



Article Identifying the Locations of Atmospheric Pollution Point Source by Using a Hybrid Particle Swarm Optimization

Wipawinee Chaiwino ^{1,2}, Panasun Manorot ^{1,2}, Kanyuta Poochinapan ^{2,3,4} and Thanasak Mouktonglang ^{2,3,4,*}

- ¹ Ph.D. Degree Program in Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand; wipawinee_chai@cmu.ac.th (W.C.); panasun_ma@cmu.ac.th (P.M.)
- ² Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand; kanyuta.p@cmu.ac.th
- ³ Advanced Research Center for Computational Simulation, Chiang Mai University, Chiang Mai 50200, Thailand
- ⁴ Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand
- * Correspondence: thanasak.m@cmu.ac.th

check for updates

Citation: Chaiwino, W.; Manorot, P.; Poochinapan, K.; Mouktonglang, T. Identifying the Locations of Atmospheric Pollution Point Source by Using a Hybrid Particle Swarm Optimization. *Symmetry* **2021**, *13*, 985. https://doi.org/10.3390/ sym13060985

Academic Editors: Yongwimon Lenbury, Ravi P. Agarwal and Elvin Moore

Received: 28 April 2021 Accepted: 28 May 2021 Published: 1 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Abstract: This research aims to improve the particle swarm optimization (PSO) algorithm by combining a multidimensional search with a line search to determine the location of the air pollution point sources and their respective emission rates. Both multidimensional search and line search do not require the derivative of the cost function. By exploring a symmetric property of search domain, this innovative search tool incorporating a multidimensional search and line search in the PSO is referred to as the hybrid PSO (HPSO). Measuring the pollutant concentration emanating from the pollution point sources through the aid of sensors represents the first stage in the process of evaluating the efficiency of HPSO. The summation of the square of the differences between the observed concentration and the concentration that is theoretically expected (inverse Gaussian plume model or numerical estimations) is used as a cost function. All experiments in this research are therefore conducted using the HPSO sensing technique. To effectively identify air pollution point sources as well as calculate emission rates, optimum positioning of sensors must also be determined. Moreover, the frame of discussion of this research also involves a detailed comparison of the results obtained by the PSO algorithm, the GA (genetic algorithm) and the HPSO algorithm in terms of single pollutant location detection, respectively. In the case of multiple sources, only the findings based on PSO and HPSO algorithms are taken into consideration. This research eventually verifies and confirms that the HPSO does offer substantially better performance in the measuring of pollutant locations as well as emission rates of the air pollution point sources than the original PSO.

Keywords: particle swarm optimization; multidimensional search; atmospheric model

1. Introduction

With the rapid growth of industrial and agricultural activities in the developing countries, the industrial and agricultural economic sectors are becoming the primary and secondary sectors that play a major role in countries' GDP. While the process of product manufacturing triggers the releases of pollutants into the atmosphere within a reasonable threshold, there are times when some industrial plants or farming activities illegally release pollutants into the air well beyond this threshold. During the harvest season, some farmers burn their fields after the harvesting to prepare for the next crops instead of using alternatives such as burying the stubble or residues back into the ground. Burning the fields is the fastest way to clean up the fields and it is followed by selling their products such as local wild vegetables. This contributes to a gradual increase in the practice of illegal land burning. The toxic fumes that emanate from industrial plants and this field burning practice have been identified as the main cause of air pollution, which affects the daily life and the health of people, especially in areas that are surrounded by

mountains. This issue faced by the local population has led some researchers to try to monitor the concentration of air pollution more accurately to inform those concerned [1–3].

However, if we want to be effective in our fight against pollution, we must first, eradicate the roots of the problem instead of just treating the symptoms. This means being able to identify the atmospheric pollution sources and their corresponding characteristics. This data can be used by the local authorities for issuing factories or farms with an initial warning while allowing an in-depth investigation of the real-time causes and sources of the detected pollution. Hence, there are many researchers who are interested in this topic and who have started improving some algorithms or models that can efficiently and effectively specify the exact positions of pollution sources [4-8]. The problem of identification of the pollution sources is reformulated into a highly nonlinear optimization. Heuristic optimization methods, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), have been applied to solve this class of problems. A genetic algorithm is another empirical search motivated by Charles Darwin's theory of evolution [9–11]. The algorithm imitates the procedure of natural selection where the fittest population are selected for reproduction and produce the offspring which are the next generation. The algorithm produces offspring which inherit the characteristic of their parents which will be a better generation. This simple idea can be applied for searching for the best solution. There are five important phases in GA which are initial population, fitness function, selection, crossover and mutation. In 2013, Quan-min BU [5] proposed an improved genetic algorithm for searching for pollution sources. Like GA, particle swarm optimization is another popular heuristic optimization method. Indeed, PSO is based on the social behavior or large groups, such as flying flocks of birds or fish schools. The moving direction and velocity of the particles are calculated based upon their experience together with social interaction with other particles in the swarm. The social interaction parameter such as cognitive, social and inertia parameters are also introduced in the model.

There are comparative results between the performance of GA and PSO [12,13]. Both methods are very popular mainly because of the implementation and the capability to solve complex problems, e.g., [14–18]. The disadvantage of the GA is that it has high implementation cost and usually requires a higher number of iterations and high number of elements. In particular, in this situation, the cost of each iteration is very high. Furthermore, GA usually converges towards a local optimum rather than the global optimum of the problem, while PSO tries to find the global optima because of their social interaction parameters.

More recently, to achieve effective methods for pollution management and satisfactory emergency responses, there have been many results concerning improving the estimation accuracy concerning airborne pollutant emission source information. In [19], Lang, J. proposed an inversion model which combines the hybrid particle swarm optimization and the Nelder–Mead simplex search method (PSO-NM) with the Gaussian dispersion model to identify the source and to examine the impacts of different atmospheric conditions on the identifications. In the same year [20] Li. H. employed the PSO-NM based on Gaussian puff dispersion model on a three-dimensional neighborhood topology which improves the performance of the PSO. More recently, Albani, R.A.S. [21] proposed an algorithm combining accurate dispersion models, Tikhonov regularization and gradient-descent optimization techniques estimating pollutant emission sources. Moreover, identifying the multiple sources of air pollution locations has also been studied [22,23].

