = 0.75 metres error range
3. Q3 = 0.375 metres error range

These quartiles allow for further analysis of how the algorithm will perform with different distance errors and how accurate the algorithm can be in achieving the closest possible distance to the actual source of leakage. For the definition of success in localising the source of leakage, any successful attempts within the battery threshold of 30 minutes are considered a success.

The algorithm selected to test the capabilities of Phantom 4 based on its suitability to the study will then be further studied via a literature review before the modification and implementation of the algorithm for the research. In the following section of the thesis, a literature review on a few swarm intelligence algorithms will be presented to provide further insights.

2. LITERATURE REVIEW

Birds are among many animals which are often observed for their feeding and migration behaviour. It is stated by James Kennedy, et al. [13] that the model relies heavily on the manipulation of inter-individual distances which are synchronously also affected by the flocking behaviour of the flock. Such a natural phenomenon would allow continuous optimization with reduced downtime Particle Swarm Optimization (PSO) takes inspiration from the bird optimizing method where each particle would mimic a bird that would optimize on their effectiveness yet influenced by the flock instinct. With leading influence, the rest of the group would be affected by the action of the leading individual during feeding scenarios. Such a scenario reoccurs when the food has depleted, and another food source has been located with a different new leader.

Optimization methods are also observed in other animals in the wild where populations are much larger and optimizing groups are larger in numbers as compared to birds. Bees are often observed for their unique organizations as well as group communications when they're harvesting for the hive. Ant Colony Optimization are as relevant as it shares a similar uniqueness in group communications, where communications are conducted in terms of pheromones which are left behind by passing ants. This insect group organization strikes interest by having optimized higher production in food by investing a lesser amount of energy spent. Often Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) are compared to solve Minimum Spanning Tree (MST) problems.

This shares relevance to natural optimizing problems where ants would scout for food and once a food has been located more ants would be attracted to the source thus building up a stronger concentration of pheromone. ACO mimics such foraging behaviour in the field of data handling [14] [15] and multi-sensory array handling [16] [17] [18].

ABC however, mimics bee foraging logic and methodology where a group of bees are separated into three different roles, namely employed bees, onlooker bees and scout bees. This behaviour allows ABC to execute optimization while seeking an additional food source. As compared to the ACO algorithm, ABC shows collaborative behaviour which was highlighted in [19], where ABC update each employed or onlooker bee at each iteration of ABC. Due to this collaborative behaviour of ABC, it is often applied in odour detection [20], gas detection [21] as well as optimizing structural damage detection [22].

Firefly Algorithm(FA) mimics different phenomena as compared to Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) which firefly mimics optimization based on common eastern firefly (*Photinus Pyralis*) mating behaviour. In the wild, female fireflies will be emitting lights on the tree branch while the males will be actively flying and searching for the brightest female.

The brighter the female emits the light, the better their genetics, as well as their health, therefore attract more males. Similar to the optimization problem, to attract other particles to influence one individual, that individual is required to have the "brightest" light, thus the rest would be influenced and attracted while the dimmer ones are lesser in fitness values. This method can be applied similarly to gas detection provided if one individual found a strong signal of gas detection the rest will follow and optimize in a similar pattern unless a brighter individual appears.

2.1. PARTICLE SWARM OPTIMIZATION (PSO)

The feeding phenomenon where when one individual found a new food source and the rest of the group will be influenced. This will continue as when the food sources deplete, and another individual found another new food source so will the rest follow. In PSO these traits can be observed in the formulation as shown in equation 1 where Inertia Factors, current motion self-confidence, swarm confidence and swarm influence share similarity with behaviour in natural phenomena. In the formula where C1 influences each particle individual improvements optimize the problem where C2 swarm influences coefficient where the whole swarm will be affected. As these coefficients are being considered, the motion and inertia factor would also affect how fast each particle will move as accordingly. This gives an idea where PSO executes as a group of collective independent particles influenced by the group's best individual to optimize a problem.

$$V_i = W_{vi} + c_1 r_1 \left(P_{best,i} - X_i\right) + \ c_2 r_2 (G_{best} \ - X_i)$$

Eq 2.1-1 Particle Swarm Optimization (PSO) formula [23]

Caption:

V_i = Velocity of the particle

W_{vi} = Inertia Weight

C_1 & C_2 = Acceleration Coefficient

r_1 & r_2 = Randomizing Numbers

P_{best} = Personal Best

G_{best} = Group Best

X_i = Position of individual particle

In nature, if one individual found a food that would influence both C1 & C2, with C2 involved, the rest of the particle would change accordingly with swarm influence as well as the change of C2, therefore, attracting other individuals to the best value. The value of C1 & C2 can be interchangeable or static values depending on the setup of the algorithm, with threshold control implemented, the values can be controlled with limitations thus influencing the swarm with collected results. This is useful in gas detection because as soon as the gas is detected the rest of the swarm will be affected by the individual that detected the concentration of the gas. While the rest of the individuals detects gas leakage, the best individual amongst the group will then be the individual which triggers the swap of with the current global best which will then influences the swarm population.

Tao ma, et. al. [23] have showcased a work where PSOs are used to locate gas leakage detection problems with multiple scenarios as well as obstacles. Four scenarios are used for the test as each scenario increases with its obstacles simulating the scenario which are often met on offshore platforms. In the testing PSO, ACO and Common Scrambling Algorithm (CSA) were used, and PSO performs consistently throughout all 4 scenarios. This proves that with obstacles involved, PSO can perform optimization without negative impacts on its performance.

B. Abhishek et. al. [24] proposed to have both PSO-HAS and PSO-GA hybrid algorithms for path planning in autonomous UAVs. Harmony Search Algorithm (HSA) Pitch Adjusting Rate (PAR) and Harmonic Memory Considering Rate (HMCR) are the two parameters within HSA that impact the solution's convergence speed. While PSO-GA hybrid is used to further improve the luck-based performance of traditional PSO optimization. The scenarios of the 3D environment are modelled with obstacles as well as a sphere which is there to behave as enemy detection radar.

Martin Saska et. al. [25] implementation of PSO by having a swarm of miniature quadri-copters carrying gas sensors having the reading of the gas sensor output as the fitness value of PSO and localising the source of gas by having PSO predict the source. A safety threshold was implemented into the algorithm for the drone's collision avoidance with one another. Results show that cooperative swarms of quadri-copters are efficient for plume detection scenarios.

Husanbir S. Panuu et. al. [26] implemented a solution having an adaptive neuro-fuzzy inference system (ANFIS) to enhance parameter selection for PSO optimization in benzene detection. The objective is to have ANFIS training with normalized parameters and through having the training PSO parameter will be selected accordingly to the current scenario. The result shows that ANFIS showed significant results compared to standard fuzzy and Neural Network based prediction.

These papers have highlighted where PSO can tackle dynamic problems in a 3-Dimensional environment as well as applicable to path planning for quadri-copters Martin Saska et. al [25] applications. PSO is also proven in Husanbir S. Panuu et. al. [26] and B. Abhishek et. al. [24]where PSO allows hybrid tuning to achieve better results than traditional PSO methods. This is due to PSO optimizing factors being based on randomizing factors as well as luck-based execution. These shows the ability where PSOs can perform in dynamical search of gas plume and utilize them for UAV gas detection Platforms.

In the following segments, we can further identify recent research conducted in latest 5 years in PSO. Each of the optimizations are further broken down into Author/year/reference number, types of algorithms, work done on the research, finding as well as challenges faced in the research. The purpose of the tabulated literature review is to provide an insight on highlight the hybridism or modification as well as modification on the algorithm to further improving optimization performances. Scope of the literature in the recent 5 years provides understanding and capability as well as understand of the PSO modification and enhancement trends.

Author / Year / Ref number	Algorithm Type	Work done	Finding	Challenges (Analysis of what their limitation)
Tao Ma / 2020 / [23]	PSO + ACO search method	• Modification of PSO with ACO initialization search methods with population control on drone swarm gas leakage detection with obstacle environment.	• The population of the swarm will directly impact the success of localizing of the gas source	• At lower population, the success rate to localize leakage reduces. As the population grows, the success increases.
Thangavelu Shankar / 2020 / [24]	PSO + Harmonic Search Algorithm (HSA)	• Using HSA to generate new harmony with ascending sorting to obtain new global best for PSO crossover while having PSO to induce Exploration capability	• The Outcome of the optimization of the hybrid algorithm outperforms PSO+GA hybrid which are often modified	• Shuffled Frog Leaping Algorithm (SFLA) drone receives lower power as compared to PSO HAS however SFLA travels much further distance to optimize as compared to PSO HSA
Martin Saska / 2014 / [25]	PSO with Smoke Source Prediction	• Implementing PSO into Micro Aerial Vehicle swarm for	• On the 18^{th} iteration of MAV PSO execution, the	• Utilizing light weight sensors causes uncertainties and measuring errors while detecting

		gas Localization while allowing Visual Relative localization or Collision Avoidance to interrupt with movements of the swarm.	swarm (population of 3) can localize the gas position with having the swarm to behave as dimensionless particle in PSO. • Second Experiment includes visual constraints and collision avoidance to interrupt with PSO, however results show that on 29th iteration of PSO execution, the swarm achieved localization.	gas concentration in the air. • Drones have the tendency to collide into each other due to the nature of particle behaviour in PSO where overlapping particles position occurs.

Hussanbir Singh Pannu / 2018 / [26]	PSO + Adaptive Neuro-Fuzzy Inference System (ANFIS)	• Implementing Artificial Neuro Fuzzy Inference System (ANFIS) into PSO with tuning on Antecedent and consequent parameters. On each executions iteration step of PSO, ANFIS evaluates if current found fitness exceeds the previous fitness and adjust ANFIS tuning if current fitness exceeds.	• Mean improvement of proposed model prediction accuracy are found to be improved at 1.18% • Differences of Actual value and predicted value of mean reduction based on real time air data was found at difference of 0.0635ppm. • Proving the capability of improving existing sensor mesh network with metaheuristic algorithm prediction collaborative with fuzzy and neural system.	• Practicality of implementing large number of sensor mesh network within urban regions can be extremely costly. Urban environments possess challenges such as high skyscrapers and highways with highspeed moving vehicle which can impact on the accuracy of the sensor readings.

Table 3.1-1 PSO work-done analysis

The following table 3.1-1 , literature highlighted in the table are compiled to be PSO research efforts within the recent 5 years period. The research efforts in the recent 5 years can be seen where PSO are still often used as the base for hybridism modification in reducing the size of swarm population while increases the productivity of the swarm performances. Within the compiled studies , majority of the research efforts shows promising results with high success rating for each of their experiments, however this are achieved with hybridism capability.

In order to further improve the capabilitty and efficiency of PSO , we have to further understand that PSO often has reduced chances of successfully localizing the source of leakage due to early convergence of the swarm to false positive location. By dispersing the swarm apart from false global best and leading to the false positive source of leakage, the swarm can further enhance its capability in localizing the source of leakage thus improving the ability to successfully localize the source of leakage with lower iteration and time taken required.

Title	Year	Published in	Modification Type	Methodology	Results and Analysis
Open Agricultural Burning Detection with Natural Inspired Swarm-based Detection Platform [27]	2022	IICAIET VOL 3	PSO + self-randomizer + z-axis reduction	**Modification:** Changing the early emergence of false local optimum with randomizing the global best particle if the threshold is achieved. The generation of z-axis velocity is checked to ensure the z-axis velocity is above the physical height threshold.	Modification of Gi-PSO focuses to have PSO jump out of early local convergence while having the generated velocity of the Z-axis for each particle to be above the limited physical height limitation. The improvements of GiPSO and PSO are that GiPSO is able to improve the chances of successfully localizing the source of leakage by 12.24% while reducing the chances of failure to localize the source by 14.35%
A Hybrid Algorithm for Gas Source Locating Based on Unmanned Vehicles in Dynamic Gas Environment [28]	2021	Mathematical Problems in Engineering Vol. 2021	PSO +Nelder Mead Algorithm	**Hybridism:** By cooperating early stages of a z-travel search, threshold PSO will be triggered upon locating gas concentration to further optimize the source of leakage. The optimization processes, Nelder -Mead Simplex Method (NMSM) will enhance its capability of optimization quality by checking the current solution with satisfaction with the NMSM threshold	The newly modified NMSM -PSO hybridism can efficiently reduce its search time and overall optimization time by 12.2% while reducing its iteration number to optimise the source of leakage by 27.6%.

Swarm Robot Implementation in Gas Searching using Particle Swarm Optimization algorithm [29]	2017	Computer Engineering and Application Vol.6 No.3	Applied PSO to real ground robots.	**Implementation:** Implementation of original PSO onto ground robots while having ground robot gas sensor to return the gas reading as fitness value. Gas is set up on each specific corner of the arena for the swarm to detect and optimize.	The optimum swarm performance can localize the source of leakage on the 20th second of the optimization attempts.
Hybrid Odour Detection System for Search and Rescue Robot Based on PSO [30]	2018	CHEMICAL ENGINEERING TRANSACTIONS Vol. 68	BP neural network + PSO	**Hybridism:** Having the BP neural network record and calculate its optimum weights and threshold to improve the upcoming iterative experiments. The more experiments run the better the outcome of the optimization capabilities.	The overall outcome of the BP neural network with PSO hybridism can achieve accuracy with errors of 8% in the detected values. As compared to the original bp neural network, the performance of PSO-BP further reduces the errors and further increases its accuracy.
A stochastic programming approach for the optimization of gas detector placement in offshore platforms [31]	2019	Ocean Engineering Vol. 187	Minimal Cumulative Detection Time model (MCDT) + PSO simulations optimization	**Hybridism:** By having MCDT handle the probability estimation, PSO further optimised the values to locate the optimum location for mounting gas sensors. The simulation grounds are using Computational Fluid	The experiment shows that the number of sensors increases with the reduction of numbers in Objective Function Value OFV. But upon reaching 1 OFV, the increment past 30 sensors remains stagnant.

				Dynamics (CFD)	
Inverse Frequency Response Analysis for Pipelines Leak Detection Using the Particle Swarm Optimization [32]	2016	International Journal of Optimization in Civil Engineering Vol. 6	Inverse Frequency Response Analysis (IFRA) + Particle Swarm Optimization (PSO)	**Hybridism:** Having IFRA as the objective function to predict frequency responses of the pipe in return for any random set of leak parameters. When the prediction is completed, PSO will be used to further optimize the source of the leak on the pipeline	When the error rate was compared based on actual leakage parameters and leakage size, it is shown that when there is only a single source of leakage, the error rate stands at 0.033%. When there are multiple sources of leakage, the highest error rate in prediction versus actual parameters is 0.6%, whereas the lowest at 0.067%.
A Pipeline Leak Detection and Localization Approach Based on Ensemble TL1DCNN [33]	2021	IEEE Access Vol. 9	Transfer Learning one-dimension convolutional neural network (TL1DCNN) + PSO	**Hybridism:** Utilizing PSO to further enhance the weight selection for TL1DCNN, PSO uses a combination of 50particles with 40 iterations to obtain optimal weight for TL1DCNN. Multiple TL1DCNN with classify and attempt to optimize the objective function, the return result will further be optimised again by PSO to group into the Ensemble TL1DCNN.	In this experiment, Ensembled TL1DCNN is compared with EMD-SVM, EMD-BP, WT-SVM, WT-BP, and 2DCNN. 3 different scenarios are used to test their performances. Ensemble TL1DCNN can achieve 100% optimization for no leakage, WT-BP and EMD-BP similarly has also achieved 100% for no leakage. However, accuracy for small leakage of TL1DCNN, it can achieve a score of 97.5%, EMD-BP achieved 83.9%, and WT-BP achieved 84.9%. This shows that Ensemble TL1DCNN achieves high scores with a low error rate compared to other proposed modified methods.
Modified PSO algorithms with	2018	Neurocomputing Vol. 292	Multiple Guaranteed	**Modification:** From Guaranteed	Several modified PSO was used in the experiment with different distance

"Request and Reset" for leak source localization using multiple robots [34]			Convergence Particle Swarm Optimization (MGC-PSO), Modification of having 2 different groups of particles in PSO based on their performance	Convergence PSO GCPSO mechanism to limit the failure and success mechanism if consecutive success, it will then swap to a higher scaling vector, if there are failures, the success counter will then be reset and the failure counters to increase, thus resulting in smaller scaling vector. The group is then separated into 2 different groups, particles with the lower fitness value will be grouped into the weaker group while the particles with a closer fitness value to the global best are grouped into the optimum group. The weaker performing group will update its' velocity based on the optimum velocity group.	scenario. Hybrid -PSO, MD-PSO, MDA-PSO, MGC-PSO. Upon comparison, hybrid PSO shares close results as compared with MGC-PSO, with implemented gaussian noise to the plume, Hybrid -PSO outperforms MGC-PSO due to the Mechanism in Hybrid PSO to have to reduce C1 while increasing C2 iteratively. Despite increasingly higher gaussian noise in the plume, MGC-PSO shows a minor increment from 0.00 m to 0.09 away from the actual destination. However, with a population size of 15 particles, MGC-PSO shows a slightly higher standard deviation as compared to Hybrid PSO at 20.94 vs 20.49, but MGC-PSO can achieve higher precision in 5.11 as compared to Hybrid-PSO at 6.57.
Novel leakage detection by ensemble 1DCNN-VAPSO-SVM in oil and gas pipeline systems [35]	2022	Applied Soft Computing Vol. 115	Modified PSO variant, Variable Amplitude PSO (VAPSO) + SVM +1DCNN	**Hybridism:** VAPSO is used for the improvement and optimization of parameters for the SVM model to prevent them from falling into local optima and affecting the overall	In the experiment, several pipelines with different numbers of kernels are used for comparison of the study. A few algorithms are also used to compare its accuracy in terms of its training set and test set experiment data.

