Modified Niche PSO with Flow of Wind for Multiple Odor Source Localization Problems in Dynamic Environments: Simulation and Measurement

Wisnu Jatmiko¹, R. Efendi¹, A. Nugraha¹, B. Kusumoputro¹, J. Perkasa¹, A. Buono², K. Sekiyama³, and T. Fukuda³

¹Faculty of Computer Science, University of Indonesia, Indonesia
(Phone: (62)-021-786-3419; e-mail: wisnuj@cs.ui.ac.id)
²Department of Computer Science, Bogor Agriculture Institute, Indonesia
³Dept. of Micro-Nano Systems Engineering, Nagoya University, Japan

Abstract

A new algorithm based on Modified Particle Swarm Optimization (MPSO) which follows a local gradient of the chemical concentration within a plume and follow direction of the wind velocity is investigated. Moreover the niche characteristic also adopted for solving the multi peaks and multi source problems. Simulations illustrate that the new approach can solve Advection-Diffusion odor model problems in such multi peaks and multi source problem. Then the statistical analysis shows that the new approach is technically sounds.

1. Introduction

Many issues have hindered odor source localization in the past. One of the common issues has been that most detection of chemicals with mobile robots has been based on experimental setups where the distance between the source and the sensor following an odor trail has been minimized to limit the influence of turbulent transport [1-2]. Another issue has been that of basing systems on the assumption of a strong, unidirectional air stream in the environment [3-6]. However, thus far not much attention has been paid to the issue of odor localization within a natural environment.

The natural environment presents two major problems that are addressed in this paper. The first problem is that in natural environments the distribution of odor molecules is usually dominated by turbulence, rather than diffusion. The other problem is the influence of the wind, which is unstable in both force and direction. Thus, when odor distribution is very complex due to turbulent flow and wind instability, current mobile robotic odor detection systems perform poorly [1-6].

To combat these natural environment issues, a new approach exploiting Particle Swarm Optimization (PSO) is presented in this paper. The PSO algorithm here is modified to include chemotaxic and anemotaxic theory along with the development of an Advection-Diffusion odor model. This Modified Particle Swarm Optimization (MPSO) is applied by multiple mobile robots which localize an odor source in natural environment where the odor distribution changes over time [7,8]. The results showed that the MPSO was capable of solving single odor source localization. However, facing multi odor source localization problem, this method failed. Then the niche characteristic will be adopted for solving the multi peaks and multi source problems.

Solving odor source localization problems in dynamic environments requires hardware and software platforms [7-11]. During the initial design stages, software evaluation is preferred because it allows easy comparison of different localization strategies for various environmental scenarios. This paper presents a two-dimensional (2-D) simulation implementation that addresses tradeoffs between computational efficiency and inclusion of realistic hardware parameters. The 2-D simulation assumes that the plume tracing occurs at a near-constant altitude that is only a few meters above the ocean floor or ground level. The 2-D algorithms presented can be extended directly to three-dimensional (3-D) problems, but implementation for three dimensions requires a significant increasing in computation capacity.

2. Modified Particle Swarm Optimization Framework

Many complex real-world optimization problems are dynamic, and change stochastically over time. These problems require measurements that account for the uncertainty present in the real world. Evolutionary algorithms (EAs), especially Particle Swarm Optimization (PSO), have proven successful in a number of static applications as well as dynamic and stochastic optimization problems. They are particularly successful because they draw their inspiration from the principles of natural evolution, which is a stochastic and dynamic process.