In this research, the focus is to identify a location of air pollution point source both with and without emission rates by using a hybrid PSO (HPSO) on a Gaussian Plume model and a numerical scheme for PDE. Indeed, this paper attempts to apply HPSO to optimize an inverse model based on a numerical scheme of PDE. The HPSO algorithm is a computational technique obtained by improving PSO. We modify the PSO algorithm in two different ways. For the single-point source, the PSO method was first adjusted by adding a multidimensional search. This method is known as the cyclic coordinate method. Besides, a step resembling a mutation from a genetic algorithm (GA) was introduced to identify the locations in two-point sources. Moreover, sensor placement was also designed

to increase search efficiency. Finally, we apply the proposed method together with the numerical method for solving PDE. With the numerical scheme, we will be able to apply HPSO to a more complicated PDE model. This research is divided and presented as four main sections. A brief description of the basics of the PSO algorithm, GA, dichotomous search and cyclic coordinate methods constitute the first frame of discussion. Next, we demonstrate how to apply the HPSO to find the atmospheric pollution point sources. The results of this study and the appropriate parameters c_1, c_2, ω are discussed in the final part where the single point source without determining the emission rate results are compared with three algorithms consisting of HPSO, PSO and GA. It must be noted that, in the other cases, only the results gained from the original PSO and HPSO are explained.

2. Preliminaries

2.1. Introduction to Particle Swarm Optimization (PSO)

In 1995, the robust evolutionary optimization algorithm inspired by the behavior of organisms such as a flock of birds and fish schools was presented by R. Eberhart and J. Kennedy [24]. It introduced another approach to solving a nonlinear optimization. For the PSO system, an individual particle is a feasible solution in a search space looking for a globally optimal solution. The moving direction and velocity of the particles are calculated based upon their experience together with social interaction with other particles in the swarm. The social interaction parameter such as cognitive, social and inertia parameters are also introduced in the model. PSO is a computational technique using individual improvement gathering with population competition to evaluate the solutions. The algorithm is based on the simulation of the simple social behavior of some animals. In PSO, the term particle is used instead of a feasible solution, a cell swarm instead of a subset of a feasible solution and fitness function instead of an objective function. The particle adjusts its trajectory toward its own previous best position and the previous best position attained by any member of its neighborhood. The fitness function is used for deciding which particle will survive or be eliminated. The locations of all particles are shifted by the velocity equations as below.

$$V_i^{n+1} = \omega V_i^n + c_1 r_1^n (Xlbest_i^n - X_i^n) + c_2 r_2^n (Xgbest^n - X_i^n)$$
(1)

$$X_i^{n+1} = X_i^n + V_i^{n+1}$$
(2)

where

- *P* is the swarm's size.
- ω is the inertia weight.
- c_1 and c_2 are two positive constants, which are called cognitive and social parameter, respectively.
- r_1^n and r_2^n are two random numbers uniformly distributed within range [0, 1].
- $X_i^n = (x_{i1}^n, x_{i2}^n, \dots, x_{iD}^n)^T$ is the *i*-th particle of the D-dimensional in *n*-th iteration. $Xlbest_i^n = (xlbest_{i1}^n, xlbest_{i2}^n, \dots, xlbest_{iD}^n)^T$ is the best previous position of *i*-th particle in *n*-th iteration.
- $Xgbest^n = (xgbest_1^n, xgbest_2^n, \dots, xgbest_D^n)^T$ is the best position of the swarm in *n*-th iteration.
- $V_i^n = (v_{i1}^n, v_{i2}^n, \dots, v_{iD}^n)^T$ is velocity of *i*-th particle in *n* iteration.

In Equation (1), ωV_i^n is the previous particle's velocity weighted by the inertia weight ω . Xlbestⁿ_i – Xⁿ_i is the direction vector from the *i*-th particle to the best of the known *i*-th particle. The direction vector $(Xgbest_i^n - X_i^n)$ is from the *i*-th particle to the best know particle. The parameters c_1 and c_2 are the size controller of vectors in the second and the third term, respectively. For more development on PSO and HPSO, see examples in [24–28]. The particle movement and a flowchart of the PSO algorithm are presented Figures 1 and 2. Different versions of modified PSOs have also been proposed to solve different types of non-linear optimization problems. See [29-32] for examples and more details.



Figure 1. Schematic movement of a particle.



Figure 2. Flowchart of PSO algorithm.

2.2. Dichotomous Search Method

The dichotomous search also known as line search is the searching tool in one dimension for solving a non-linear programming problem. Consider the function $f : \mathbf{R} \to \mathbf{R}$ to be minimized over the interval [a, b]. First, we pick a small number $\varepsilon > 0$. Then, according to the flowchart Figure 3, the Dichotomous search involves two steps as follows.

Initialization step: Choose the small constant $2\varepsilon > 0$ and the final length of $\delta > 0$. Let $[a_1, b_1]$ be the uncertain initial interval. Give k = 1 and then go to main step. Main step :

1. Consider λ_k and μ_k given below.

$$\lambda_k = rac{a_k + b_k}{2} - arepsilon \qquad \mu_k = rac{a_k + b_k}{2} + arepsilon$$

2. If $f(\lambda_k) < f(\mu_k)$ then $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$. Otherwise, let $a_{k+1} = \lambda_k$ and $b_{k+1} = b_k$.

3. If
$$b_k - a_k < \delta$$
, stop. Give $k = k + 1$ and do step 1.



Figure 3. Flowchart of dichotomous search method.