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>performance of the model. SVM is then integrated with a deep learning model to substitute for the original fully connected layers (FCL) and softmax classifier

VAPSO
The inertia weights of the particles are based on SVM output values, whereas the optimization movements are based on traditional PSO methods.

The C1 max reduces slowly as iteration increases. Upon each Iteration starts, the C1 will return to its C1MAX and repeats its process to C1MIN.

C2 of the VAPSO utilize the reverse methods of C1 weight controlling, whereas as iteration increases the weight of C2 will increase from its C2Min to C2MAX.

This method ensures as longer iteration increases, in</td>
<td>
<table>
<tr><td>Algorithm Type</td><td>Accuracy training Set</td><td>Accuracy Test Set</td></tr>
<tr><td>1DCNN-VAPSO-SVM</td><td>97.72%</td><td>96.61%</td></tr>
<tr><td>1DCNN-LPSO-SVM</td><td>96.88%</td><td>95.76%</td></tr>
<tr><td>1DCNN-MPSO-SVM</td><td>97.61%</td><td>96.40%</td></tr>
<tr><td>1DCNN-NLPSO-SVM</td><td>97.56%</td><td>96.18%</td></tr>
<tr><td>1DCNN-TVACPSO-SVM</td><td>97.56%</td><td>95.82%</td></tr>
<tr><td>1DCNN</td><td>96.93%</td><td>94.92%</td></tr>
<tr><td>1DCNN - SVM</td><td>90.67%</td><td>87.30%</td></tr>
</table>
It is noticeable that by comparing to its’ predecessor, 1DCNN-SVM, 1DCNN-VAPSO-SVM can outperform with much higher accuracy in training data sets as well as accuracy with actual test data sets. 1DCNN-VAPSO-SVM has improved its accuracy with the actual test set from
</td>
</tr>
</table>

				the early stage of the experiment the swarm will exhibit more global search, whilst on higher iterations, the swarm will begin to exhibit more local search capability.	87.30% to 96.61%.
Research on oil-gas Pipeline Leakage Detection Method Based on Particle Swarm Optimization Algorithm Optimized Support Vector Machine. [36]	2021	International Conference on Education Technology Management VOL 2076	PSO +SVM	**Hybridism:** Traditional PSO were used for parameter selection for optimum parameters so SVM can use it for optimal prediction.	3 different algorithm were used to optimize the parameters for SVM to train on based on the gas sensor detection value, namely Particle Swarm Optimization (PSO), Genetic algorithm, and Grid Search Algorithm (GSO) The experiments are separated into 4 different sets of experiments, 18,28, 38, and 48 tests sets, each set is tested on each of the algorithms mentioned above, and the scoring of methods is scored based on accuracy and running time taken. On 18 test sets, PSO and GSO achieved 94.4%accuracy while GA achieved 88.89% accuracy, however, GA requires only 3.98 seconds to complete while GSO requires 9.50 seconds. PSO achieved the best score with 94.4% in 0.99 seconds. Such performance was shown in PSO achieving high

					accuracy with a shorter time, however, PSO performance in 48 test sets was reduced to 91.67% with 0.98 seconds, GSO can achieve 93.75% but a longer time is required at 8.51 seconds. GA achieved 83.33% accuracy with 2.99 seconds required on 48 test sets.

Table 3.1-2 PSO research in the recent 5 years period

2.2. ANT COLONY OPTIMIZATION (ACO)

Other than observing birds' feeding behaviour, scientists have long since observed and studied insects' behaviour and organisational phenomenon. Ants' communication is unlike insects such as fireflies or cicadas, where light signals or calls are being utilised for communication; due to their weak vision, communications are often carried out with pheromones. Once a scout ant finds a new and viable food source, it will return to the hive and attract more ants to follow them. Through this situation, more ants will leave stronger pheromones, signalling a food source is nearby, and more ants will follow to forage the food source.

$$p\left(c_i^j \middle| s_p\right) = \frac{\tau_{ij}^a \cdot \left[\eta(c_i^j)\right]\beta}{\sum c_i^l \epsilon N\left(s_p\right)\tau_{il}^a \cdot \left[\eta\left(c_i^l\right)\right]\beta} \text{ , } \forall c_i^j \epsilon N(s_p)$$

Eq 2.2-1 Ant Colony Optimization Selection Formula [38]

Equation 2.2-1 shows the optimisation where τ symbolises the pheromone on the arc and η represents the heuristic information. $N(s_p)$ are solution components set to maintain feasibility. α and β are the parameters with a fixed value in the initialising step, determining the relative importance of the pheromone value and heuristic information. p stands for the probability that ants would select the available path. This formula shows where the algorithm executes optimisation based on the strength of the available pheromone on the edge of each iteration.

The stronger the pheromone strength signifies the higher fitness value and more favourable solutions by the majority of the population. This is suitable for gas detection because when more particles sense gas in one direction, more will follow as the pheromone grows over time, thus leading to more wandering ants towards the source of gas leakage.

Tao Ma et. al. [37] showcased similar work as [23], but instead of testing the PSO platforms on an offshore scenario, ACO is utilised to test gas leakage detection on offshore platforms. Four scenarios are used for the experiment stage for optimising gas leakage, with each scenario increasing in the number of obstacles as well as the complexity of obstacles. Two results were obtained during this experiment, where with a single source of leakage, the success rate is much higher than the two sources of leakage. In single-source optimisation, multiple instances have achieved a 100% success rate with lower iteration required, whereas double sources only have one instance that achieved a 100% success rate.

Qing-Hao Meng et. al. [38] have an ACO hybrid with an Upwind surgeon robot swarm to optimise leakage source. This solution excels when the robot arrives where no pheromones are detected, which determines that no individual has discovered this location. If such a scenario occurs, the robot will scatter and move upwind to search for the last memory where gas was detected. The robots can optimise the source, and efficiency increases along with the number of populations, where the group size of three and four robots are able to optimise up to a 100% success rate. In contrast, a group size of 2 only manages 80%.

YuHua Zou et al. [39] introduced a modified ACO where it comes into three stages of execution for multi-robot odour source localisation. The first stage is Local Traversal Search, Global Search and Pheromone Update. Global Probability Search is highlighted as a modification due to that if the currently available edge has no pheromone for a specified time. It will perform a Global search where the stochastic location in the defined environment where concentration is higher. Contribution as such would aid ACO to improve further the tendency to lower random wandering if no gas is detected.

Tao Ma / 2020 / [30]	ACO modification with search space division	• Modification of recognizing the best candidate for the ACO swarm to follow as well as modifying to have all individual to move towards best robot for gas source confirmation. If the concentration of all the robots isn't the same as the best robot, the swarm will disperse out again for localization. Optimizing search space are separated out into search square spaces.	• Performance of robot swarm optimizing a single leakage source reduces with the growth of population size. However, the performance of robot swarm to optimize two leakage spots increases significantly as compared to a single leakage source.	• Rate of failure to optimize the problem increases due to with growth of numbers in population, other individual will not be able to achieve the same saturation as the best robot due to search space limitation thus failing the actual gas confirmation stage in Optimizing stage criteria to fulfil.
Qing-hao Meng/ 2012 / [31]	Adapted AACO + Upwind Surge (AACO +US) For Chemical Plume Tracing (CPT)	• AACO adapts the population of the swarm into several subgroups to carry out optimization. Lower fitness group will be attracted to follow subgroup with higher	• Incremental of success rate that first robot to approach the source grow alongside with the growth of population from 2 robots per group to 4 robots per group. However, group size of 3 has the highest	• Limitation on performance is depending on the size of the robot groups due to collision avoidance behaviour amongst each robot. Constriction of the search ground along with the increase in size of group population will constantly trigger collision avoidance thus affecting the efficiency

	problem	concentration. When no pheromone (Gas reading) was detected, these robots will scatter and move towards upwind. Subgroup with higher pheromone will surge upwind and explore more areas to prevent local optima prematurely.	success rate for all robots to achieve localization of the source.	and increment in failure rate.
Yuhua Zou / 2008 / [32]	ACO	• Modified ACO with local traversal search stage into each individual among Gas detection robot swarm. Gas concentration reading acting as strength of	• With implementation of proposed local search modification improved the performance of localizing the source in much shorter time as compared to original ACO. While the original ACO tend to	• If there are two different leakage sources while one has stronger gas concentration over the other, the robot swarm will confirm and localize source with the highest concentration while ignore the other leakage source. This is a challenge when the scenario has multiple

		pheromone which is then used for path selection by ACO. Local traversal search is done in 5 points which revolves around the starting location with calculation of each position around radius, *d,* around the centreline.	stray further away from each of the robots and has higher tendency to explore into environment randomly. The proposed local search methodology helps to reduce possibility of traditional ACO where each particle wanders randomly when no gas concentration was found. The movement path of modified ACO shows much orderly movement towards the source with increment of iterations.	leakage source to be optimized and localized.

Table 2.2-1 ACO work-done analysis

ACO can optimise problems with a higher concentration of data or an array of incoming data; however, ACO also exhibits random wandering where the particles roam in a random direction to prospect for potential pheromones. In ACO, optimisation is often carried out by identifying the path with the highest concentration of pheromone will be selected amongst the possible path selection. This algorithm shows an advantage where when there is a large amount of data, optimisation would be easier to carry out in return with the longer time consumed for the optimising problem as the algorithm utilise high counts of the selected path as the best path. This is beneficial in detecting the best solution to optimise a problem; if a problem requires a much shorter time, it may be challenging for ACO.

2.3. ARTIFICIAL BEE COLONY (ABC)

Eusociality in insects often relates not only to ants, but bees share similarities in their social behaviour. Throughout 22 days life span of a bee, they will carry out different tasks accordingly to their age. Scientists have been interested in their foraging group, where bees are separated into different groups for different stages of foraging tasks. It is separated into scouting bees, employed bees and the onlooker bee. Unlike ants, where each ant can be an onlooker ant or scout ant, bees each carry out tasks accordingly to time determined by the sun's location.

$$v_{mi} = x_{mi} + \emptyset_{mi}(x_{mi} - x_{ki})$$

Eq 2.3-1 Profitability of Food Source [41]

$$fit(\vec{x}_m) = \begin{cases} \dfrac{1}{1 + f_m(\vec{x}_m)} & if\ (\vec{x}_m) \geq 0 \\ 1 + abs\big(f_m(\vec{x}_m)\big) & if\ \ f_m(\vec{x}_m) < 0 \end{cases}$$

Where $f_m(\vec{x}_m)$ is the objective function value of solution $\vec{x}_m$

Eq 2.3-1 ABC Phase 1 Employed Bees [41]

Equation 2.3-1 shows the formula for the profitability of a food source, depicting the neighbourhood food source in bee swarm memory. x_k Symbolises the randomly chosen food source, i stand as a random parameter index and a random number within the range of [-a, a]. The formula is then calculated as shown in Equation 2.3-1.

Equation 2.3-2 shows that if given the results of the objective functions are larger or equivalent to 0, then the fitness values are calculated by one divided by 1+ fm(solution vector to be optimisation problem), else if it's less than 0, then one plus the absolute value of fm(solution vector to be optimisation problem).

$$P_m = \frac{f_m(\vec{x}_m)}{\sum_{m-1}^{SN} f_m(\vec{x}_m)}$$

Eq 2.3-2 ABC Phase 2 Employed Bees Optimization [41]

Equation 2.3-2 shows the second phase of ABC, where onlooker bees will take the results from Employed bees, where the probability is calculated by the fitness values provided by employed bees. $(\vec{x}_m)$ Indicates food source, which is calculated by the onlooker bee; once this probability of food source has been chosen, f_m which indicates the neighbourhood sources is then determined by using the equations; when the fitness value is then computed, more onlookers will be recruited to the richer sources with more positive feedback behaviour. This algorithm design allows the optimisation to select the higher neighbouring sources to be optimised when two neighbouring sources are found.

YingHao Zhao et al. [40] proposed combining the Tabu Search and Chaos Search methods into Artificial Bee to improve the searching efficiency further. As compared to traditional methods, the original roulette wheel selection has also been replaced with tournament selection to improve the global searching ability of Tabu Chaos Artificial Bee Colony (TCABC). Analysis of the experiment shows the performance of TCABC compared to actual measure damage is accurate, with minor error of 0.78% to 1.76% as the highest error rate.

Ye Jiang et al. [41] proposed a Weighted Global ABC (WGABC) that considers the global factor utilising the advantage of the global best solution to guide the search for a candidate solution. This work suggests ensuring a good convergence rate for local search; a smaller weight is much recommended, whereas larger weights ensure faster global convergence speed. Since smaller weights ensure a good local convergence rate, the weight coefficient should reduce as iteration increases. Traditional ABC and PSO were then used in the test to compare with the performance of WGABC. WGABC shows that the detected gas concentration errors reduce as more cycles occur; GABC and traditional ABC exhibit similar behaviour. Still, WGABC shows the lowest error percentage, as low as 0.007%, ABC shows 0.015%, and GABC shows 0.012%.

Aveek Dutta et al. [42] have applied ABC to multi-sensory arrays to detect and compare tea quality. In this study, each employed bee is assigned to a single food source to be optimised. Five sensors are utilised for the experiment, and each experiment instance is set up with controlled parameters. Fuzzy C-Means (FCM) algorithm is compared with ABC for the training set and test set for the accuracy of results. Four sets of tea taster scores are allocated for training and testing; ABC results for the experiments show consistency in accuracy compared with FCM, where taster Score 3 shows 18.75% in error scoring, and 6 FCM is unable to obtain optimisations with 100%. ABC shows an 11.25% error in taster score three and an 11.66% error rate in taster score 6. This shows that FCM needs to distinguish between two types of tea samples, whereas ABC satisfactorily classifies these two kinds of tea. The classification of the results is shown according to the correctness of the total samples detected, where overall ABC achieved 96.55%, and FCM only obtained 75.86% correctness.

J. Enríquez-Gaytán et al. [43] integrated optimisation methods where K-means cluster the data clusters before it is used for ABC optimisation. K-Means algorithm suffers from being stacked into “local – minimum”, where ABC overcomes these issues by having scout bees seek out other potential food sources. This allows the algorithm to optimise the existing minimum food source and seek potential global food sources. The design of this hybrid algorithms function by having a K-Means algorithm to reduce the dimension from 360 x 8 to 360 x2. After the K-Means algorithm downsizes the dimension, ABC will optimise for the centre of the data cluster.

In TCABC [40], Tabu searches are separated into two parts, namely T1 & T2, where T1 stores the recently visited food source, preventing bees from revisiting the same source. T2, on the other hand, stores the location of deserted food sources; this can aid the bees in the selection of adventuring towards new food sources than optimising the existing food sources. As for the tournament selection strategy, each time two food are being compared, the ones with the highest fitness will then gain 1 point while the ones with the least fitness will receive 0 points; at the end of this strategy ones with the highest points will receive the higher probability of being selected as the best food source.

In these experiments, it can be observed that ABC can tackle global minimum while exploring possible neighbouring food sources. ABC exhibits group collaboration while optimising the peak problems; this shows the algorithm's ability to differentiate and optimise the "food source", which is more beneficial with higher profit yields.