The interaction of the robot with the PSO algorithm is described as follows: Suppose that a population of robots is initialized with certain positions and velocities; let \( \mathbf{x}_i(t) \) and \( \mathbf{v}_i(t) \) denote the position and the velocity vector of the \( i \)-th robot at the iteration time \( t \) \( (t=1,2,\ldots) \). In addition, let \( p_b \) and \( p_g \) be defined as the best local and the best global position found in plume distribution that is under evaluation by the robot at position \( \mathbf{x}_i(t) \). The position and the velocity are updated to improve the fitness function at each time step. When a robot discovers a pattern that is better than any previously found, the positional coordinates are stored in the vector \( p_b \), the best position found by robot \( i \) so far. The difference between \( p_b \) and the current position \( \mathbf{x}_i(t) \) is stochastically appended to the current velocity \( \mathbf{v}_i(t) \). This causes a change to the trajectory the robot would take at that position. The stochastically weighted difference between the population’s best position \( p_g \) and the individual’s current position \( x_i \) is also added to the velocity, in order to adjust for the next time step. These adjustments to the robot behavior direct the search around two best positions.
The value of $p_g$ (the best global position for concentration of the gas) is determined by comparing the best performances of all the members of the population. The performances are defined by indices from each population member; and the best performer’s index is assigned as the variable $g$. Thus, $p_g$ represents the best position found by all members of the population.

Each robot is equipped with an ad-hoc wireless network and global positioning system (GPS). Through the ad-hoc network, each robot transmits and collects the information about the gas concentration, while the position of the robot is determined by the GPS.

The concept of standard PSO is described in eq. (1) and (2).

$$V_i(t) = \chi(V_i(t-1) + c_1 \text{rand}((p_i(t-1) - x_i(t-1)) + c_2 \text{Rand}((p_g(t-1) - x_i(t-1))))$$

$$x_i(t) = x_i(t-1) + V_i(t)$$

(2)

After finding the two best values, the particle velocity and position is updated with eq.(1) and (2). The functions $\text{Rand}()$ and $\text{rand}()$ are random functions returning a value between (0,1). Coefficient $\chi$ is constriction factor, which is less than 1. The coefficient $c_1$ and $c_2$ are learning parameters, where $c_1 = c_2 = 2$.

The main problem with standard PSO applications in dynamic optimization problems is that the PSO will eventually converge to an optimum; it thereby loses the diversity necessary for efficient exploration of the search space.

Applying Coulomb’s law, a charged swarm robot is introduced in order to maintain diversity of the positional distribution of the robots and to prevent them from being trapped in a local maximum. This enhances adaptability to the changes of the environment. Suppose that robot $i$ can observe the present position of the other robots ($x_p$) and has a constant charge $Q_i$ in order to keep a mutual distance away and maintain positional diversity. Two types of swarm robots are defined: neutral and charged robots. For all neutral robots $Q_i = 0$; hence, no repulsive force is applied to the neutral robots. For charged robots, the mutual repulsive force between robots $i$ and $p$ is defined according to the relative distance, $x_i - x_p$ as follows:

$$a_{ip} = \begin{cases} 
\frac{Q_i Q_p (x_i - x_p)}{r_{core}^2} & |x_i - x_p| < r_{core} \\
\frac{Q_i Q_p}{r_{perc}^2} (x_i - x_p) & r_{core} < |x_i - x_p| < r_{perc} \\
0 & r_{perc} < |x_i - x_p| 
\end{cases}$$

(3)

where, $(i \neq p)$ $r_{core}$ denotes the diameter inside which a constant, strong repulsion force is applied and $r_{perc}$ denotes the recognition range of robot. Hence, if the mutual distance is beyond $r_{perc}$, there exists no repulsion force between the robots.

In the case of $r_{core} \leq r \leq r_{perc}$, the repulsion force is dependent on the mutual distance. Then, taking the summation of the mutual repulsion force, robot $i$ defines collective repulsion force by:

$$a_i(t) = \sum_{p \neq i} a_{ip}$$

(4)

where $N$ is number of the robots. The charged swarm robot is described in equations (5) and (6)

$$V_i(t) = \chi(V_i(t-1) + c_1 \text{rand}((p_i(t-1) - x_i(t-1)) + c_2 \text{Rand}((p_g(t-1) - x_i(t-1)))) + a_i(t)$$

$$x_i(t) = x_i(t-1) + V_i(t)$$

(5)

(6)

where, the first part of eq.(5) is responsible for finding and convergence to the optimal solution, while the second part maintains diversity of the swarm distribution and prevents robots from being trapped in a local maximum. Also, if all robots are set to the neutral, Charged PSO (CPSO) is reduced to the standard PSO, as described in eq. (1) and (2).