2.3. Cyclic Coordinate Method

The cyclic coordinate method is the multidimensional search without the assistant of the derivative. The objective function $f : \mathbf{R}^n \to \mathbf{R}$ is the minimized form feasible solution x along the suitable direction d by the Dichotomous search technique. More specifically, we define the feasible solution

$$x = [x_1, x_2, ..., x_n]^T$$

to be an *n* dimensional vector and let the $d = [d_1, d_2, ..., d_n]^T$ be a zero vector in *n* dimensions except for a 1 at the position that we are investigating. Therefore, the variable x_i is updated by $x_i + d_i$, while the other variables are kept fixed. The summary below of this method for minimizing the objective function *f* of multi-variables and a flowchart (Figure 4) are illustrated.

Initialization step: Select a scalar $\varepsilon > 0$ to be used for terminating the method. Let $d = [d_1, d_2, ..., d_n]$ be the searching small direction and choose the initial point x_1 . Before going to the main step we have to substitute x_1 for y_1 and let k = j = 1. Main step:

1. Begin with finding the optimal solution λ from minimizing the problem

$$f(y_j + \lambda d_j)$$

subject to $\lambda \in \mathbf{R}$

and then let $y_{j+1} = y_j + \lambda$

- 2. If j < n, replace j by j + 1 and repeat step 1. Otherwise, if j = n, go to step 3.
- 3. Let $x_{k+1} = y_{n+1}$. If $x_{k+1} x_k < \varepsilon$, then stop. Otherwise, give $y_1 = x_{k+1}$ and j = 1, surrogate k by k + 1 and go to step 1.



Figure 4. Flowchart of cyclic coordinate method.

3. Methodology

where

3.1. Hybrid Particle Swarm Optimization Method (HPSO)

Our hybrid PSO is inspired by a Genetic algorithm (GA) by which each particle can be randomly mutated or evaluated to get to a new region of search domains. For that reason, we incorporate the dichotomous search and the cyclic coordinate search into our PSO. Furthermore, in order to avoid being stuck in some local optimal, we introduce additional procedure to look for another possible local optimal; the procedure is explained as follows.

3.1.1. Optimization Problem Formulation

We begin with providing definitions of the variables in a particle. A particle represents a location of all n contamination point sources (m) and their emission rates (kg/s). Therefore, the *i*-th particle

$$X_i = (x_i, y_i, H_i, Q_i)$$

$$\begin{aligned} x_i &= [x_i^1, x_i^2, ..., x_i^n] \\ y_i &= [y_i^1, y_i^2, ..., y_i^n] \\ H_i &= [h_i^1, h_i^2, ..., h_i^n] \\ Q_i &= [q_i^1, q_i^2, ..., q_i^n] \end{aligned}$$

when q_i^j is the emission rate corresponding the pollution source *j*th, which is located at (x_i^j, y_i^j, h_i^j) of the *i*th particle. For all i = 1, 2, 3, ..., p, *p* is a number of particles (swarm) and *n* is a number of sources. The fitness function is defined to be a summation of difference between measured and approximated concentration square. In this experiment, we used the Gaussian Plume model for simplicity. Therefore, the approximated concentration C_T is the summation of concentration from *n* different pollution sources. Note that we could use other techniques of approximation in the experiment. $C_T(x_i, y_i; H_i, Q_i)$ is in kg/m³ from the Gaussian Plume model, which is defined by

$$F(x_i, y_i; H_i, Q_i) = \sum_{i=1}^{s} [\mu_i - C_T(x_i, y_i; H_i, Q_i)]^2, \text{ where } s \text{ is the number of sensor.}$$
(3)

where

$$C_T(x,y;H,Q) = \sum_{j=1}^n C(\bar{x},\bar{y},z;H,Q),$$

with

$$C(\bar{x},\bar{y},z;H,Q), = \frac{Q}{4\beta u\sqrt{r_y r_z}} e^{\left(\frac{-y}{4r_y}\right)} \left[e^{\left(-\frac{(z-H)^2}{4r_z}\right)} + e^{\left(-\frac{(z+H)^2}{4r_z}\right)} \right]$$

 $\bar{x} = xs - x, \bar{y} = ys - y$ are the shifted coordinates, so the position of source corresponds to $(0, 0, H_i)$ and

$$r_y(x) = \frac{1}{u} \int_0^x K_y(\xi) d\xi$$
$$r_z(x) = \frac{1}{u} \int_0^x K_z(\xi) d\xi$$

where *u* is wind velocity (m/s), K_y and K_z are diffusion coefficients of *y* and *z* directions, respectively [22]. This close form can be used under the ensuing assumptions.

- The foul gas is discharged at a constant rate Q (kg/s) form the source $\vec{x} = (0, 0, H)$, which is placed at height H above the ground surface.
- The wind velocity is constant and aligns in the x-axis direction, which is written as $\vec{u} = (u, 0, 0)$ when $u \in [0, 5]$ has a unit of m/s.
- The solution is in steady state, which makes the wind velocity and other functions independent of time.

The HPSO for Identifying the Single Air Pollution Location and Its Emission Rate

We start with ranking the fitness value of all particles. The first two particles X_1 and X_2 are selected in order to create a new particle by finding the midpoint of the two particles. The middle point is taken as a starting point in the search along the direction d, where d is $\frac{X_1 - X_2}{\|X_1 - X_2\|}$. Then go to a selection step; we evaluate the fitness cost of the new particle and select the particles equal to the number of a swarm by ranking the fitness value. Which particle has the most fitness value will be eliminated.

The HPSO for Identifying the Two Air Pollution Locations and Their Emission Rates

For the case of two pollutant sources, we specify just the locations of the source. We adapt the original PSO with the step as follows. In the first step, we rank the fitness cost of each particle and choose the first two particles to create the new particle by the steps below.