Yinghao Zhao / 2020 / [34]	ABC + Tabu Search Method	• Two list is introduced into ABC for improvement of food source searching as well as better generation of next step movement. First list, T1 records the memory of recently found food source. When the number of list exceeds T1 list, the first recorded food source in T1 will be replaced. While the second list, T2 holds the memory of food source which are explored several times without improvement, thus deserted by the bee's particle.	• TCABC shows consistency in performance of low statistic as compared to Traditional ABC in overall Standard Deviation as well as lower worst solution. However, performances for best solution, TCABC has also improved from ABC performances throughout different tests scenario.	• The experiment study suggests a scenario where search area is fixed into segmented sensory proximity. with fixed detection area, increase in number of sensors will result in neighbouring minimal value which are often known as local optima amongst two sensors.
Ye Jiang / 2016 / [35]	Weighted Global ABC (WGABC)	• Controlling gas sensory particle in MATLAB with GABC which initializes with large initial weight coefficient, as iteration increases, the weight coefficients reduce to assure of better local	• Population of particles remains as 20 particles throughout comparison with WGABC, PSO and ABC. WGABC has performed consistency in optimizing function with low cycles required as compared with PSO and	• With consideration of reduction in weight coefficient, the paper suggests a consistent number of particles at 20 particles to test WGABC performance. A study based on increment of particles impact on efficiency of performance is needed as at higher iteration count, the weight

		search capability •	ABC.	coefficient of ABC will be reduced accordingly.
J. Enríquez-Gaytán / 2020 / [37]	ABC + Principal Component Analysis (PCA)	• Using PCA for marking readings of gas sensors and then having ABC to find the centre of the cluster data. 3 main gas reading are focused which is Ethylene, Methane, and carbon monoxide.	• The experiment shows that ABC can locate the centre of the clustering data, ABC optimization results are relatively close to PSO performances showing the possibility of using ABC as an alternative for cluster centre optimization.	• With the differences in distancing between different types of gases to optimize, when two cluster of gas reading are closer, ABC may not be able to achieve centre optimization for two different gases as particle will fall into local optima scenario.

Table 2.3-1 ABC work-done analysis

2.4. FIREFLY ALGORITHM (FA)

Eusociality behaviour doesn't exist only in organisational insects such as bees; wild mating behaviour also encourages swarming behaviour. In fireflies, swarming behaviour is observed during night-time, when female fireflies emit light on the trees to attract male fireflies for mating. This is because female fireflies emerge without wings, while male fireflies emerge as adults with wings. The attractiveness of a firefly is determined by the cold light emitted on the abdomen; the brighter it is, the more attractive one mating partner is.

The same concept is applied in the FA optimisation algorithm, where the higher the fitness value, the brighter the individual becomes. In the swarming behaviour of FA, other individuals are influenced by the brightness of an individual; once a brighter individual occurs, the rest of the group will be influenced by the brightest individual and cause the ones with the least brightness to converge towards the brighter individual.

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + a \in_i^t$$

Eq 2.4-1 Firefly Algorithm [46]

Equation 2.4-1 shows the formula of the firefly optimisation algorithm; formula α indicates the scaling factor, which controls the step size of individual random walks, while γ is a scale-dependent parameter which controls the visibility of the firefly. $\beta_0 e$ indicates the attractiveness constant when two fireflies are zero above the equation [44]. $\in_i^t$ indicated in the formula indicates the random factor where its drawn from normal distribution. x_i^t indicates the current position of the firefly , whereas x_j^t indicates the brighter individual in the swarm.

KuangWei et al. [45] have proposed a modified FA by limiting the step size selection for the algorithm during optimisation; this step size is controlled in a range. As the number of iterations increases, the step size will reduce accordingly. These aid sizes are much larger in the early optimisation steps; explorations are encouraged as the iterations continue, and reducing the step size will aid in fine-tuning the efficiency in optimising the problem. The selection of the best individuals is also improved by introducing an adaptive circle around the best individual, and this circle will decrease accordingly as the iterations increase. Such improvements tackle the problem where the “brightest individual” may not be in the optimal position. This modification reduced the number of iterations required for optimising peak problems. The improved Firefly Algorithm also exhibited reduce in Average Operation time compared to the traditional Firefly Algorithm and levy flight Firefly Algorithm.

Deng Long Ma et al. [46] have highlighted where in traditional Passive Firefly Algorithm (PFA) forms a new group by updating the parameters of each individual through a comparison of each brightness. Active Firefly Algorithm (AFA) is introduced in response to PFA limitations by adopting Gaussian Normal Distribution. Standard Deviation of normal distribution of distances of the current individual compared to the brightest are also adopted. With this, comparison and updating of each brightness and parameter are not required as each parameter is updated as the algorithm optimises the peak problems. Consistent parameter updates are optimised, resulting in AFA producing skill scores lower than PFA in 3 of 4 release cases. This proves that AFA can obtain a more accurate estimation than PFA, even with smaller populations.

Due to the nature of FA, where the brightness of each individual is required to be evaluated, and adjustment of parameters is required before carrying out the next step, this will consume more time required in optimising problems compared to PSO and ABC. FA shows potential to take on dynamic problems as the behaviour changes and optimal solution changes, as well as the experiments, are carried it; this shows promise with its capability to take on dynamical problems. However, modification in parameters update will also improve the time required for the updating process to tackle problems which weigh heavily on time taken, such as localisation on gas leakage sources

2.5. RECENT WORK ON PSO RESEARCH IN THE PREVIOUS 5 YEARS PERIOD

Title	Year	Paper Highlight	Conclusion
A review of swarm intelligence algorithms deployment for scheduling and optimization in a cloud computing environment [47]	2021	A review study on Swarm Optimization in cloud computing: • PSO • ACO FA	Hybridism of PSO and Modification of PSO parameters are often brought up in the modification of PSO.
Particle Swarm Optimization and Fuzzy Logic Control in Gas Leakage Detector Mobile Robot [48]	2015	Implementation of the ground robot with gas sensor collaborative with PSO and Fuzzy logic. Simulation Work Using a sensor to determine the “fitness” value	Utilizing fuzzy logic to rule the movement control about velocity generated by PSO.
Performance Analysis of the PSO algorithm: an experiment study [49]	2021	From our experimental analysis, we conclude that the PSO protocol gives better performance overall if many particles are present in the swarm and the algorithm run for enough time to provide optimal results, but sometimes the target cannot be found. In other words, it can be concluded that the behaviour of the PSO is improved if sufficient input is provided for the algorithm to run; however, the nodes may not always be able to find the target.	PSO performances are better and rather consistent if the population are larger and as iterative execution increase over time, the performance will improve accordingly.

Application and improvement of swarm intelligence optimization algorithm in gas emission source identification in the atmosphere [50]	2018	Three different algorithms are used to compare performances. Firefly Algorithm (FA), PSO, ACO	PSO outperforms in terms of boundary constraints and computational efficiency. ACO and FA perform poorly in larger boundaries compared to PSO, but when the boundaries reduce ACO, FA performance improves. Note that is suggested that to improve the efficiency for these SI to perform better in Atmospheric optimization, location parameters are required to improve the computational efficiency.
A Novel Method for Source Tracking of Chemical Gas Leakage: Outlier Mutation Optimization Algorithm [51]	2021	To improve the performances of current swarm intelligence algorithm performances and avoid local minimal which most SI will fall into in the earlier iterative optimization stages. Objectively is to optimize gas plume optimization problems.	Outlier Mutation Optimization (OMO): Utilizes exploration and exploitation concepts for optimization.
UAV swarm control strategies: A case study for	2017	Implementing PSO onto a group of pixhwak drones mounted with a gas sensor. With drones	3 improvements are made before the execution of the

leak detection [52]		naturally having physical inertia weight, each drone will be evaluated according to their detected concentration of gas where the highest gas is grouped best and update the PSO formula accordingly.	optimization process: 1. Separation Moving away from each other 2. Alignment Align towards the average heading of others 3. Cohesion Group closer to form a swarm
A Hybrid Algorithm for Gas Source Locating Based on Unmanned Vehicles in a Dynamic Gas Environment [53]	2021	Hybridism between PSO algorithm to optimize problem while Nelder Mead Simplex Method (NMSM) to check if optimization of NMSM optimized global best can surpass the current global best. If the reflected NMSM output does not surpass the existing global best, the current solution will be updated instead	The performance of PSO+NMSM reduced maximum time and maximum iterations as compared to standard PSO. In Standard PSO maximum time increases as the size of the group grows, however, PSO+NMSM maximum time reduces as compared to standard PSO.
An improved particle swarm optimization method for locating time-	2019	Modified PSO with additional Acceleration coefficient with upwind velocity. The experiment is conducted in Computational	The new Modified PSO with Upwind velocity successfully leave local

varying indoor particle sources [54]		Fluid Dynamics (CFD) simulations. A new search strategy is implemented where a random search strategy will be activated if the plume is lost, or Modified PSO will continue to take effect.	extremum areas and rapidly locate time-varying particle source in an indoor environment. The new method has improved with a success rate above 96% with a localization time of approximately 55 seconds.
Modified PSO algorithms with “Request and Reset” for leak source localization using multiple robots [55]	2018	Changing the strategy of the Standard PSO optimization method where if 50% of the current particle fitness is lower than the average fitness, randomly reset their position and velocity to be sent away from the global best to explore while another half of the population follows the historical globally best particle to strengthen local search around the best particle. If the new optimum group’s global best particle fitness is lower than the previous, the existing optimum group will be reset.	In comparing performances between hybrid PSO and MGC-PSO in terms of population growth, standard deviations of MGC-PSO reduce along with the increment in populations.
A multi-information fusion ‘‘triple variables with iteration’’ inertia weight PSO algorithm and its	2019	Implementing Linear Decreasing Inertia Weight (LDIW) to handle iterative control for weight to improve exploration and exploitation.	The cases of controlling the maximum and minimum coefficient of the inertia weight are judged by case

application [56]			matching of the successfulness of each of the particles. More consecutive success will reduce the maximum coefficient while lower to no consecutive successfulness will increase its maximum intertie weight coefficients.

A Review of Representatives Swarm Intelligence Algorithms for Solving Optimization Problems [57]	2021	Review Studies on current trends and research contributions from 2000 to 2020 on different algorithms category: • Ant Colony Optimization • Artificial Fish Swarm • Particle Swarm Optimization • Bacterial Foraging Optimization • Artificial Bee Colony • Other swarm intelligence algorithm The analysis review in this paper also categorises this analysis for types of research this algorithm is used to solve problems. The categorization following the trend as follows: • Scheduling Problems • Power Systems • Parameter Optimization • Image Processing • Signal Processing • System Identification Robot System	

Table 2.5-1 PSO studies in the past 5 years

In the table 2.5.1 – 1 PSO studies in the past 5 years compiles the research efforts which are done on PSO in the research field. From this table we can further understand that PSO are often used for parameters optimization or hybridism tactic is used to further enhance the capability of PSO. Paper such as Y. YUTNG ET. AL. [34] takes on approach such as request and reset where if the results of the swarm are below satisfactory the swarm will then be communicated to dispersed and reset with randomized location . This can be utilized to further prevents the probability of drone swarm spiralling towards false localization.

To conclude the literature review, we can observe that optimization algorithms such as PSO, Firefly Algorithm(FA), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) retains their capability in localizing and optimizing of a provided issue. When the research objective are applied to further filters algorithms to be chosen, PSO better suits the objective of the research as FA, ACO, and ABC requires larger populations to perform optimizations.

Furthermore, when PSO are compared with other algorithms, we can see that PSO are better suited to take on dynamical challenge of the methane plume as the plume would change alongside with the increment of time. Algorithm such as FA, ACO and ABC are harder to manoeuvre to dynamical optimization as the algorithms would have multiple stages of optimization and layers of communications.

The research focuses on improving the performance of swarm-based optimization with lower populations size on the swarm .While the intention is to improve the productivity of swarm algorithms , we have to also satisfy the criterion of reducing the time taken for each successful optimization to further proves the reliability and applicability to realistic problems possessed by gas plume dispersion as well as the availability of flight time restricted by drone-based gas leakage detection platform.

With considerations of both criteria, algorithms with high population swarm size and long optimization time will not be feasible as well as applicable to the research. This will conclude that algorithms as such as Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) will not be selected as both algorithms will require high swarm size particle populations as well as longer progressive optimization time to further improve their optimization methods as it has multiple optimization stages. Firefly Algorithm (FA) are computationally least complexity as compared to Particle Swarm Optimization (PSO) , however, the convergence time for Firefly Algorithm (FA) are slower as compared to Particle Swarm Optimization (PSO).

Firefly Algorithm (FA) low convergence originates from the optimization methods where the particles are often travelling along in one direction. Provided that the optimization such as gaussian gas plume model which changes dynamically with the increment of time , Firefly Algorithm (FA) are not suitable for the experiment. Particle Swarm Optimization (PSO) is a population-based optimization algorithm where the size of the population will manipulate the efficiency of the swarm, such factor will satisfy the requirements of the research focus.

Furthermore, Particle Swarm Optimizationg (PSO) are often implemented for dynamical challenges as Particle Swarm Optimization (PSO) are good for multi-objective optimization. To localize the source of leakage, multiple particles are required to compare and optimize for the best global best individual selection where it will then influence the swarm newly generated velocity movement. This will benefit the maneuver behavior to challenge optimization problem such as gas plume dispersion problem.

3. METHODOLOGY

Due to the nature of the gas plume expansion and drone detection platform utilised, simulation is selected as this study's methodology, as more results can be obtained autonomously. The intention of modification of Gaussian Improved Particle Swarm Optimization (GiPSO) focuses on the communication between the individual swarming behaviour as well as dispersion on threshold. While having intentions to maximize the capability of GiPSO with lower population count on the swarm , the global communication of the drone swarm controls is required to be focused on as well. As the population grows in the experiment , the communication between each individual and swarm as a while requires a thresholding to further prevents the possibility of having false global best individual thus leading towards false positive localization.

Upon investigation of PSO behaviour with dynamically evolving optimization problem such as the gas plume model, PSO has high tendency to fall into false local optima. This is due to plume may shift direction as the factor , t, increases over time of the experiment. As the optimization continues with the time factor , t, random shifts of the plume will affect the PSO as the same current global best individual may retain as the false best individual thus leading the swarm into failure to localize the source of leakage.

To improve such issue, limitation such as re-dispersion of the swarm can be utilized to further reduce the tendency of the swarm falling into false local optima and encourages wilder pattern search when the same individual retains as the global best. As the swarm move towards the source of leakage, the probability of other individual may detect higher concentration thus becoming the current global best, this scenario will then deactivate such mechanism unless the same global best has retain until the threshold.

The experiments are designed to reduce the local minima by applying a threshold to how often one individual can remain as the global best and improving the search performance with the integration of the Gaussian gas plume model. For each completion of the experiment, the results will be logged into the comma-separated value (CSV) file according to the instance count, iteration count, categories, and how many iterations are needed until the particles localised the source.

To further evaluate the performances of GiPSO's ability to optimise the global optima of the gas plume dispersion, Objective Function Value (OFV) and time limitation to conclude a success with the low time taken are indicated based on the specification of DJI Phantom 4.

The objective Function Value of our research focuses on how much horizontal range versus precision we can apply to evaluate further the performance of GiPSO and PSO compared to its ability to pinpoint the source of leakage. The categorisation of OFVs is further separated into 3 different quarters to evaluate the performance of GiPSO and PSO in each OFV to be 25% closer to the source of leakage to satisfy successful criteria. The OFV for the population study is standardised to be the source of leakage without the horizontal error range of DJI Phantom 4. This can help us to study how GiPSO can perform with growth in population in localising sources of leakage with high precision.

The categorisation of results into successful, high iterations and failure is based on the time taken to localise the source of leakage. Suppose the experiment instance can localise the source of leakage within 150 instances of movements. In that case, it will then be categorised as successful localisation as it would take less than 35 minutes within the available flight time of a DJI Phantom 4. However, experiments that succeeded in localising the source of leakage within 150 to 300 instances of the experiment will then be categorised as high iterations.

This is because the time taken for declaring a high instances success would take 35 to 45 minutes, which is the cut-off limit for DJI Phantom 4 flight time. Failure instances are categorised for any experiment exceeding 300 instances in attempts to localise the source of leakage.

The primary focus for the categorisation and limitations for OFV and success indication is based on the specification of DJI Phantom 4. This allows our research to produce realistic results and restriction to the actual scenario in the real-world environment.