Let the first two particles be $X_1 = [x_1^1, x_1^2, y_1^1, y_1^2]^T$ and $X_2 = [x_2^1, x_2^2, y_2^1, y_2^2]^T$. The new particle X_{new} is with the new random parameter $\alpha \in [0, 100]$ where

$$(x_{new}^{1}, y_{new}^{1}) = \alpha \frac{(x_{1}^{1} + x_{2}^{1}, y_{1}^{1} + y_{2}^{1})}{2}$$
$$(x_{new}^{2}, y_{new}^{2}) = \alpha \frac{(x_{1}^{2} + x_{2}^{2}, y_{1}^{2} + y_{2}^{2})}{2}$$
$$X_{new} = [x_{new}^{1}, x_{new}^{2}, y_{new}^{1}, y_{new}^{2}]^{T}$$

After obtaining the new particles, the particle that provides the worst fitness value is removed. The improved structure of the PSO algorithm is indicated as the following Algorithm 1.

Algorithm 1 HPSO algorithm.

- 1: **Initailize** X_i , V_i and $Xlbest_i$ for each particle *i*
- 2: **while** (not termination condition)
- 3: **for** each particle *i*
- 4: Evaluate objective function;
- 5: Update *Xlbest_i* and *Xgbest*
- 6: end for.
- 7: **for** each *i*
- 8: calculate V_i;
- 9: update $X_i = X_i + V_i$;
- 10: Evaluate objective function;
- 11: Choose the first two best particles to create two new particles
- 12: Creating a new particle
- 13: Evaluate objective function of two new particle and choose the
- 14: next generation of the particles
- 15: end for
- 16: end while

4. Numerical Results

The experiments are designed to investigate the ability of the HPSO algorithm; they are separated into two main parts compose of two-dimensional and three-dimensional domain problems. The results of the identification of the air pollution point source location and its corresponding emission rate are also presented and discussed in two and three-dimension problems. Furthermore, we not only identify the single air pollution point source location but also determine the locations of two-point sources in the two-dimensional domain. For two-point sources, the experiments are designed into two cases. One is to identify the locations of two sources when positioning the source locations beside each other and the other is to identify the locations of two sources when placing source locations in overlapping positions. In the original PSO the parameters c_1 , c_2 and ω play an important role in searching for the optimal solution. This means the relevant c_1 and c_2 help the particles keep a balance among the *Xlbest*, *Xgbest* and *V* directions to find the optimal solution. Besides, the appropriate ω increases a chance to find the optimal solution (reduce the change to stick at the local optimal). Therefore, the value ω and c ($c = c_1 = c_2$) are determined for each experiment. For some problems, there are many values of ω and cthat are able to provide the optimal solution. However, the values ω and c produce the best results that are picked; these parameters are searched for in the set $\{0.1, 0.2, 0.3, ..., 1\}$ and the most appropriate parameters of all experiments are shown in Table 1.

The Experiment in 2D				
Problem case	ω (PSO)	c (PSO)	ω (HPSO)	c (HPSO)
Specifying single-source locations	0.7	0.5	0.4	0.7
Specifying two-source locations	0.8	0.5	0.6	0.3
Specifying single-source locations and its corresponding emission rate	0.8	0.6	0.8	0.4
The Experiment in 3D				
Specifying single-source locations	0.7	0.3	0.5	0.7
Specifying single-source locations and its corresponding emission rate	0.7	0.5	0.6	0.2

Table 1. Appropriate parameters of all experiments.

This problem has to do with the identification of the position of atmospheric pollution on the domain size 50,000 m \times 35,000 m with light wind (5 m/s). The concentration is emitted with a rate of 35 kg/s by a single pollution point source, which is located at (0, 0) coordinate. For identification of the two-point sources of air pollution, we already mentioned that there are two experiments; the first experiment sources are located at (300, -5000) and (4000, 5000) with emission rates 35 kg/s and 40 kg/s, respectively. In the second experiment, the source locations are moved to (200, 0) and (3000, 0) with the same emission rates.

Firstly, we verify that the number of iterations and the number of particles can reduce the distance error of HPSO by fixing the number of iterations (100 iterations) while the number of particles is increased by 10 from 10 to 250 and by fixing the number of particles (100 particles) while the number of iteration is increased by 10 from 10 to 250. The results are shown in Tables 2 and 3, respectively. Both sets of results illustrate that, as the number of iterations and particles exceeds 150, the distance error remains homogeneous. In all experiments, mean square error is used to measure the effectiveness of the algorithms.

	HPS	HPSO		PSO	
The Number of Particles	Average Distance Error (m)	Computational Time (s)	Average Distance Error (m)	Computational Time (s)	
10	3218.63	0.71610	7689.64	0.268653	
20	2628.88	0.86134	2631.55	0.479630	
30	1175.62	1.69867	1942.41	0.716689	
40	699.23	2.06156	682.51	0.907543	
50	238.14	2.47007	517.01	1.136285	
60	50.04	2.87347	764.96	1.342402	
70	3.74	3.27136	461.86	1.562531	
80	1.41	3.66756	407.51	1.767342	
90	0.37	4.08449	351.72	2.004292	
100	0.17	4.45915	170.14	2.189354	
110	0.04	4.86884	120.02	2.394523	
120	0.0042	5.28437	90.31	2.642548	
130	$1.12 imes10^{-4}$	5.71370	76.18	2.856800	
140	$5.29 imes10^{-5}$	6.09393	70.85	3.057815	
150	$2.10 imes10^{-5}$	6.51467	40.93	3.264904	
160	$1.91 imes 10^{-5}$	6.60431	40.01	3.510096	

Table 2. Comparing results of HPSO when the number of iterations is fixed.

Н		50	PS	0
The Number of Particles	Average Distance Error (m)	Computational Time (s)	Average Distance Error (m)	Computational Time (s)
170	$9.41 imes10^{-6}$	6.95562	20.18	3.730219
180	$6.20 imes 10^{-6}$	7.36398	21.12	3.900210
190	$5.39 imes 10^{-6}$	7.69186	10.22	4.140191
200	$3.03 imes10^{-6}$	8.13077	10.36	4.390146
210	$1.00 imes10^{-6}$	8.69686	11.47	4.579430
220	$8.71 imes10^{-6}$	8.91746	15.31	4.776681
230	$3.10 imes 10^{-7}$	9.17680	12.47	4.964160
240	$7.09 imes 10^{-5}$	9.47666	20.21	5.152613
250	$2.64 imes10^{-5}$	9.70849	8.98	5.570200

Table 2. Cont.

Table 3. Comparing results of HPSO when the number of particles is fixed.

	HPS	50	PS	0
The Number of Iterations	Average Distance Error (m)	Computational Time (s)	Average Distance Error (m)	Computational Time (s)
10	4195.36	0.437180	14,840.15	0.2475500
20	1640.33	0.854225	12,164.85	0.4676400
30	236.27	1.284434	11,535.78	0.6864380
40	34.92	1.643332	5759.85	0.9030310
50	11.34	2.027071	3168.66	1.1229110
60	11.18	2.422206	1125.05	1.3300200
70	3.37	2.851604	913.43	1.5588340
80	0.94	3.259195	673.01	1.7654640
90	0.35	3.625978	418.90	1.9691390
100	0.20	4.022224	125.63	2.2105750
110	0.0059	4.462938	110.10	2.4005550
120	$1.20 imes 10^{-4}$	4.866495	109.20	2.6671640
130	$2.00 imes 10^{-5}$	5.320598	90.04	2.8114300
140	$9.71 imes10^{-6}$	5.640714	75.00	3.0862320
150	$1.80 imes 10^{-6}$	6.042532	65.51	3.2879560
160	$1.20 imes10^{-6}$	6.435387	30.60	3.5156750
170	$1.00 imes10^{-6}$	6.872544	30.10	3.7261230
180	$9.26 imes10^{-7}$	7.270416	20.46	3.9712130
190	$4.16 imes10^{-6}$	7.745574	20.39	4.1502190
200	$1.19 imes 10^{-6}$	8.177410	11.66	4.3569590
210	$4.07 imes 10^{-6}$	8.480385	17.41	4.5510942
220	$7.54 imes10^{-6}$	8.929684	14.55	4.8100910
230	$4.10 imes10^{-6}$	9.183009	21.95	5.0166410
240	$3.02 imes 10^{-6}$	9.468920	4.12	5.2980970
250	$4.27 imes10^{-6}$	9.711908	31.09	5.4158020

Next, we point out that, by increasing the number of sensors, the distance error is also reduced. In these experiments, we increase the number of sensors by 2 and observe the results, which are manifested as Table 4.

From Table 4, when the number of sensors is more than two, the distance errors are insignificantly different. So there is good reason to believe that the number of sensors need not be so high. Thus, for all experiments, except two-point sources, in this research we use 100 particles, 200 iterations with four sensors.

The Number of Sensors	Average Distance Error (m)		
	PSO	HPSO	
2	5912.48	1109.41	
4	15.42	$1.12 imes 10^{-5}$	
6	12.41	$4.38 imes10^{-6}$	
8	15.89	$2.48 imes10^{-6}$	
10	10.24	$1.87 imes10^{-7}$	

Table 4. The HPSO results with various numbers of sensors.

4.1. Complexity

In this sub-section, the comparative study of the complexity between the original PSO and our proposed HPSO is discussed. On each iteration of the HPSO, the dichotomous search or the cyclic coordinate method is executed. Let p be the number of particles used in PSO and HPSO. Let also *M* be the maximum number of iterations for both PSO and HPSO. Then in the original PSO, the number of evaluations of the objective/fitness function is $p \times M$. As for the HPSO, each of the inner iterations for the dichotomous search or the cyclic coordinate method concerning the fitness function has to be evaluated twice. If we assume that the maximum number of the inner iterations is *m*, then the number of evaluations of the objective/fitness function on our HPSO is $2p \times m \times M$. It is clear that, on each iteration, the implementation cost of HPSO is higher than PSO. Hence, if the maximum number of iterations is fixed, HPSO takes more computational time. However, the new proposed HPSO is much more effective. To illustrate this point, consider Table 2, where HPSO and PSO are compared for a fixed number of iterations. According to this table, for PSO with 210 particles, the error is about 11.47 m and the computational time is about 4.58 seconds. As for HPSO, there is about the same computational time but with fewer particles—100 particles; the error is only 0.17 m. Hence, even each iteration for HPSO has a higher computation cost above the original PSO; overall, HPSO is considered superior to PSO, especially for situations for which a minimal number of iterations is required.

4.2. The Results in Two-Dimensional Domains

The identification of the location of single and multiple air pollution point sources is tested. For comparing the results, the initial data such as the initial particles/chromosomes, the number of particles/chromosomes, the number of sensors and the sensor's positioning are set to be the same. The results of the three algorithms explicate that the HPSO has the lowest average distance error followed by the original PSO and the original GA. Observe that the HPSO method does not stick at some local optima; consequently, HPSO produces less average distance error than PSO and GA.

4.2.1. Identification the Air Pollution Locations without Finding Emission Rate in Two Dimensions

The results of determining the location of a single air pollution point source are displayed in Figures 5 and 6. The comparative results between GA, PSO and HPSO are illustrated in Figures 5 and 6. Five experiments on each technique were randomly selected and plotted as a line graph on the left and the bar graph on the right displays the average for each technique. As we expect, HPSO outperforms the other techniques. In this experiment, GA has the lowest ability to locate the pollution point source among all three techniques. As for determining the locations of two air pollution point sources, results are displayed as Figures 7–10. In these experiments, 300 particles and 500 iterations with four sensors are used for running the program. In both cases of the experiment, the HPSO method processes lower average distance errors than the PSO method. In the first case, the best predicted locations that provide the best average error of the distance errors are (332.16, -5001.74) and (3784.24, 4986.78). In the second case, the best predicted locations are (320.07, 208.84)

and (2907.65, 159.45). Notice that the average distance errors of the HPSO in the case of putting the source location in an overlap of each other is higher than the other case. Note that, for the case of multiple point source, the problem is much more complex than a single point source. The number of point source and emission rates of each point source need to be prescribed. Each point source can have different emission rates.