3.1. GiPSO PSEUDOCODE

```
GiPSO Initialize Population

While ( Counter < Counter_max )

        If particle (x) ( pbest(x) > gbest)) Then ( pbest(x) = gbest)

        End

         For (step size < Δt )

                If t < t_max

                        Then x_{i,d} = x_{i,d} + V_{i,d}

                Else

                        V_{i,d} = Wv_{i,d} + c_1 r_1 (p_i − x_{i,d}) + c_2 r_2 (p_2 − x_{i,d})

                        If V_2 > σHs Then V_2 = abs (V_2) − σHs )

                        Else V_2 = V_2

                End

                For x = 1 : population size

                Switch(Gbest)

                Case 1:

                Gbest(x − 1)! = Threshold(n): GbestCount = GbestCount +

                Case 2:

                   GbestCount = Threshold(n): Gbest = PSO GENERATE

          PSO VELOCITY GENERATE: V_{i,d} = Wv_{i,d} + c_1 r_1 (p_i − x_{i,d}) + c_2 r_2 (p_2 − x_{i,d})

                GbestCount = 0


            End

        End

                If x_{i,d} = Objective_Max

                Then break

End
```

Figure 3.1-1 GiPSO Pseudocode

The design of GiPSO focuses on preventing the swarm algorithm from falling into the local optimum in the early stage, which will cause the swarm to fail in localising the global optimum.

As shown in GiPSO consists of a self-best reduction mechanism where when the same individual has remained the global best for too long, the global best will be randomised to a random individual, and the swarm is dispersed again. This mechanism will not be triggered in the later stages when the swarm is closer to the source of leakage. The closer the swarm is to the source, the more individuals will consistently replace the global best individual.

$$\boldsymbol{V_{i,d} = Wv_{i,d} + c_1 r_1 (p_i - x_{i,d}) + c_2 r_2 (p_2 - x_{i,d})}$$

$$\textbf{If } V_2 > \sigma Hs \textbf{ Then } V_2 = abs\,(V_2) - \sigma Hs\,)$$

$$\textbf{Else } V_2 = V_2$$

Eq 3.1 -1 GiPSO Physical Stack Limitation

Aside from the local optimum prevention, GiPSO also includes a mechanism to prevent the swarm from flying into a danger zone in the real world as shown in Equation 3.1-1. The mechanism will allow the user to set the physical stack height of a real-world chimney as the minimum height threshold.

This threshold will help to prevent the swarm from venturing into areas below the height of the chimney; as methane gas (CH4) is lighter than air, it will disperse at a higher height than the chimney height. This mechanism serves not only as a safety feature for the swarm to avoid collision possibility but also helps the swarm to avoid unnecessary ventures to places where there may not be trails of gases.

```
Traditional PSO
While ( Counter < Counter_max )
    If particle (x) ( pbest(x) > gbest)) Then ( pbest(x) = gbest)
    End
     For (step size < Δt )

            V_i,d = W v_i,d + c_1 r_1 (p_i − x_i,d) + c_2 r_2 (p_2 − x_i,d)

        End

        For x = 1 : population size
            If particle (x)(fitness > pbest(x)) Then (fitness(x) =
pbest(x))
                counter = counter + n
         End
        End
    End
    If x_i,d = Objective_Max Then break
End
```

```
Gi-PSO
While (Counter < Counter_max )
        If particle (X) (pbest(X) > gbest)
            Then (pbest(X) = gbest)
        End
        For (step size < Δt )
            If t <Δ tmax
                Then X i, d = X i, d + V i, d
             Else
                V i , d = Wv i , d + C1R1 (P I – X i , d) + C2R2(P2 – X i , d)
                    If V2 > σ Hs
                        Then V2 = abs (V2) – σHs)
                    Else V2 = V2
                    End
        If gbest ( X - 1 ) != gbest(X)
           Then gbestCount = 0
        Else If (gbestCount = 0)
                Then gbestCount + n
        Else
             Gbest = V i , d = Wv i , d + C1R1 (P i – X i, d ) + C2R2(P2 – X i , d)
        End
        For x = 1 : population size
              If gbest ( X - 1 ) != Threshold(N)
                          Then gbestCount = 0
              Else If (gbestCount = 0)
                          Then gbestCount + n
              Else
                  Gbest = V  = Wv  + CR (P I – X I , d ) + CR(P – X  )
              End
        End
        If X i,d = ObjectiveMax
            Then break
        End
End
```

Figure 3.1-2 Traditional PSO vs GiPSO

In Figure 3.1-2, the differences between GiPSO and traditional PSO can be observed in terms of their pseudocode representation. In the GiPSO pseudocode, the modifications done for the movement velocity clamp based on the z-axis control and self-global best reduction mechanism are highlighted.

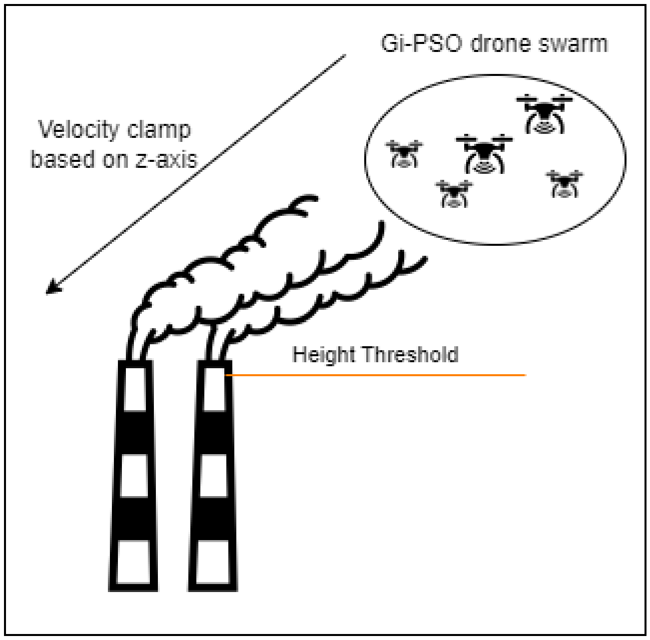

Figure 3.1-3 GiPSO modification visualization

By understand the Gaussian Gas Plume model , we identified that the plume behaves as such that it would expand in a enlarging conical shape. Thus , the direction fashion of the plume goes in a decreasing fashion from the end of the plume towards to the source of leakage.

While reducing the generated z-axis velocity for each of the particles with z-axis velocity clamp we have to ensure that that newly generated z-axis velocity should also be above the physical height clearance , h , to ensure the drone swarm does not collide with potential pipelines or chimney.

As the runtime of the experiment , t, increases the generated z-axis will gradually be reduced for each of the generated velocity thus increasing the chances of having the drones to be on parallel level with the gas plume. This ensures that the drone swarm remain above the physical clearance while generating reduced z-axis velocity similarly to gas plume formation.

When a new set of velocities is generated on each iteration of movement, the z-axis movement velocities should be checked if the newly generated velocities are above the physical height threshold. The conditional checking ensures that the swarm does not generate a z-axis velocity that is lower than the threshold provided by the operators. This mechanism represents a safety precaution for the swarm in reducing the possible risk of physical collision and providing a limitation for the swarm to conduct optimisation in regions without gas plume expansions.

3.1.1. GLOBAL BEST THRESHOLD

GiPSO modification focuses on two key elements, as highlighted in gaussian gas plume model where , where plume raise from the source of leakage , Δh, and z – axis of plume raised from plume center height , σz, are the factors that drones can exploit to improve the efficiency further in detecting and localising the source of leakage. The coefficient of z, σz, spreads larger the further the drone travels away from the gas source. This characteristic inspires the modification made by iteratively controlling the generated z-axis position for the drone swarm. With this modification, the drone will eventually reach a level where the swarm will carry out optimisation on the centreline of the plume with the highest chance of localising the source of leakage.

$$Gbest = \begin{cases} Gbest = 0 \ , & if\ Gbest < n \\ emptyCount + 1 \ \ \ , & if\ Gbest == n \\ emptycount = \ \emptyset, Gbest.random, & Otherwise \end{cases}$$

Eq 3.1.1-1 Self-Reduction of Global Best Reduction

One of the challenges faced by traditional PSO occurs when a particle has been recognised as Gbest and no new gas particle is detected. Gbest will influence every other particle to steer towards the similar “lost” path, resulting in a high number of iterations and yet being unable to localise the leakage source.

This problem is alleviated with the modification of the algorithm by limiting how often one particle can be Gbest by setting a threshold to control it. The notation n can be changed accordingly to the desired threshold that the user prefers. If any Gbest particle is maintained as Gbest for more than n, then the Gbest particle is randomised to a new Gbest.

This modification is essential to encourage the overall swarm to follow Gbest and to remove any inconsistent Gbest in the group due to the sensory value as the gas plume model may produce “chaotic” gas plumes. The “chaotic” characteristic of the gas plume model is defined as the inconsistency of gas concentration in the plume, which is unlike a real-world plume, where the gas concentration is shared along the plumes.

A class known as self best reduction function is created to keep track of the global best in case the global best has remained as the same individual for a fixed number of iterations. The use case of this condition checks if one individual has remained as the global best individual for this fixed number of iterations. The counter will be reset, and another randomised individual will be the new global best. When each iteration is completed, the counter will increase until the threshold has been reached.

The modification to randomise Gbest will aid not only in encouraging more adventuring to seek for the gas plume but also in preventing the same candidate from remaining as Gbest for too many iterations, as the swarm being unable to detect gas traces for too many iterations will cause stagnation in the optimisation process.

3.1.2. Z-AXIS COEFFICIENT CLAMPING

After implementing the self-reduction function, the overall PSO-generated z-axis is relatively randomised. The study on the Gaussian plume model suggests that the further the plume travels away from the source of leakage, the more σz increases significantly; thus, this characteristic can be utilised for a modification in PSO to control the generated z-axis for the optimisation process.

$$newVelocity\ (\sigma z) = \begin{cases} \sigma z - \Delta h & , \quad if\ \ abs(\sigma z) > \Delta h \\ \sigma z = newVelocity(\sigma z) & , \quad if\ abs\ (\sigma z) < \Delta h \end{cases}$$

Eq 3.1.2-1 Condition for Z-axis Reduction

This modification is to encourage the iterative reduction of the newly generated velocity on the z-axis to reduce the coefficient of z accordingly, hence reducing unnecessary exploration and bringing the swarm closer to the source of leakage.

According to the Gaussian gas plume model, the closer the plume is to the source of leakage, the lower the coefficient of the z-axis. Thus, reducing the newly generated z-axis velocity will help to increase the performance when localising the source of leakage. The Move_particles() class cross-checks the 3rd element of the newly generated velocity for each particle, ensuring the regulations of reducing or maintaining the generated value are followed. The condition functions in such a way that if the absolute value of the newly generated z-axis velocity is higher than Δh, then the new velocity is reduced accordingly.

The benefit of this modification is that the PSO goes closer to the source of the plume the longer the iteration goes on in the optimisation of the source of leakage. An increment in the number of iterations also helps the particles to be closer to the source, thus increasing the chances of having more particles to optimise the problem.

3.2. GAUSSIAN GAS PLUME MODEL

To simulate dynamical problems, such as the plume dispersion in a natural environment, the Gaussian gas plume model can be used as a reference to define the function for the PSO to optimise. The Gaussian gas plume model shows that Δh dictates the centreline of the plume from the stack height, while σz dictates the dispersions of the plume away from the centreline as the distance increases away from the source.

A modification was done on a chemical plume in Kok Seng Eu's experiment [58], where a randomising factor was included in generating each new experiment instance. This scenario mimics a real-life scenario where gas leakages may not originate from the same source. The randomisation process includes randomising the direction and Δh such that each experiment will be tested with unique settings of plume dispersions.

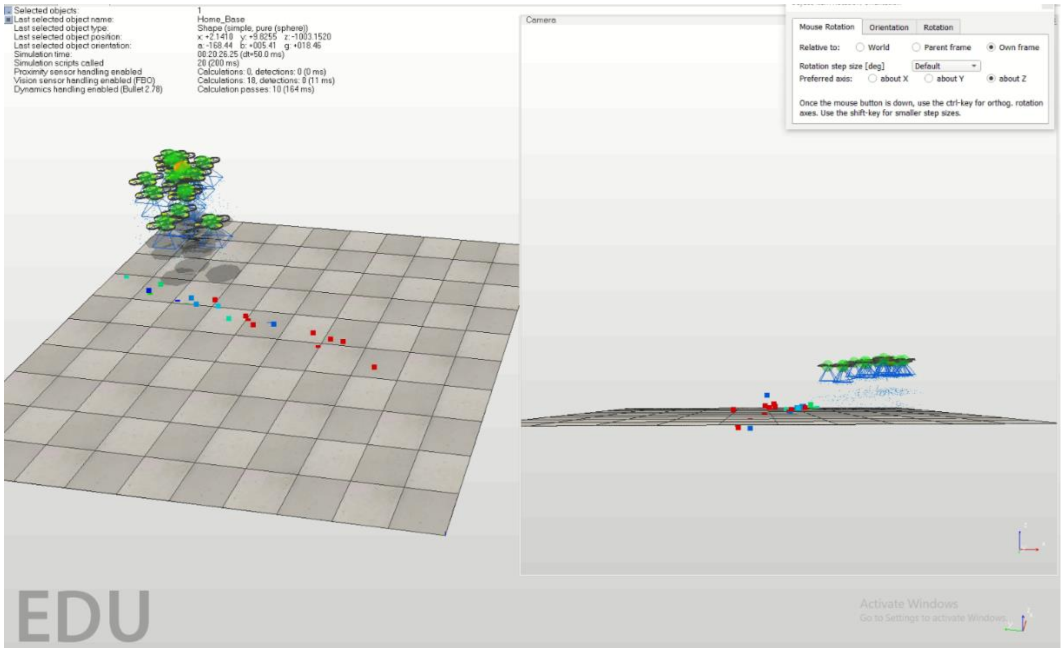

Figure 3.2-1 Gaussian Gas Plume Model in Coppeliasim

The Gaussian gas plume model is set up such that when each instance of the experiment restarts, the parameters such as the plume height threshold, wind direction, and emission rate will be changed accordingly. Hence the change of these factors provide the algorithm with new challenge upon start of every new instance of experiment.

*Figure **Error! Use the Home tab to apply 0 to the text that you want to appear here.** Changes in Gas Plume Model in time comparison*

Figure 3.2-2 shows the mechanism of the gas plume modelling in CoppeliaSim, where the top subfigure shows the plume at the timestamp of 08:55 seconds, while the bottom subfigure shows the plume at 09:28 seconds. Within a second, it can be observed that the plume is generated unorderly, where the particles in the cone-shaped plume change accordingly with time. In this experiment, the wind factors, such as the wind direction and speed, are consistent.

The green circle shows the closest distance to the source of leakage, the orange circle indicates a further distance from the source of leakage, and the red circle indicates the general area of the plume cone expansion. A change in shape is observed in the centre of the plume within the difference of a second. Therefore, this model represents a dynamic and realistic optimisation problem for optimisation by the swarm algorithm in a simulation.

3.3. OBJECTIVE FUNCTION VALUE (OFV) SETUP

According to the technical specification sheet of DJI Phantom 4, the GPS horizontal hovering accuracy may vary by ± 1.5 metres. As such, the experiment will be evaluated in terms of time taken and success count. The objective function value is separated into three quartiles from the original horizontal GPS error range, namely Q3, 0.375 metres; Q2, 0.75 metres; and Q1, 1.125 metres. Q1 is the closest to the actual source of leakage, while Q3 is the furthest away from the source of leakage.

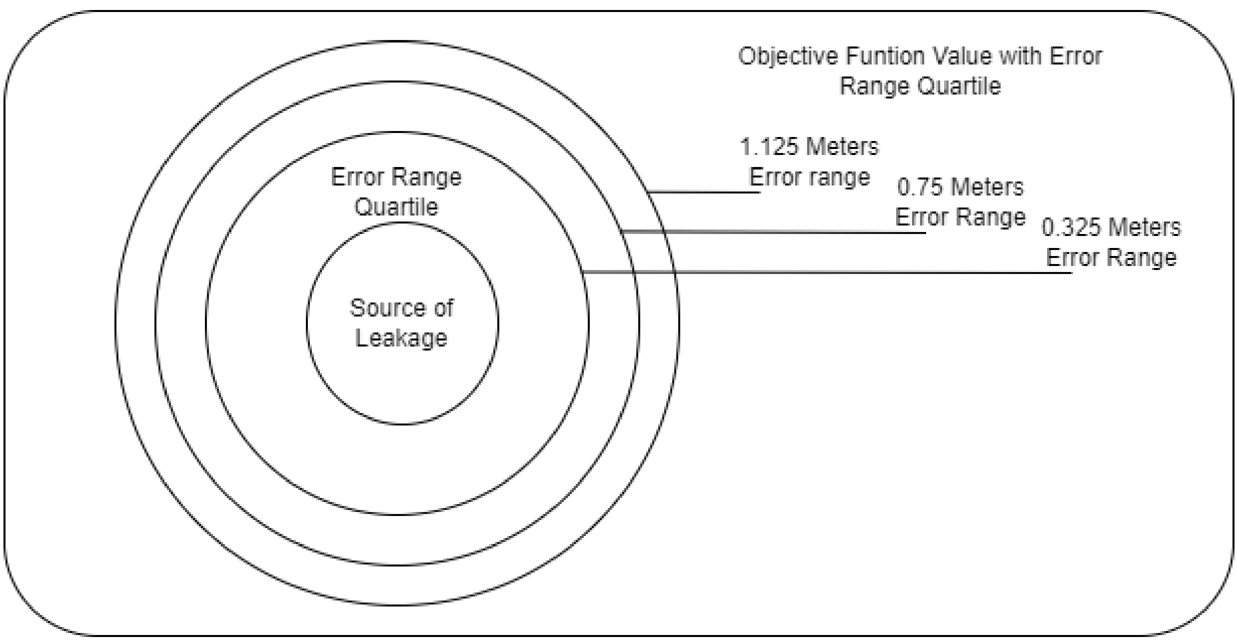

Figure 3.3-1 DJI Drone GPS Error Range Quartile

GiPSO and PSO will be tested with these error range quartiles as the cut-off value; if the location of the drones is on the cut-off value or lower than the cutoff value, the leakage source localisation will be considered a success. If the drone swarm remains out of the OFV and exceeds the time limit, the localisation of the leakage source will be recognised as a failure. The segregation of the OFV into three different sectors provides a quartile analysis with different levels of precision that can be accepted as satisfactory results by both PSO and GiPSO. With each quartile reducing the acceptance proximity by 25%, the OFV is rendered to be more precise by 25% for each quartile, but the difficulty for the results to be satisfactorily declared as optimal global results also increases.

3.4. EXPERIMENT SETUP

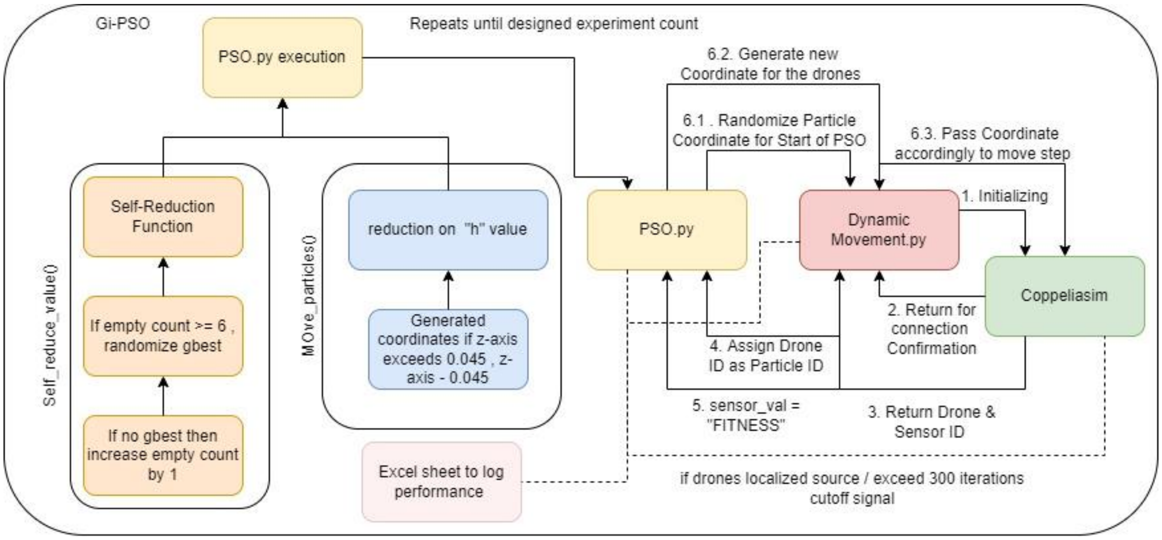

Figure 3.4-1 GiPSO Setup

Python is selected as the foundation of PSO development as Python are stable and flexible as it provides libraries that are essential for the execution of the PSO. CoppeliaSim comes with support compatibility for Python; hence, Python code can be used to communicate with CoppeliaSim for issuing commands or extracting values across the platform. For ease of debugging, a command file, Dynamic_Movement.py, is used to communicate between the listening server of CoppeliaSim and PSO.py. The fitness value and the particles of PSO are integrated with the sensors' detection value and the drones' identification. Each drone acts as a particle of the PSO population, while each drone's sensor will dictate the drone swarm's fitness. The dynamic_movement.py will determine the cutoff condition.

In the simulated environment of CoppeliaSim, optimisation problems are developed based on the Gaussian gas plume model. With this model, GiPSO will generate a new set of movement velocities before the actual movement in the simulation for each movement iteration. The design of the gas plume model developed in CoppeliaSim utilises point cloud objects combined with multiple segmented Box-Mueller transformations with increasing radius. This design allows the plume to be emitted from the point cloud to expand accordingly with increasingly larger conical shapes similar to a realistic gas plume. Major contribution factors such as gas density; wind direction in the x, y, and z axis; and wind speed are included within the point cloud scripting format. This methodology addresses the research questions and objectives, particularly the first research objective, where the research questions and objectives focus on optimisation problems with realistic gas plume dispersion. With the implementation of the modified point cloud, the algorithm can then attempt to optimise the gas plume, which has similar characteristics to real-world plume dispersion characteristics, as the plume will disperse according to the Box-Mueller transformed x and y coefficients for maximum and minimum dispersion limitation.

Both the z-axis coefficient clamp and global best threshold mechanism are implemented within the GiPSO core functions. As depicted in Figure 3-6, the dynamic movement core framework will then be called PSO.py for the generation of new movement velocity based on the sensory detection value as the fitness value in the algorithm.

In conjunction with the research problems, research objectives, and third research question concerning population control, Dynamic Movement.py will require the operators to manoeuvre the core framework and modify the population number. By changing the population number, such data will be forwarded to PSO.py as the swarm size, where this parameter will be used to generate velocity for each particle accordingly. The handler for both the drones and sensor resides within Dynamic_movement.py; this information is then processed and parsed into PSO.py for further processing before the newly generated movement velocities are returned into the simulation in CoppeliaSim.

4. RESULTS AND ANALYSIS

The results are categorised into three different categories, where fewer iterations are much more favourable. Experiments that require less than 150 iterations to localise the source of leakage will be recognised as successful localisation. Experiment instances that require more than 150 iterations will be considered successful with a high number of iterations if the source of leakage is successfully located. Still, a much longer time is needed to locate the source of leakage for such experiment instances. On the other hand, any experiment with more than 300 iterations is considered failed, as the localisation of the source of leakage has taken too long.

$$Iterations, i \begin{cases} Results = success, & if\ i < 150\ iteration \\ Results = high\ iteration, & if\ 150 > i < 300 \\ Results = failed, & if\ i > 300 \end{cases}$$

Eq 3.4-1 Result Category Logic

The categorisation is determined as such because an experiment with less than 150 iterations will take a much shorter time, whereas an experiment that exceeds 150 iterations will take a longer time to localise the source; meanwhile, an experiment that exceeds 300 iterations faces the possibility of exceeding real drones’ battery life span before optimisation is completed.

To evaluate the feasibility of GiPSO and PSO to be implemented into a drone platform, the time restriction limitation is also considered in the experiment to understand further the time required for each successful localisation of the leakage source. Only localisation of the source of leakage within the time frame of 1 minute to 35 minutes is considered a successful localisation.

4.1. TIME TAKEN COMPARISON GiPSO VS PSO

For the initial comparison of the performances of PSO and GiPSO, the time taken for each algorithm was recorded to determine if they were within the time limitation. The cutoff condition that indicates a successful localisation of the source of leakage will be divided into three categories derived from the GPS error range of DJI Phantom 4. The horizontal GPS error range for DJI Phantom 4 is within the range of ± 1.5 metres; thus, the error range can be divided into three quartiles, namely 1.125 metres, 0.75 metres, and 0.375 metres. The purpose of this objective function value (OFV) is to divide the problem into multiple segmentations that the algorithm can analyse to ensure that only highly precise results close to the actual source of leakage will be declared successful.

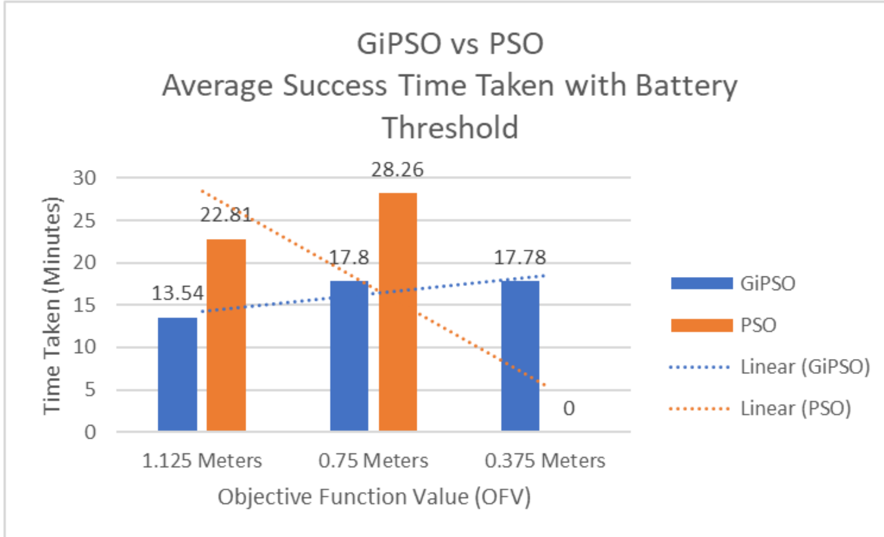

Figure 4.1-1 Average Success Time Taken Comparison

The analysis results for the average time taken for successful localisation of the leakage source with the battery threshold are shown in Figure 4.1-1 for the two different algorithms, GiPSO and PSO. In this analysis, the results are presented for Quartile 1 (Q1), Quartile 2 (Q2), and Quartile 3(Q3), which are the horizontal error ranges away from the actual source of leakage. The optimum process for the algorithm to declare that the source of leakage localisation is successful requires two criteria to be fulfilled: the result must be the highest concentration of gas reading from the sensor and within the declared range of Q1, Q2, and Q3. In figure 4.1-1 we can observe that the performances of GiPSO grows consistent in average time taken for each success instance of localizing the source of leakage.

However, when the comparison of GiPSO and PSO are being observed , as the OFV increases in precision for satisfactory acceptance requirement, PSO has begun to lose the ability to localize the source of leakage, while GiPSO maintaining the average time taken as compared to OFV at 0.75 meters away from the source of leakage.

Comparatively , we can observe that the time taken for OFV 1.125 Meters, 0.75 Meters and 0.375 Meters in PSO stands from 22 minutes up to 28 minutes. On the other hand, GiPSO performances in these OFV stands from averagely of 13 minutes up to 17 minutes. This shows that GiPSO has further improve the time taken for success in localization of source of leakage by 38% lower as compared to PSO performances.

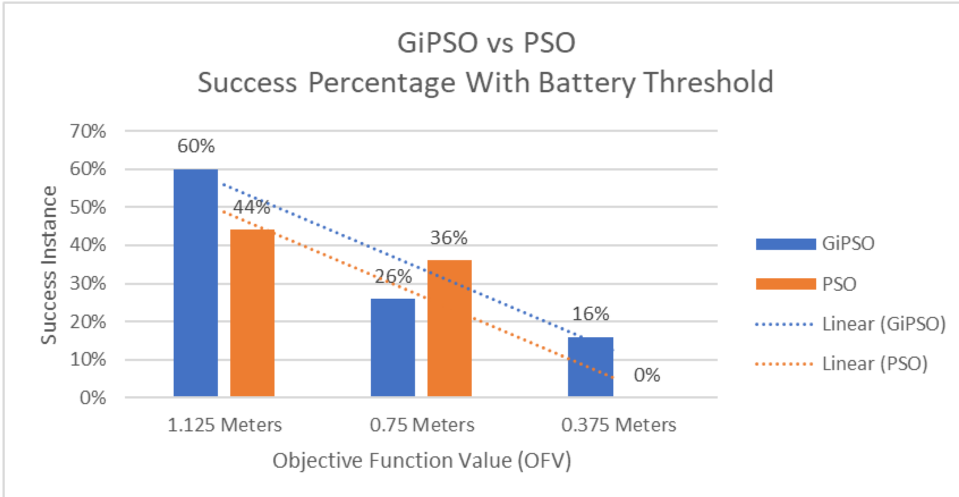

Figure 4.1-2 Success Percentage Comparison

Considering the battery flight time as a cut-off limitation for the localisation, both algorithms can be further studied in terms of their performance with parameters from the DJI Phantom 4 specifications. As depicted in Figure 4.1-2, the comparison of GiPSO and PSO's success percentage and the average time taken is inclusive of the battery threshold limitation. In Figure 4-2, GiPSO's success percentage exceeded that of PSO by 16% with a 1.125 metres error range. However, PSO outperformed GiPSO when the error range was reduced to 0.75 metres. While PSO was more successful, the average time taken for each successful localisation was 28 minutes, while GiPSO took only 17 minutes on average for each successful localisation. This result is due to GiPSO's consistent control in the z-axis velocity coefficient clamp.

On each z-axis velocity generation, GiPSO would consistently reduce the z-axis until the z-axis was above the physical stack height. However, PSO's generation of z-axis velocity was randomised, similar to the x-axis and y-axis velocity generation. This characteristic leads to PSO providing fewer consistent results with more chaotic z-axis velocity in each velocity generation, thus requiring more time to achieve each instance of successful localisation. Despite GiPSO having a lower success rate in localising the source of leakage, the average time taken to achieve optimum localisation was much lower than PSO.

The result for Quartile 3 (0.375 metres away from the actual source of leakage) shows that GiPSO could localise the source of leakage while PSO could not. Despite the instances of success for GiPSO being only at 16% of the overall experiment, the average time taken per success remained the same as Quartile 2 (0.75 metres). With higher difficulty than that of 0.75 metres, the success percentage was reduced by 10%; however, the average time taken was still the same. On the other hand, PSO failed to localise the source of leakage.

4.1.1. GiPSO VS PSO Q1(1.125 Meters) OFV Analysis

The first analysis in the experiment is for the radius of 1.125 metres away from the actual source of leakage. This radius is considered the objective value, as the horizontal error range for the GPS position is ± 1.5 metres in DJI Phantom 4 Pro. As such, the furthest quartile was analysed to determine how GiPSO and PSO performed with a larger error radius during optimisation.

However, three criteria must be met for an experiment instance to be declared a success. One criterion is that the global optimum must be within a radius of 1.125 metres from the actual source of leakage. Another criterion is that the time taken must be within the time limitations of 35 minutes, which represents the battery life span. The final criterion for the declaration of success is that the highest sensor detection value must be within the cutoff radius for successful localisation. The time taken for each successful instance was automatically logged by the algorithm into a CSV file for further data visualisation of the algorithms' performance.

GiPSO		
1.125 Meters OFV Within Threshold		
8.04	12	96.48
13.04	8	104.32
18.04	6	108.24
23.04	3	69.12
28.04	1	28.04
Average Per Success		13.54
		13M 30S

Table 4.1.1-1 GiPSO Result with Q3 (1.125 Meters) OFV

PSO		
1.125 Meters OFV Within Threshold		
22.81	22	501.82
Average Per Success		22.81
		22M 48S

Table 4.1.1-2 PSO with Q1 (1.125 Meters) OFV

In Table 4.1.1-1, it is noticed that PSO had 22 instances of experiment which succeeded in achieving a satisfactory value for a new experiment instance to be started. Meanwhile, from 8 to 28 minutes, GiPSO achieved 30 instances of success out of 50 experiment instances. In terms of performance, GiPSO's performance improved by 32.68% within 30 minutes as compared to PSO's performance.