Figure 6. Optimal location of predicted single-point air pollution location (HPSO).



Figure 7. The result of specifying only the locations of two air pollution point sources in the case of positioning the source locations beside each other.



Figure 8. Optimal locations of predicted double-point air pollution locations.



Figure 9. The result of specifying only the locations of two air pollution point sources in the case of placing sensors in overlapping positions.



Figure 10. Optimal locations of predicted air pollution location of placing sensors in overlapping positions.

Specifying the location together with its corresponding emission rate of the air pollution point source is the last experiment in the two-dimension domain problem. We use 100 particles and 200 iterations to run the program and the results of the PSO method and HPSO method are exhibited as Figures 11 and 12. The HPSO takes an average distance error 16 times less than the PSO and provides an average emission rate error about 24 times lower than the PSO. Hence, we can conclude that our HPSO outperforms PSO. So, we can conclude that our HPSO outperforms the original PSO.



Figure 11. The result of specifying the location and its corresponding emission rate of single air pollution point source.



Figure 12. Optimal locations of air pollution point sources with emission rate Q.

4.3. The Result in Three Dimensional Domains

For the problem in three-dimensional domains, we set the exact source location at (0, 0, 20) on domain size 35,000 m \times 50,0000 m \times 50 m with the emission rate 35 kg/s. The frame of this problem is divided into two parts which are to identify only the pollution source location and to identify both the location and the emission rate of the pollution source. For the three-dimensional domain experiments, we use 100 particles with 200 iterations.

4.3.1. Identification the Air Pollution Locations and Emission Rate in Three Dimensions

Figures 13 and 14 present the result for determining the location of a single-point of the pollutant source and its emission rate. The minimum average error of both items belongs to the HPSO method. Our HPSO outperforms the original PSO.



Figure 13. The result of specifying the location and its corresponding emission rate of single air pollution point source in 3D.



Figure 14. Optimal locations of air pollution point sources in 3D when its emission rate is also predicted.

4.3.2. Identification the Air Pollution Locations without Finding Emission Rate in Three Dimensions

According to Figures 15 and 16, the HPSO algorithm produces an average distance error 29 times less than the PSO algorithm. The HPSO's computational time is a little bit higher than the PSO's. In addition, Figure 16 displays the best predicted location of air pollution point source using the HPSO, which is located at (0.01, 0.02, 20.05).



Figure 15. The result of specifying only the location of single air pollution point source in 3D.



Figure 16. Optimal locations of air pollution point sources in 3D.

According to all of the experiments using the Gaussian Plume model, HPSO outperforms PSO. Overall, it is more effective. In order to achieve effective results for pollution management and satisfactory emergency responses, the Gaussian Plume model is not enough. We need to approximate the numerical solution of the set of some governing PDE system. Solving for an estimated solution is very important.

4.4. The Result with a Numerical Method in PDE

In this numerical experiment, we focus on the couple models for pollutant transport, that is, the momentum equations [7,22]:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = v \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right],\tag{4}$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = v \left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right],\tag{5}$$

and the convection-diffusion equation:

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} + v \frac{\partial C}{\partial y} = k \left[\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right] + S, \tag{6}$$

with the following initial-boundary conditions

$$u(x,y,0) = u_0(x,y), v(x,y,0) = v_0(x,y), C(x,y,0) = C_0(x,y)$$

$$u(x,y,t) = f(x,y,t), v(x,y,t) = g(x,y,t), C(x,y,t) = h(x,y,t), \forall (x,y) \in \partial\Omega,$$

where u(x, y, t), v(x, y, t) are horizontal wind velocity (m/s), C(x, y, t) is the concentration of the pollutant (kg/m³), S(x, y) is the source term (kg/m³s), v is kinetic viscosity (m²/s) and k(x, y) is conductivity (m²/s). In this work, we are interested in point-source behaviour. For the *n* point-source, the source term is given by

$$S(x,y) = \sum_{i=1}^{n} Q_i \delta_{(x_i,y_i)}(x,y),$$

where Q_i is the mission rate of the *i*th source and the delta function is defined by

$$\delta_{(x_i, y_i)}(x, y) = \begin{cases} 1, \text{ if } x = x_i \text{ and } y = y_i \\ 0, \text{ else.} \end{cases}$$

The conductivity function k(x, y) represents the ability according to which the pollutant goes through the median. Thus, the conductivity function depends on the space variables, i.e., the area on the domain. In this study, we assume that the pollutant is emitted in a homogeneous area; that is, the conductivity is constant: $k(x, y) = k_0$. This assumption can be applied when the source is located in the high area and releases the pollutant to the lower region.

4.4.1. Numerical Method

For the rectangular domain $\Omega = [x_L, x_R] \times [y_L, y_R]$, we define the discrete space $\{(x_i, y_j)|x_i = x_L + ih_x, y_j = y_L + jh_y\}$, where $h_x = (x_R - x_L)/N_x$ and $h_y = (y_R - y_L)/h_y$. The temporal domain [0, T] is discretized as $\{t_n = n\tau\}$, where $\tau = T/N$. Let $f_{ij}^n = f(x_i, y_j, t_n)$, the discrete differential operators are given below:

$$(f_{ij}^n)_{\hat{t}} = \frac{f_{ij}^{n+1} - f_{ij}^{n-1}}{2\tau}, \quad (f_{ij}^n)_x = \frac{f_{i+1,j}^n - f_{ij}^n}{h_x}, \quad (f_{ij}^n)_{\bar{x}} = \frac{f_{ij}^n - f_{i-1,j}^n}{h_x}$$
$$(f_{ij}^n)_y = \frac{f_{i,j+1}^n - f_{ij}^n}{h_y}, \quad (f_{ij}^n)_{\bar{y}} = \frac{f_{ij}^n - f_{i,j-1}^n}{h_y}, \quad (\bar{f}_{ij}^n) = \frac{f_{ij}^{n+1} + f_{ij}^{n-1}}{2}$$

The numerical scheme for models (4)–(6) is:

$$(u_{ij}^{n})_{\hat{t}} + u_{ij}^{n}(\bar{u}_{ij}^{n})_{\hat{x}} + v_{ij}^{n}(\bar{u}_{ij}^{n})_{\hat{y}} = \nu \left[(\bar{u}_{ij}^{n})_{x\bar{x}} + \bar{u}_{ij}^{n})_{y\bar{y}} \right]$$
(7)

$$(v_{ij}^{n})_{\hat{t}} + u_{ij}^{n}(\bar{v}_{ij}^{n})_{\hat{x}} + v_{ij}^{n}(\bar{v}_{ij}^{n})_{\hat{y}} = \nu \left[(\bar{v}_{ij}^{n})_{x\bar{x}} + \bar{v}_{ij}^{n})_{y\bar{y}} \right]$$
(8)

$$(C_{ij}^{n})_{\hat{t}} + u_{ij}^{n}(\bar{C}_{ij}^{n})_{\hat{x}} + v_{ij}^{n}(\bar{C}_{ij}^{n})_{\hat{y}} = k \left[(\bar{C}_{ij}^{n})_{x\bar{x}} + \bar{C}_{ij}^{n})_{y\bar{y}} \right] + S_{ij}^{n}.$$
(9)

The scheme is second order in both time and space.

4.4.2. Results of Identifying Single Pollution Point Source

In this experiment, giving an initial guess of the pollution source, we use the above numerical scheme to solve a system of partial differential equations for atmospheric model for concentration on the domain space (x, y). As a result, for each particle on each iteration, solving for the system of PDE is required for evaluating a fitness function. Hence, a fast and accurate numerical method is needed.

- Domain size: 10 × 2
- The number of particles is 10
- The number of iterations is 30
- Exact location is (0.5, 1)

We considered three cases for different numbers of time steps. In each case, the program was run five times and then the average error was calculated. The best predicted location is presented in Table 5. According to the results of the experiment, the HPSO is also capable of identifying the location of the point source based on an approximated solution of a system of PDE. The Figures 17–19 illustrate how the algorithm identifies the location of the pollution source.

Table 5. Result of Single Pollution Point Source.

Time Steps	The Best Predicted Location	Average Error
20	(0.548, 1.785)	0.0787
40	(0.499, 0.899)	0.0154
60	(0.500, 1.015)	0.0109



Figure 17. Optimal locations of air pollution point source in 2D with 20 time steps.



Figure 18. Optimal locations of air pollution point source in 2D with 40 time steps.



Figure 19. Optimal locations of air pollution point source in 2D with 60 time steps.

5. Conclusions

This research paper is another comparative study of GA and PSO. Due to the nature of the problems, PSO actually outperforms GA. PSO is able to locate the pollution source more accurately. The results on the Gaussian Plume model are very fascinating. However, in order to apply the optimization techniques to the differential models for pollutant transport, a more effective technique must be used. To evaluate a fitness function for each particle, estimating a solution of a system of PDE is required. In this work, we estimate a solution

of a system of partial differential equations for the atmospheric model for concentration on the domain space. It turns out that all of the experiment results explicate that HPSO produces better results than GA and PSO. Besides, HPSO is able to identify the location and its emission rate more accurately when we increase the number of particles and iterations with appropriate parameters ω and c. While the HPSO performs with very high efficiency for specifying the location and its emission rate of single air pollution, in the case of two sources the HPSO gives quite high distance errors. The HPSO is also applicable to a more complicated system. We developed a second order in both space and time numerical schemes for solving pollutant transport equations, system of PDEs. As in the experiment with the numerical method in PDEs, the HPSO method is also able to accurately locate the pollution source. This illustrates that HPSO can be used to optimize the inverse model based on the numerical scheme for PDE. As for future work, a fast numerical scheme for solving PDE could be developed.

Author Contributions: Conceptualization, T.M.; Formal analysis, K.P.; Investigation, W.C. and K.P.; Methodology, W.C. and T.M.; Project administration, T.M.; Software, W.C., P.M. and T.M.; Supervision, T.M.; Writing—original draft, W.C.; Writing—review and editing, K.P. and T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Centre of Excellence in Mathematics, The Commission on Higher Education, Thailand and by Chiang Mai University, Chiang Mai Thailand.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request. All the computer codes used in this study are available from the corresponding author upon request.