In terms of successful performances, GiPSO had 12 instances of success out of 30 experiment instances within an average time taken of 8 minutes. The success achieved by GiPSO shows that the distribution for successful experiment instances was concentrated around instances with a lower time taken to achieve success when compared to PSO. PSO achieved success only within an average time of 27 minutes and 48 seconds. This result shows that the capability of PSO to localise the source of leakage successfully within the battery limit is possible; however, longer times are required to achieve these successes. In terms of overall success in localising the source of leakage, GiPSO achieved a success percentage of up to 60% out of 50 experiment instances; on the other hand, PSO only achieved a success percentage of 44% in the experiment.

Figure 4.1.1 – 1 GiPSO vs PSO results Q1 (1.125 Meters) OFV

In figure 4.1.1 – 1 we can observe the performance of GiPSO has achieve 60% of success rate of experiment while 40% of the experiment instances has failed. On the other hand PSO has achieved success rate of 37% while having the failure rate of 63% amongst 50 instances of experiments. The observation in this visualization shows that the improvement of GiPSO with the OFV of Q1 with error acceptance range of 1.125 meters radius from the circle has improved by 23% as compared to PSO. This shows that with the modification done within GiPSO has successfully improve the capability to increase the rate of success while lowering the rate of failure in localizing the source of methane leakage within the battery life of the simulated drone environment.

4.1.2. GiPSO VS PSO Q2 (0.75 Meters) OFV Analysis

In the following analysis, the requirement to be considered a successful experiment instance is further narrowed down to a radius of 0.75 metres from the actual source of leakage. The Quartile 2 analysis analyses how GiPSO can take on the challenge of a much smaller global optimum acceptance radius. In the Q2 experiment setup, the difficulty of achieving a successful instance is increased by 25% because the acceptable success condition is much more precise than the Q1 radius. Comparing the success achieved by PSO and GiPSO in this quartile provides insights into how each algorithm will perform with a radius from the actual source of leakage with higher precision.

GiPSO		
0.75 Meters OFV Within Threshold		
Median of Time	Instances	Median x Instance
8.18	2	16.36
13.18	2	26.36
18.18	6	109.08
23.18	1	23.18
28.18	2	56.36
Average Per Success		17.80
		17M 48S

Table 4.1.2-1 GiPSO Q2 (0.75 Meters) OFV

PSO		
0.75 Meters OFV Within Threshold		
Median of Time	Instances	Median x Instance
28.26	18	508.68
Average Per Success		28.26
		28M 15S

Table 4.1.2-2 PSO Q2 (0.75 Meters) OFV

The results of the optimisation analysis for the Q2 horizontal error range of DJI Phantom 4 are shown in Table 4.1.2-2. With an objective function value and radius of 0.75 metres around the actual source of leakage, the optimisation condition is slightly more challenging compared to that of 1.125 metres away from the actual source of leakage. In this quartile, GiPSO achieved 13 successful optimisations in 50 experiment instances, while PSO achieved 18 successful instances out of 50 experiment instances. However, PSO achieved 18 successful instances within the average time taken of 28 minutes out of 30 minutes, while GiPSO achieved 6 instances of success within the average time taken of 18 minutes. In terms of the average time taken per instance of the experiment, GiPSO required 17 minutes; however, for PSO to achieve a single successful instance, 28 minutes were required. The performance of GiPSO improved the success rate of the swarm localisation of the source of leakage by up to 38.65% when compared to PSO.

To conclude, in the Quartile 2 OFV analysis of the performance of PSO and GiPSO, the overall success of GiPSO was observed to have fallen to 13 successful instances; meanwhile, PSO achieved 18 successful experiment instances. Although PSO had more successful instances in localising the source of leakage, the average time taken for each successful experiment instance required up to 28 minutes and 15 seconds. On the other hand, despite having lesser successful experiment instances, GiPSO required only 17 minutes and 48 seconds for each successful instance. This distinction highlights the difference in performances between the two algorithms.

Figure 4.1.2 -1 GiPSO vs PSO results Q2 (0.75 Meters) OFV

In figure 4.1.2-1 the performance of GiPSO has reduced Success rate when the OFV further narrows down with higher precision requirement. As compared to Q1 (1.125 Meters) with 60% success rate, GiPSO has achieved 26% success rate with Q2 (0.75 Meters) OFV. PSO has achieve higher success rate as compared to GiPSO at 32% success rate while having a slightly lower rate of failure rate at 68% as compared to GiPSO at 74%. Despite PSO do have slightly higher chances of successfully localize the source of leakage, in table 4.1.2-1 and 4.1.2-2 we can observe that the time taken for GiPSO to achieve each success averages out at 17 Minutes and 48 seconds, while PSO requires longer time with 28 minutes and 15 seconds. This shows that PSO are able to achieve slightly more success rate in localizing the source of leakage with the cost of reaching closely to the maximum flight time available for the drone battery limitations.

4.1.3. GiPSO VS PSO Q3 (0.375 Meters) OFV Analysis

GiPSO		
0.375 Meters OFV Within Threshold		
Median of Time	Instances	Median x Instance
10.9	3	32.7
15.9	2	31.8
20.9	0	0
25.9	3	77.7
Average Per Success		17.78
		17M 48S

Table 4.1.3-1 GiPSO Q3 (0.375 Meters) OFV

In the Q3 objective function value, the error range was reduced to a radius of 0.375 metres away from the actual source of leakage. PSO results are not available for comparison in Quarter 3 as since PSO couldn’t optimize the source of leakage. In comparison to the Q2 and Q1 error ranges, 0.375 metres represent a criterion that is harder to be satisfied and a higher precision due to being closer to the source of leakage.

The advantage of having such close proximity is the further enhancement of the precision that can be achieved by GiPSO and PSO in the optimisation with controlled time allocation. With the limitation of 30 minutes set as the condition of failure due to battery limitation, PSO was observed to fail to achieve any success within 30 minutes, which implies that PSO was unable to achieve success within the battery life threshold. On the other hand, GiPSO achieved 8 instances of success in the experiment.

Figure 4.1.3 – 1 GiPSO vs PSO results Q3(0.325 Meters) OFV

In figure 4.1.3-1 , we can observe where both of the algorithms are being tested with a more precise OFV error range. With high accuracy acceptable OFV, we can observe that GiPSO managed to achieve success rate of 16% while PSO fails to localize the source of leakage. GiPSO does not only retain the capability of localizing the source of leakage, but the average time also required for GiPSO to successfully localize the source of leakage averages out at 17 minutes and 48 seconds.

4.1.4 OVERALL GiPSO PERFORMANCE IMPROVEMENT IN TIME TAKEN

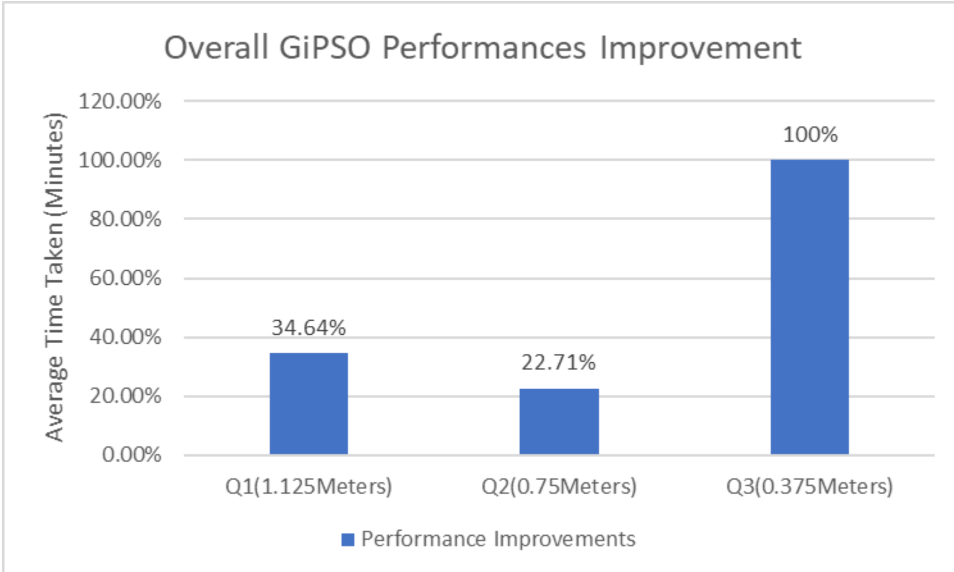

Figure 4.1.4-1 GiPSO Overall Performance Improvement

In Figure 4.1.4-1, we can observe that the performance of GIPSO compared to PSO in Q1 results, GiPSO, has improved by 34.64% of the overall performance. As the accuracy increases, GiPSO has a lower chance of localizing the source of the gas plume. However, regarding the time taken, GiPSO has improved performance by 22.71%. When the acceptance range has further reduced to 0.375Meters error range from the source of leakage, we can observe that GiPSO exhibits the capability to localize. However, PSO has failed to carry out optimization.

From figure 4.1 - 3, we can observe the performances of GiPSO as compared to PSO. GiPSO has improved the ability to localize the source of leakage regarding horizontal error range as OFV. The performances of GiPSO show improvements in both capabilities to increase the success of optimizing the source of gas plume dispersion. It has also reduced the time required to localize the source of leakage compared to PSO. As the OFV reduces and the satisfactory criteria narrow down to higher precision, GiPSO retains the capability of optimizing the source of leakage. In contrast, PSO has failed to localize the source of leakage entirely.

These improvements in the time taken and the potential to optimize the source of leakage shows that the modification on GiPSO has further improved the performances compared to PSO. In the following sections, our research will then utilize GiPSO with altered swarm populations to further evaluate the performance on population impact. This study will control the swarm population from 4 to 20 drones. Each of the experiment sets will increase the population by two drones to provide insights into a closer gap in population growth and how each growth of two drones can impact the capability of GiPSO to optimize the source of leakage.

4.1.5 STATISTICAL ANALYSIS OF GiPSO vs PSO

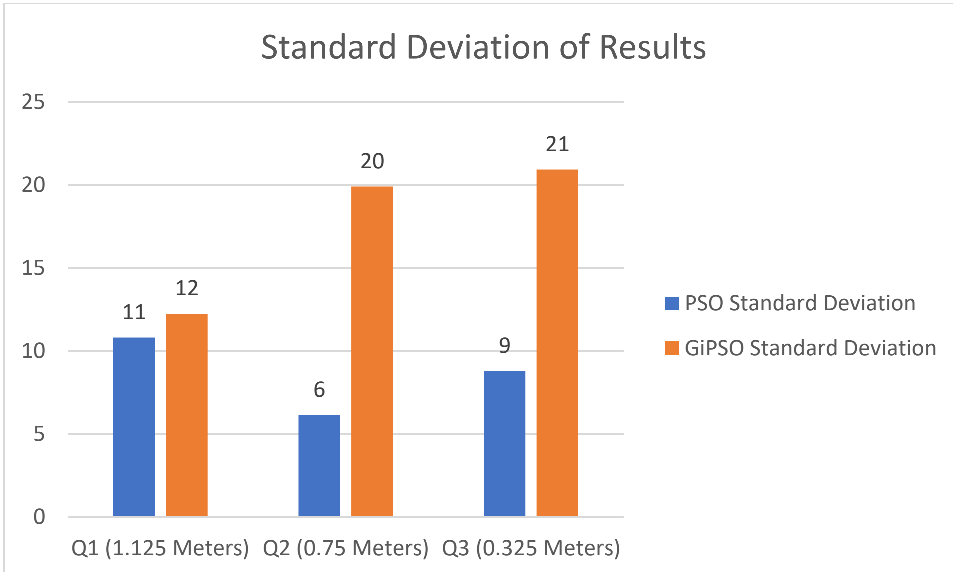

Figure 4.1.5-1 Standard Deviation of GiPSO Results vs PSO Results

In figure 4.1.5 – 1 we can observe the analysis of GiPSO results alongside with PSO results. In the figure, PSO will be represented with blue bar, however GiPSO will be represented with orange bar, each of the data representation will be clustered according to the results gathering based on the OFV quarters. In the analysis we can observe that the standard deviation of Q1 for PSO values are high as the time variations ranges from 20 minutes to 36 minutes , although GiPO has a higher value for standard deviation at 12, the variation of time taken ranges from 5 minutes to 25 minutes. Although the deviation of GiPSO are larger as compared to PSO the time ranges variation are lower.

Furthermore, in Q2 we can observe the standard deviation of PSO holds the value of 6 however, GiPSO indicates a larger value for standard deviation, This shows that the deviation of time taken range are larger as compared to PSO, this is because the time variation taken to successfully localize the source of leakage for GiPSO are spreaded out from 5 minutes to 30 minutes span. However, PSO standard deviation values are lower as the time taken are closely group around 25-30 minutes.

This shows that although the original PSO has higher consistency in time taken to localize the source of leakage, the average time required to localize the source of leakage are higher as compared to GiPSO. The similar trend can also be observed for OFV of Q3 as the values of standard deviation of PSO is at 9 , however GiPSO has a wider spread value of 21. This phenomenon are caused by close grouping of PSO grouping of experiment counts with more experiment ranges from 41 minutes to 62 minutes. However, GiPSO time taken ranges from 3 minutes to 35 minutes. Although GiPSO spread of time taken are larger in terms of ranges as compared to PSO, GiPSO satisfies the requirement to lower the time taken for localizing the source of leakage.

4.1.6 MEDIAN RESULTS OF GiPSO vs PSO

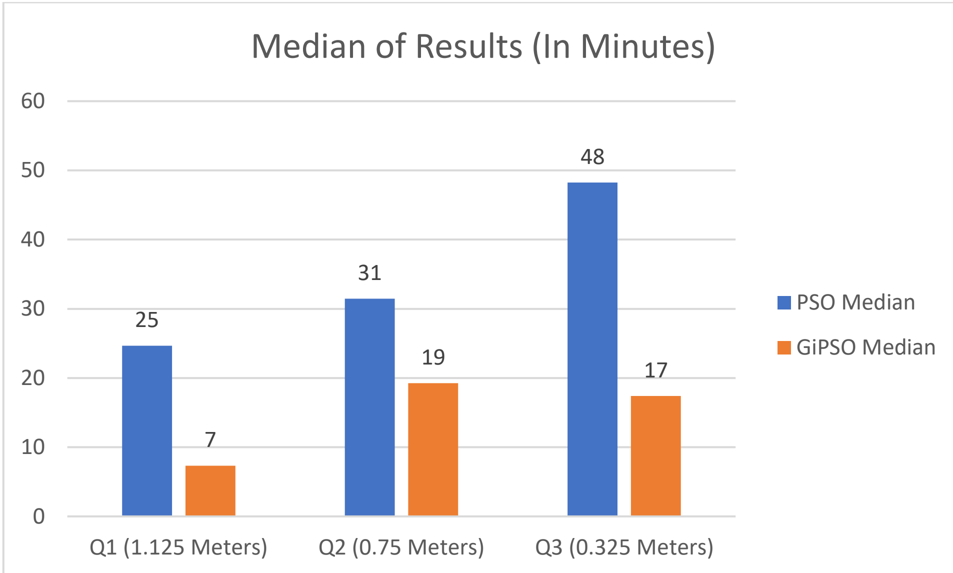

Figure 4.1.6-1 Median Results of PSO vs GiPSO

Upon analyzing the standard deviation of GiPSO versus PSO we have to analyze the median of the time taken for both algorithms to further understand their efficiency. In figure 4.1.6 -1 , the data are visualized in the format of both algorithms data median in terms of minutes grouped by their OFV quarters. In Q1 quarters of OFV we can observe that the median time taken for PSO stated at 25 minutes while the time taken for GiPSO is at 7minutes, this suggests that with larger error range the average time taken for GiPSO to localize the source of leakage has improved by 18 minutes as compared to PSO.

Similar fashion can be observed in both Q2 and Q3 OFV as the error range narrows further thus requiring a high precision to satisfy the acceptance conditions. GiPSO has improved by 12 minutes in Q2 OFV and 31 minutes in Q3 OFV, respectively. This shows that despite with higher precision requirement to satisfy the optimization requirement, the overall median in terms of time taken to localize the source of leakage fo GiPSO has improved as compared to PSO.