Acknowledgments: This research was partially supported by the Centre of Excellence in Mathematics, The Commission on Higher Education, Thailand and by Chiang Mai University, Chiang Mai Thailand.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Khedo, K.K.; Perseedoss, R.; Mungur, A. A wireless sensor network air pollution monitoring system. arXiv 2010, arXiv:1005.1737.
- 2. Hasenfratz, D.; Saukh, O.; Sturzenegger, S.; Thiele, L. Participatory air pollution monitoring using smartphones. *Mob. Sens.* **2012**, 2, 1–5.
- James, J.Q.; Li, V.O.K.; Lam, A.Y.S. Sensor deployment for air pollution monitoring using public transportation system. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation (CEC), Brisbane, Australia, 10–15 June 2012; pp. 1–7.
- 4. Cervone, G.; Franzese, P. Non-Darwinian evolution for the source detection of atmospheric releases. *Atmos. Environ.* **2011**, *45*, 4497–4506. [CrossRef]
- 5. Bu, Q.; Wang, Z.; Tong, X. An improved genetic algorithms for searching for pollution sources. *Water Sci. Eng.* **2013**, *6*, 392–401.
- Cantelli, A.; D'orta, F.; Cattini, A.; Sebastianelli, F.; Cedola, L. Application of genetic algorithm for the simultaneous. identification of atmospheric pollution sources. *Atmos. Environ.* 2015, *115*, 36–46. [CrossRef]
- Sharan, M.; Yadav, A.K.; Singh, M.P.; Agarwal, P.; Nigam, S. A Mathematical Model for the Dispersion of Air Pollutants in Low Wind Conditions. *Atmos. Environ.* 1996, 30, 1209–1220. [CrossRef]
- 8. Chaiwino, W.; Mouktonglang, T. Identication of Atmospheric Pollution Source Based on Particle Swarm Optimization. *Thai J. Math.* **2019**, *17*, 125–140.
- 9. Deb, K. An introduction to Genetic Algorithms. Sadhana 1999, 24, 293–315. [CrossRef]
- 10. Sivanandam, S.N.; Deepa, S.N. Introduction to Genetic Algorithms; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
- 11. Houck, C.R.; Joines, J.; Kay, M.G. A Genetic Algorithm for Function Optimization: A Matlab Implementation. *Ncsu-ie tr* **1995**, *95*, 1–10.
- 12. Razvan, C. Comparison between the Performance of GA and PSO in Structural Optimization Problems. *Am. J. Eng. Res. (AJER)* **2016**, *5*, 268–272.
- 13. Shahid, S.; Ruchi, S. A Comparative Study of Genetic Algorithm and the Particle Swarm Optimization. *Int. J. Electr. Eng.* **2016**, *9*, 215–223.

- 14. Das, P.K.; Behera, H.S.; Panigrahib, B.K. A hybridization of an Improved Particle Swarm Optimization and Gravitational Search Algorithm for Multi-robot Path Planning. *Swarm Evol. Comput.* **2016**, *28*, 14–28. [CrossRef]
- 15. Fernandes, C.; Pontes, A.J.; Viana, J.C.; Gaspar-Cunha, A. Using multiobjective evolutionary algorithms in the optimization of operating conditions of polymer injection molding. *Polym. Eng. Sci.* **2010**, *50*, 1667–1678. [CrossRef]
- 16. Fernandes, C.; Pontes, A.J.; Viana, J.C.; Gaspar-Cunha, A. Using Multi-objective Evolutionary Algorithms for Optimization of the Cooling System in Polymer Injection Molding. *Int. Polym. Process.* **2012**, *27*, 213–223. [CrossRef]
- 17. Fereshteh, S.; Nima, J. Service Allocation in the Cloud Environments Using Multi-objective Particle Swarm Optimization Algorithm Based on Crowding Distance. *Swarm Evol. Comput.* **2017**, *35*, 53–64.
- Shahid, S.; Ruchi, S. Multi-Objective Optimization of Gate Location and Processing Conditions in Injection Molding Using MOEAs: Experimental Assessment. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization (EMO 2015), Guimarães, Portugal, 29 March–1 April 2015; pp. 373–387.
- 19. Cui, J.; Lang, J.; Chen, T.; Cheng, S.; Shen, Z.; Mao, S. Investigating the Impacts of Atmospheric Diffusion Conditions on Source Parameter Identification based on an Optimized Inverse Modelling Method. *Atmos. Environ.* **2019**, 205, 19–29. [CrossRef]
- Li, H.; Zhang, J.; Yi, J. Computational Source Term Estimation of the Gaussian Puff Dispersion. Soft Comput. 2019, 23, 59–75. [CrossRef]
- Albani, R.A.S.; Albani, V.V.L.; Silva, N.A.J. Source Characterization of Airborne Pollutant Emissions by Hybrid Metaheuristic/ Gradient-based Optimization Techniques. *Environ. Pollut.* 2020, 267, 115618. [CrossRef]
- 22. Stockie, J.M. The Mathematics of Atmospheric Dispersion Modeling. SIAM Rev. 2011, 53, 349–375. [CrossRef]
- 23. Sharan, M.; Singh, S.K.; Issartel, J.P. Least square data assimilation for identification of the point source emissions. *Pure Appl. Geophys.* **2012**, *169*, 483–497. [CrossRef]
- 24. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995.
- 25. Bai, Q. Analysis of Particle Swarm Optimization Algorithm. Comput. Inf. Sci. 2010, 3, 180–184. [CrossRef]
- Shi, Y.; Eberhart, R. A Modified Particle Swarm Optimizer. In Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
- Clerc, M. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 3, pp. 1951–1957.
- Eberhart, R.; Shi, Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In Proceedings of the 2000 Congress on Evolutionary Computation, CEC00 (Cat. No.00TH8512), La Jolla, CA, USA, 16–19 July 2000; Volume 1, pp. 84–88.
- Alanis-Tamez, M.D.; López-Martín, C.; Villuendas-Rey, Y. Particle Swarm Optimization for Predicting the Development Effort of Software Projects. *Mathematics* 2020, 8, 1819. [CrossRef]
- Zhang, M.; Long, D.; Qin, T.; Yang, J. A Chaotic Hybrid Butterfly Optimization Algorithm with Particle Swarm Optimization for High-Dimensional Optimization Problems. *Symmetry* 2020, *12*, 1800. [CrossRef]
- Liang, X.; Li, X.; Ercan, M. A PSO—Line Search Hybrid Algorithm. In Proceedings of the Conference: Computational Science and Its Applications—ICCSA 2009, International Conference, Seoul, Korea, 29 June–2 July 2009; pp. 547–556.
- Liu, Y.; Qin, Z.; Shi, Z. Hybrid Particle Swarm Optimizer with Line Search. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), The Hague, The Netherlands, 10–13 October 2004; Volume 4, pp. 3751–3755.