To conclude the analysis of median of time taken to localize the source of leakage, GiPSO has successfully reduced Q1 time taken by 28%, Q2 by 61%, and Q3 by 35%. While the modification conducted on GiPSO has successfully decreases the time taken to localize the source of leakage, it has also increased the capability of localizing the source of leakage while the precision increases without having larger increment towards time taken to localize the source of leakage as compared to PSO.

4.2. POPULATION STUDY ON GiPSO

For the population study, the conditions of success are the number of experiment instances to achieve the global optimum should be less than 150 instances, and the cutoff should be achieved. The experiment setup used drone populations of 4, 6, 8, 10, 12, 14, 16, 18, and 20 drones to study the performance of GiPSO. For each population swarm size, 50 experiment instances were run before the results were analysed.

In Figure 4-3, the performance of successful attempts in localising the source of leakage was observed to be a progressive growth curve as the population increased from 4 drones to 8 drones. Increasing the population size from 4 drones to 6 drones led to an increase in the success rate from 73.68% to 78.95%, while a further increase in the population size to 8 drones resulted in a slight decrease in the success rate to 78.94%. However, from the population size of 8 drones to 14 drones, GiPSO's success rate increased along with the growth in the swarm size. From a success rate of 78.94% for an 8-drone population, the success rate rose to 92% for a 14-drone population.

4.2.1. SUCCESS RATE OF GiPSO POPULATION STUDY

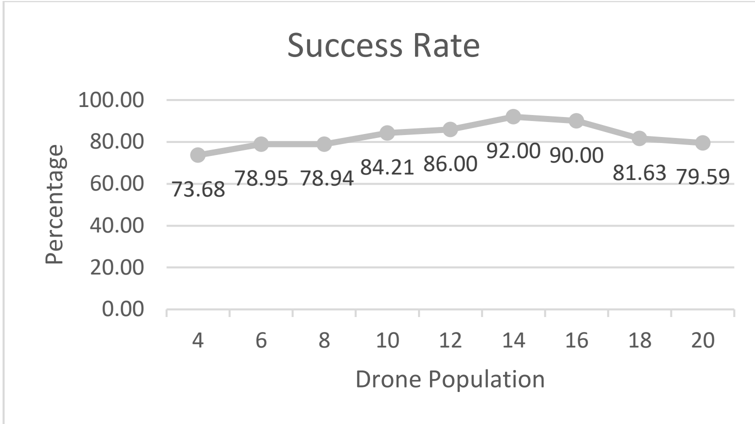

Figure 4.2.1-1 Success Rate of GiPSO population study

However, a performance bottleneck can be observed in the data shown in Figure 4.2.1-1. When the population increased to 16 drones, the success rate of localising the source of leakage dropped to 90% compared to the 92% success rate for 14 drones. A decline in a similar fashion can be observed as the drone population continued to increase from 16 drones to 18 and 20 drones. The swarm success rate decreased when the swarm size increased from 16 drones to 20 drones.

The success rate fell from 90% to as low as 79.59%. This result indicates that as the swarm size increases, the swarm will inevitably change the global best individual on each movement iteration.

This change will further randomise the direction of the swarm, as GiPSO's generation of new movement velocity is based on the location of the current global best individual. As the global best individual changes rapidly, the swarm will be randomised more frequently, thus steering the swarm into a chaotic movement fashion.

The analysis of the success rate and the population growth size shows the capability of GiPSO to improve its performance with an increment in the swarm population. The improvement in GiPSO's performance with increasing population growth shows that as the population size increases, the swarms can optimise a larger scale of the search zone.

With the capability of optimising a larger search zone, the time required to localise the source of leakage successfully will be reduced, and the time criteria to achieve the best global solution will be satisfied. However, as the population continues to grow, there will be a limitation where the population will reach a bottleneck and inevitably fall into redundancy as the population will begin to move chaotically. Hence, the swarm will take longer to achieve the desired results within 35 minutes.

4.2.2. HIGH ITERATIONS SUCCCESS OF GiPSO POPULATION

In Section 4.2.2, the performances of the swarm optimisation with a high number of iterations will be analysed. This analysis will focus on GiPSO with a high number of iterations, where successful localisation of the source of the leakage requires a time above 35 minutes and below 50 minutes. This analysis provides an overview of how often GiPSO can achieve success while requiring a longer time. Through this analysis, the effect of the drone population in achieving successful optimisation will be studied further.

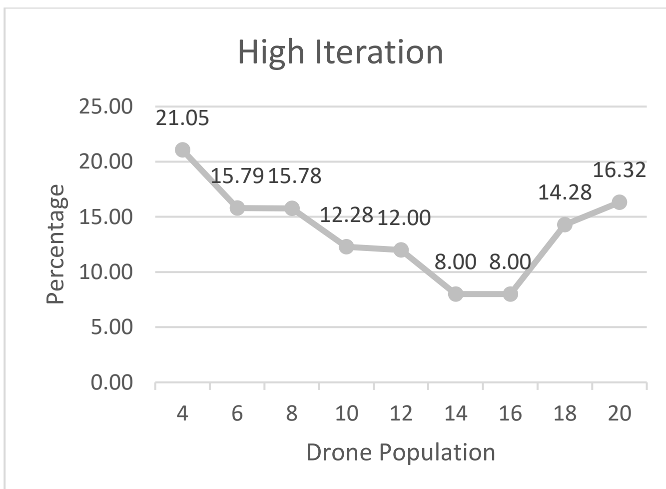

Figure 4.2.2-1 High Iteration Success of GiPSO Population Study

Figure 4.2.2-1 shows that as the population or swarm size increased, GiPSO's tendency to achieve success reduced while requiring a longer time. From the drone population of 4 drones to 14 drones, the success percentage of GiPSO with a high number of iterations reduced from 21.05% to as low as 8%.

This result shows that the possibility of reducing the time taken for each instance of success increased alongside an increment in the population count. However, increasing the drone population from 16 to 20 drones led to an increase in the success percentage of GiPSO with a high number of iterations. With a 16-drone swarm population, the percentage of successful instances was 8%.

With a population of 18 drones, the success percentage of GiPSO with a high number of iterations increased to 14.28%, and for 20 drones, it increased further to 16.32%. This analysis shows that for any population above 16 drones, increasing the swarm size increases the possibility of success while requiring a longer time taken. The phenomenon that causes this condition is the redundancy of population, which leads to the rapid switching of the global best candidate. The side effect of such a scenario is that the swarm will disperse chaotically, which inevitably affects the capability of the swarm to optimise the global best solutions in a shorter time.

4.2.3. THE FAILURE RATE OF THE GiPSO POPULATION STUDY

To further assess the performance of GiPSO with population growth as the control parameter, the failure rate will be analysed in Section 4.1.3. In contrast to the successful attempts, failure rate analysis can provide insights into how GiPSO performs with increasing swarm size. Experiments which exceed the 1-hour time frame and have yet to achieve the conditions to be declared as successful instances will be designated as failed instances.

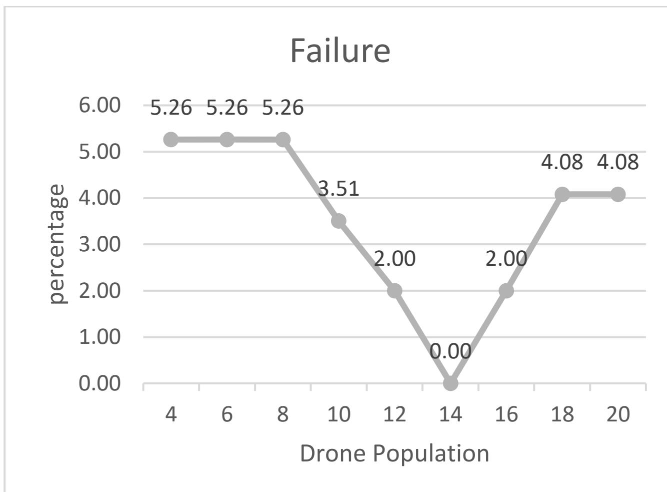

Figure 4.2.3-1 Failure Rate of GiPSO in Population Study

Figure 4.2.3 - 1 shows that from the swarm size of 4 to 8 drones, the failure rate was 5.26% of 50 experiment instances. The improvements of the GiPSO performances in localising the source of leakage can be observed when increasing the drone population from 10 to 14 drones, where the failure rate decreased from 5.26% to 3.51%, 2%, and 0%. This result indicates that the performance of GiPSO had the lowest failure rate with a drone size of 14 drones.

However, as the drone population continued to increase to 16, 18, and 20 drones, a rise in failure rate can be observed. This result indicates that GiPSO achieved peak performance when the drone population was 14 drones. However, upon reaching 16, 18 and 20 drones, GiPSO began to fail to optimise the source of leakage. The rise in failed instances is due to the redundancy created by the increasing number of redundant drones causing the swarm to move chaotically, thus redirecting the swarm away from the actual source.

Drone Population	Success Percentage	High Iteration Percentage	Failure Percentage
4	73.68%	21.05%	5.26%
6	78.95%	15.79%	5.26%
8	78.94%	15.78%	5.26%
10	84.21%	12.28%	3.51%
12	86.00%	12.00%	2.00%
14	92.00%	8.00%	0.00%
16	90.00%	8.00%	2.00%
18	81.63%	14.28%	4.08%
20	79.59%	16.32%	4.08%

Table 4.2.3-1 GiPSO Population Study Result Compilation

From the table 4.2.3-1 shown above, the performance of GiPSO can be further analysed based on different results categorisation. Success percentage signifies success in localising the source of gas leakage within the available drone flight time of 35 minutes. On the other hand, success percentage with a high number of iterations represents success in localising the source of leakage while requiring more than 35 minutes and less than 55 minutes, which would be challenging for commercial drones to achieve.

Failure percentage signifies the failure to localise the source of leakage and satisfy the 55-minute time limitation. The different levels of result segregation are differentiated based on the available flight time of DJI Phantom 4 Pro.

According to the table above, an improvement trend can be observed for the success percentage when the drone population increased from 4 drones to 14 drones. The success percentage increased by 18.32%. While the success percentage increased, the success percentage with a high number of iterations reduced by 13.05%. This result shows that the increment of population led to consistent growth of the optimisation capability. Meanwhile, the failure percentage was reduced by 5.26% when the drone population increased from four drones to 14 drones.

While the optimum performance of GiPSO was achieved with a 14-drone population, a performance bottleneck was noticed for the drone population of 16 to 20 drones. As the population grew beyond 14 drones, the success percentage deteriorated from 92% to 79.59%, representing a significant drop of 12.41% in successful optimisation within the time limit of 35 minutes. Alongside it, the success percentage with a high number of iterations increased from 8% to 16.32%, representing an increment of 8.32% in successful optimisation within 45 minutes.

This phenomenon shows that as the population exceeds the 14 drones required for peak performance, the swarm will be steered into chaotic movements due to the redundancy of the population. As the experiment continues , each of the redundant population will change the global best individual in a chaotic fashion causing the swarm to move in a chaotic pattern thus affecting the swarm's ability to localize the source of leakage efficiently therefore increasing the failure percentage and high iteration percentage.

From the population study of GiPSO, we can observe that as the population grows, the performance exhibits growth in successful localisation while reducing the rate of failure as well as the rate of high iteration success. This shows that GiPSO can perform localisation and achieve global optima within the available flight time of the DJI Phantom 4 battery available flight time of 35 minutes. While we can observe that GiPSO peaks out its' performances at 14 drones' population, the performance faces a performance bottleneck after 14 drones. From a drone population of 16 drones onwards to 20 drones, the performance shows an increment in failure rate along with its' high iteration rates. This indicates that GiPSO has begun to reduce its' performances due to excessive population, which chaotically influences the direction as well as the generation of the velocity for the swarm particles.

To conclude the population study, we can observe that GiPSO performances show positive growth in its' performances as the population grows. The change of performances in successful optimisation with the low time taken extends perpendicular, along with the development of drone population sizes.

As the swarm population grows to 14 drones, the performance begins to bottleneck and eventually decreases its performances in low time taken global optimisation. GiPSO can perform global optimisation with a low population count. GiPSO shows a high success rate in global optimisation within available flight time with DJI Phantom 4 as battery benchmark in the lower particle population count

5. DISCUSSION

The data and results obtained from the experiments suggest that GiPSO has better performance than PSO in successfully localising the source of leakage in a simulated environment. The optimisation problem for both PSO and GiPSO is developed based on the Gaussian plume model, representing a challenging problem with a realistic gas plume behaviour. The result indicates that with larger error ranges such as Q1 OFV (1.125 metres), PSO performs more competitively than GiPSO. However, when the satisfactory proximity criterion is reduced to Q3(0.325 metres) from the actual source of leakage, PSO fails to achieve any success within the provided battery threshold. On the other hand, GiPSO can achieve successful optimisations with such precision.

In line with the research problem of investigating the drone population's impact on the swarm algorithm for movement velocity control in the optimisation of dynamic problems, GiPSO indeed showed improvement in performances alongside the growth of the drone population. However, within this research, it is noted that the peak performance of GiPSO was achieved when the drone population was 14 drones, resulting in a 92% success rate in localising the leakage source within the 35-minute threshold with no proximity error from the actual source of leakage. As the population continued to increase to 16, 18, and 20 drones, it is noticeable that GiPSO's capability to localise the source of leakage within the provided time limit began to deteriorate; simultaneously, the failure rate rose accordingly as the population increased from 14 drones to 20 drones.

These results are relevant when considering dynamical optimisation problems such as gas plume dispersion. The challenge posed by such dynamical problems is that the dispersion characteristics of gas plumes detected by drones may not be fixed at every point in time. Instead, the dispersion of gas plumes may vary due to external factors such as wind direction and wind speed.

The constraint of the methodological choices in this study is that the early search phases are purely based on the randomisation of velocity. Such randomisation in the early search phases may result in the algorithm falling into an extended early search phase, provided no drones have picked up gas traces. To overcome such a limitation, perhaps other methodologies such as Ant Colony Optimisation (ACO) and Artificial Bee Colony (ABC) can provide a strategic methodology to optimise dynamic problems such as gas plume dispersions.

5.1 CONCLUSION

In Conclusion, this research aimed to investigate the adaptation of existing swarm intelligence with Gaussian Gas Plume model modification to optimize the source of gas leakage. The findings revealed using the control in Z-Axis coefficient clamping in movement velocity generations for the particles. Upon achieving the self-best threshold, the re-dispersion of the particles further improves the swarm performances in localizing the source of leakage with the shorter time required to localize the actual source of leakage.

Regarding research questions to investigate the population impact on swarm intelligence in optimizing dynamic problems, GiPSO has shown peak performance at 14 drone populations as failure instance shows 0 instances and most of the success in localizing the source of leakage to be on the low time taken. This has proven that GiPSO can improve performance consistently when the population grows from 4 drones to 14 drones. Re-dispersion of the swarm particles shows the ability stated in the research objective to employ dynamical control for swarm intelligence in controlling the movement velocity of the drone swarm. Re-dispersion of the swarm particles in GiPSO allows the control of the swarm preventing the swarm from falling into local optima, which often exhibits in the original PSO.

The study contributes to the modification and adaption of PSO field research in oil and gas optimization via drone-based detection platform. The findings of this research suggest that swarm intelligence has the potential to improve and enhance the existing drone detection platform in the localization of gas leakage. As such, this study highlights the improvements and significance of introducing swarm intelligence into gas detection platforms and underscores the need for continued research in the oil and gas area.

5.2 FUTURE WORK SUGGESTIONS

Th modification performed on GiPSO exhibits improvements on original PSO platform with integration of Drone based gas leakage detection platform. The capability to achieve higher success in localizing the source of leakage with lower swarm population size and shorter time required to achieve localization of source of methane leakage. With exhibition of improvement of GiPSO in localizing the source of in randomized scenario such as wind direction, wind speed which would affect the optimizing problem.

As such, future experiment suggestions such as localization of undersea natural gas leakage with swarm intelligence enabled underwater drones would further challenges the capability of modification such as GiPSO. This is because when natural gas leaks under the seas, the direction of the plume are different with the terrestrial as it does not affect by wind factor, but instead by the water pressure , water depths as well as water flow direction.

Additionally , this suggestion would also be challenged by moon phase as it would affect the direction of the sea current therefore additional affects the sizing of the optimization problem. This research can provide insights on how to simulate realistic plume behaviour with Gaussian Gas Plume Model as well as methods to encourage the detection platform to prevent PSO inspired methods to fall into local optima thus failing to localize the source of leakage.

Researchers can benefit from this research work to understand as well as providing ideas to further improve PSO with Gas Related Optimization methods via simulations or realistic world optimization with actual quad drone detection platform.

REFFERENCES

[1] B. A. .. G. joao, J. P. C. R. Joel, A. L. .. R. Ricardo, K. Neeraj and K. Sergey, “IoT-Enabled Gas Sensors: Technologies, Applications, and Opportunities,” *Journal of Sensor and Actuator Networks,* vol. 8, no. 57, pp. 1-29, 2019.

[2] A. Tahani, T. Manh-Kien, V. Reid and P. Michael, “A Review of Methane Gas Detection Sensors: Recent Developments and Future Prespective,” *Invention,* vol. 5, no. 28, pp. 1-18, 2020.

[3] D. Ming, Z. Chuantao, M. Shuzhao, Z. Yu, D. Qialing, W. Yiding and K. T. Frank, “Development and Measurements of a Mid-Infrared Multi-Gas Sensor System for CO, CO2 and CH4 Detection,” *Sensors,* no. 17, 2017.

[4] Z. Yue, D. Peipei, Z. Chuantao, Y. Weilin and W. Yiding, “A Handheld mid-infrared Methane Sensor using a Dual-step Differential Method for Additive/Multiplicative Noise Supression,” *Infrared Physics & Technology,* vol. 1, no. 79, pp. 151-159, 2016.

[5] J. B. Mate, L. L. Shao-Xiong, H. K. Kang and M. S. Timothy, “A Chemiresistive Methane Sensor,” *PNAS,* vol. 118, no. 2, pp. 1-6, 2021.

[6] Z. Hang, D. JinHui, C. Can and L. Shuhai, “Stress and Strain Analysis of Sperical Sealing Cuups of Fluid-Driven Pipeline Robot in Dented Oil and Gas Pipeline,” *Engineering Failure Analysis,* no. 108, pp. 1-14, 2020.

[7] S. Heru and H. Ahmad Nur, “Designing A Wheeled Robot Model For Flammable Gas Leakage Tracking,” *Second International Conference on Informatics and Computing,* vol. 11, pp. 1-6, 2017.

[8] Z. G. Mohd, M. N. Noorhayati, R. Noraziah and P. Sulastri, “Development of Microcontroller Based Mobile Gas Sensing Robot,” *Procedia Engineering,* vol. 41, pp. 1190-1196, 2012.

[9] A. A.-R. Adel, “On The Atmospheric Dispersion adn Gaussian Plume Model,” in *2nd International Conference on Waste Management , Water Pollution, Air Pollution, Indoor Climate*, Corfu, 2008.

[10] H. Z. Harold, “Electric Fields of Flowers Stimulate the Sensory Hairs of Bumble Bees,” in *Topological Defects in Liquid Crystals*, Texas, PNAS (Proceedings of the National Academy of Sciences of the United States of America), 2016, pp. 7020-7021.

[11] J. G. Juan, G. G. Eduardo and B. Antonio, “A Survey on Robotic Technologies for Forest Firefighting: Applying Drone Swarms to Improve Firefighters' Efficiency and Safety,” *Applied Sciences,* vol. 11, no. 363, pp. 1-18, 2021.

[12] S. Amit and Hamad Karki, “Application of Robotics in onshore oil and gas Industry - A review Part 1,” *Robotics and Autonomous Systems,* vol. 75, pp. 490-507, 2015.

[13] K. James and E. Russell, “Particle Swarm Optimization,” in *IEEE*, Washington DC, 1995.

[14] K. L. S., S. K., S. R. S., A. Enas, A. N., R. Gustavo and U. J., “An effect of big data technology with ant colony optimization based routing in vehicular ad hoc networks: Towards smart cities,” *Jorunal of Cleaner Production,* vol. 217, pp. 582 - 593, 2019.

[15] M. K. Priyan, D. G. Usha, S. Revathi, C. Naveen and V. Ramachandran, “Ant colony optimization algorithm with Internet of Vehicles for intelligent traffic control system,” *Computer Networks,* vol. 144, pp. 154-162, 2018.

[16] L. H. S. and N. B., “Novel low-complexity DS-CDMA multiuser detector based on ant colony optimizatio,” *IEEE 60th Vehicular Technology Conference,* vol. 3, pp. 1939 - 1943, 2004.

[17] C. H., w. Z., K. R. and Y. C., “Global path planning for explosion-proof robot based on improved ant colony optimization,” *2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS),* pp. 36-40, 2016.

[18] N. Anand and S. Rajeshwar, “IEEMARP- a novel energy efficient multipath routing protocol based on ant Colony optimization (ACO) for dynamic sensor networks,” *Multimed Tools Appl ,* vol. 79, p. 35221–35252, 2020.

[19] S. K. Mustafa and F. Oguz, “A directed artificial bee colony algorithm,” *Applied Soft Computing ,* vol. 26, pp. 454 - 462, 2015.

[20] D. Aveek, T. Bipan, B. Rajib and B. Nabarun, “Black tea quality evaluation using electronic nose: An Artificial Bee Colony approach,” *2011 IEEE Recent Advances in Intelligent Computational Systems,* vol. 2011, pp. 143-146, 2011.

[21] K. M. Santosh, T. C. Felix and K. T. M., “Leak detection of pipeline: An integrated approach of rough set theory and artificial bee colony trained SVM,” *Expert Systems with Applications,* vol. 39, no. 3, pp. 3071 - 3080, 2012.

[22] Z. Yinghao, Y. Quansheng, Y. Zheng, Y. Xiaolin and J. Buyu, “A Novel Artificial Bee Colony Algorithm for Structural Damage Detection,” *Advances in Civil Engineering,* vol. 2020, pp. 1-22, 2020.

[23] M. Tao, L. Shuhai and X. Huaping, “Location of natural gas leakage sources on offshore platform by a multi-robot system using particle swarm optimization algorithm,” *Journal of Natural Gas Sciences and Engineering,*

vol. 84, pp. 1 - 15, 2020.

[24] A. B., R. S., S. T., E. Geoffrey, S. P. and R. A., “Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning,” *SN Applied Sciences,* vol. 2, 2020.

[25] S. Martin, L. Jan and P. Linor , “Plume Tracking by a Self-stabilized Group of Micro Aerial Vehicles,” in *Modelling and Simulation for Autonomous System*, Springer, 2014, pp. 44 - 55.

[26] S. P. Husanbir, S. Dilbag and K. M. Avleen, “Improved Particle Swarm Optimization Based Adaptive Neuro-Fuzzy Inference System for Benzene Detection,” *CLEAN -Soil, Air, Water,* vol. 46, no. 5, 2018.

[27] L. Jia Jun, Y. Kian Meng, E. Kok Seng and N. Qiang, “Open Agricultural Burning Detection with Natural Inspired Swarm-based Detection Platform,” in *International Conference on Artificial Intelligence in Engineering and Technology*, Sabah, 2022.

[28] H. Yu , L. Lei and J. Renxing, “A Hybrid Algorithm for Gas Source Locating Based on Unmanned Vehicles in Dynamic Gas Environment,” *Mathematical Problems in Engineering,* vol. 2021, pp. 1 - 8, 2021.

[29] L. H. Nyayu, S. H. Ade, Y. Falah, N. Siti and Y. Irsyadi, “SWARM ROBOT IMPLEMENTATION IN GAS SEARCHING USING PARTICLE SWARM OPTIMIZATION ALGORITHM,” *Computer Engineering and Applications,* vol. 6, no. 3, pp. 127 - 138, 2017.

[30] S. Jin, D. Jian, S. Weifeng and W. Yuanzhi, “Hybrid Odor Detection System for Search and Rescue Robot Based on PSO,” *CHEMICAL ENGINEERING TRANSACTIONS,* vol. 68, pp. 151 - 156, 2018.

[31] Z. Cancan, Z. Bo, Q. Shen and M. Chao, “A stochastic programming approach for the optimization of gas detector placement in offshore platforms,” *Ocean Engineering,* vol. 187, pp. 1 - 8 , 2019.

[32] H. R. Mohammad, H. Ali and M. V. S. Hossein, “Inverse Frequency Response Analysis for Pipelines Leak Detection Using the Particle Swarm Optimization,” *Internation Journal of Optimzation in Civil Engineering,* vol. 6, no. 1, pp. 1 - 13, 2016.

[33] Z. MengFei, Y. YanHui, X. YinZe, H. YingChao, C. YiJun, L. JunJie and P. HaiTan, “A Pipeline Leak Detection and Localization Approach Based on Ensemble TL1DCNN,” *IEEE Access ,* vol. 9, pp. 47565 - 47578, 2021.

[34] Y. Yuting, Z. Ru, W. Ji and L. Juming, “Modified PSO algorithms with “Request and Reset” for leak source localization using multiple robots,” *Neurocomputing,* vol. 292, pp. 82 - 90, 2018.

[35] Y. Dandi, H. Nan, L. Jingyi and J. Daan, “Novel Leakage detection by Ensemble 1DCNN-VAPSO-SVM in oil and gas pipeline systems.,” *Applied Soft Computing,* vol. 115, pp. 1 - 16, 2022.

[36] G. JianFeng, Z. Yu, N. Kai, Z. HuaiZhi, H. Bin and Y. JiaLin, “Reseach on Oil-gas Pipeline Leakage Detection Method Based on Particle Swarm Optimization Algorithm Optimized Support Vector Machine,” in *Journal of Physics : Conferece Series*, 2021.

[37] M. Tao, L. Shuhai and X. Huaping, “Multirobot searching method of natural gas leakage sources on offshore platgorm using ant colony Optimization,” *International Journal of Advanced Robotic Systems,* pp. 1 - 16, 2020.

[38] M. Qing-Hao, Y. Wei-Xing, W. Yang, L. Fei and Z. Ming, “Adapting an Ant Colony Metaphor for Multi-Robot Chemical Plume Tracing,” *Sensors,* vol. 12, pp. 4737-4763, 2012.

[39] Z. Yuhua and L. Dehan, “A Modified Ant Colony Algorithm Used for Multi-robot,” *ICIC,* pp. 502 - 509, 2008.

[40] Z. Yinghao, Y. Quansheng, Y. Zheng, Y. Xiaolin and J. Buyu, “A Novel Artificial Bee Colony Algorithm for Structural Damage Detection,” *Advances in Civil Engineering,* vol. 2020, 2020.

[41] J. Ye, H. Ziqing, L. Yanhai, X. Zhengyi and W. Jianming, “Weighted Global Artificial Bee Colony Algorithm,” *Sensors,* vol. 16, 2016.

[42] D. Aveek, T. Bipan, B. Rajib and B. Nabarun, “Black Tea Quality Evaluation Using Electronic Nose: An Artificial Bee Colony Approach,” *2011 IEEE Recent Advances in Intelligent Computational Systems,* pp. 143 - 146, 2011.

[43] E. G. J., G. C. F., A. M.-C. J. and M. F.-N. L., “A Clustering Method Based on the Artificial Bee Colony Algorithm for Gas Sensing,” *17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE),* vol. 17, pp. 1 - 4, 2020.

[44] Y. Xin-She and H. Xing-Shi, “Why the Firefly Algorithm Works?,” in *Nature-inspired Algorithms and Applied Optimization*, Springer, 2018, pp. 245-259.

[45] T. KuangWei, W. ZhongBin, S. Lei , T. Chao and L. Peiyang, “A Novel Pipeline Leak Recognition Method of Mine Air Compressor Based on Infrared Thermal Image using IFA and SVM,” *Applied Science,* vol. 10, pp. 1 - 22, 2020.

[46] M. DengLong, G. JianMin, Z. ZaoXiao and W. QingSheng, “An Improved Firefly Algorithm for Gas Emission Source Parameter Estimation in Atmosphere,” *Digital Object Identifier,* vol. 10, pp. 1 - 58, 2019.

[47] Q. Yousef, T. A. Mafawez, J. Ayman and N. Khalid, “A review of swarm intelligence algorithms deployment for scheduling and optimization in cloud computing environment,” *PeerJ Computer Science,* vol. 7, pp. 1 - 17, 2021.

[48] J. M. Kanda and U. M. Nur, “Particle Swarm Optimization and Fuzzy Logic Control in Gas Leakage Detector Mobile Robot,” in *International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology*, Bandung, Indonesia, 2015.

[49] A. Suychismita, S. Ipsita, S. Jashaswi, A. Anisha, M. Pranati and K. D. Ranjan, “Performance Analysis of the PSO Algorithm: An Experimental Study,” in *International Conference on Smart Data Intelligence (ICSMDI)*, Kongunadu, 2021.

[50] M. Denglong, T. Wei, W. Qingsheng, Z. Zaoxiao, G. Jianmin, Z. Qunfeng, W. Xiaoqiao, X. Fengshe and S. Xingmin, “Application and improvement of swarm intelligence optimization algorithm in gas emission source indentification in atmosphere,” *Journal of Loss Prevention in the Process Industries,* vol. 56, pp. 262 - 271, 2018.

[51] X. Zhiyu, X. Zhengyi, L. Dan and W. Jianming, “A Novel Method for Source Tracking of Chemical Gas Leakage: Outlier Mutation Optimization Algorithm,” *MDPI SENSOR,* vol. 22, no. 71, pp. 1 - 27, 2021.

[52] B. Rafael, C. d. S. Roberto , C. B. R. Alexandre and M.-C. Felix, “UAV swarm control strategies: A case study for leak detection,” in *18th Conference on Advanced Robotics (ICAR)*, Science and Technology Park, Hong Kong, 2017.

[53] H. Yu, L. Lei and J. Renxing, “A Hybrid Algorithm for Gas Source Locating Based on Unmanned Vehicles in Dynamic Gas Environment,” *Mathematical Problems in Engineering,* vol. 2021, pp. 1 - 8, 2021.

[54] F. Qilin, C. Hao, L. Fei , L. Xiaoran, L. Shichao and X. Jiheng, “An improved particle swarm optimization method for locating time-varying indoor particle sources,” *Building and Environment ,* vol. 147, pp. 146 - 157, 2019.

[55] Y. Yuting, Z. Ru, W. Ji and L. Juming, “Modified PSO algorithms with “Request and Reset” for leak source localization using multiple robots,” *Neurocomputing,* vol. 292, pp. 28 - 90, 2018.

[56] L. Mi , C. Huan, S. Xin, L. Sa, Z. Ming and L. Shengfu, “A multi-information fusion ‘‘triple variables with iteration’’ inertia weight PSO algorithm and its applciation,” *Applied Soft Computing Journal,* vol. 84, no. 2019, pp. 1 - 19 , 2019.

[57] T. Jun, L. Gang and P. QingTao, “A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems" Applications and Trends,” *Journal of Automatica Sinica,* vol. 8, no. 10, pp. 1627 - 1643, 2021.

[58] E. Kok Seng and Y. Kian Meng , “Chemical Plume Tracing: A three-dimensional technique for quadrotors by considering the altitude control of the robot in the casting stage,” *International Journal of Advanced Robotic Systems,* pp. 1- 21, 2018.

[59] Z. Yudong, A. Praveen, B. Vishal, B. Saeed and Jie Yan, “Swarm Intelligence and it's Application,” *The Scientific World Journal,* vol. VOLUME 2013, p. 3, 2013.

[60] S. Malek, “Research Gate,” Research Gate, 31 March 2018. [Online]. Available: https://www.researchgate.net/post/What_is_Velocity_in_Particle_Swarm_Optimization. [Accessed 15 September 2021].

[61] B. Fabrizio , “Towads Data Science,” 28 June 2019. [Online]. Available: https://towardsdatascience.com/swarm-intelligence-inside-the-ant-colony-9ffbce22a736. [Accessed 26 August 2021].

[62] H. Mei Jian, “USM,” USM, [Online]. Available: http://urrg.eng.usm.my/index.php/en/news-and-articles/20-articles/227-abc-artificial-bee-colony-algorithm-part-ll. [Accessed 7 September 2021].

[63] M.-L. Jauer, “Development and Implementation of an Advanced Fuel Jettison Simulation and Comparison with the ICAO Standard Procedure,” *Impact of Rapid Fuel Jettison Procedures on Air Traffic Management.,* p. Impact of Rapid Fuel Jettison Procedures on Air Traffic Management., 2018.

PRESENTED PAPER

L. Jia Jun, Y. Kian Meng, E. Kok Seng and N. Qiang, “Open Agricultural Burning Detection with Natural Inspired Swarm-based Detection Platform,” in *International Conference on Artificial Intelligence in Engineering and Technology*, Sabah, 2022.