In this section, the integration of chemotaxis and anemotaxis properties to the PSO is introduced. Again, chemotaxis causes the Modified PSO robots to follow a local gradient of the chemical concentration, while an anemotaxis-driven PSO measures the direction of the fluid’s velocity and navigates “upstream” in the plume to find the odor source. This methodology is well known as odor-gated rheotaxis (OGR) since it is employed by animals to find food.

2.1. Conceptual Idea

As explained in Eq. (1) and (2) earlier, unless the position and velocity are updated in the PSO algorithm, there is no guarantee the robot direction will follow the plume upstream to the source. To combat this issue we utilized wind
information. Assume the velocity from the basic PSO becomes an intermediate velocity \( V^*(t) \) from which the robots can know the direction of the wind \( W(t) \) at every step in time. The movement of the robot can be controlled by analyzing the angle \( \theta \) between the intermediate velocity vector of the robot and the wind direction vector. Note that the angle is a relative direction, its mean depends on the direction of the wind at this time step. With this concept, the robot movement not only will follow the gradient of the chemical concentration but also will follow the direction “upstream” of the wind. As a more detailed explanation, let us reformulate \( V^*(t) \) and \( W(t) \) as vectors defined as follows:

\[
V^*(t) = v_x \mathbf{e}_x + v_y \mathbf{e}_y
\]

(7)

\[
W(t) = w_x \mathbf{e}_x + w_y \mathbf{e}_y
\]

(8)

The angle of the two vectors \( V^*(t) \) and \( W(t) \) in two-dimensional space becomes an inner product and is defined as:

\[
\theta = \cos^{-1}\left( \frac{V^*(t) \cdot W(t)}{\|V^*(t)\| \|W(t)\|} \right)
\]

(9)

For implementation, we use the controlling parameter \( \chi_\theta \) to decide the velocity of the robot. After getting the intermediate velocity of the robot, \( V^*(t) \), the Wind Utilization (WU) algorithm will calculate the angle \( \theta \) as mentioned in Eq. 9. Then the controlling parameter, \( \chi_\theta \), is calculated. The continuation function for the controlling parameter \( \chi_\theta \) is described as follows:

\[
\chi_\theta(W(t), V^*(t)) = \frac{1}{2}(1 - (W(t), V^*(t)))
\]

(10)

The modified PSO with Wind Utilization (WU) concept is described from eq. (11) to eq. (12):

\[
V_i(t) = \chi_\theta V^*_i(t)
\]

(11)

\[
x_i(t) = x_i(t-1) + V_i(t)
\]

(12)

2.2. Deal with Multiple Source

The limitation of PSO is premature convergence to a local solution or one solution. These situations are also finding in multiple odor source localization problem [12]. To cope this situation, niche method with deflection procedure, is adopted [9-10],[12]. The deflection approach operates in multiple odor density function, adapting it to remove or lose when the one source was found.

3. Implementation Framework

The odor source localization problem in dynamic environments is related to several issues from biology, physical chemistry, engineering and robotics. This paper proposes a comprehensive approach to offer a sound technical basis for odor source localization in a dynamic environment.

3.1. Environment

In this paper, we adopted an extended Advection-Diffusion odor model by Farrell et al. [11] because of its efficiency. It represents time-averaged results for measurement of the actual plume, including chemical diffusion and advective transportation. In addition, the Advection-Diffusion odor model has a key factor to approximate the meandering nature of the plume, in that the model is sinuous.

The Advection-Diffusion model is composed of a large number of advected and dispersed filaments. Given a large number of filaments, the overall instantaneous concentration at \( x_o = (x, y) \) is the sum of the concentrations at that location contributed by each filament:

\[
C(x_o, t_o) = \sum_{i=1}^{M} C_i(x_o, t_o)
\]

(13)

where \( C \) is the concentration of the plume \( \text{(molecules/cm}^3) \), \( t_o \) is the number of iterations, and \( M \) is the number of filaments currently being simulated.

The Advection-Diffusion gas concentration at the location \( x_o \) due to the \( i \)-th filaments is expressed by:
\[ C_i(x_o, t_o) = \frac{q}{\sqrt{8\pi^3}} \exp \left[ \frac{-r_i^2(t_o)}{R_i^2(t_o)} \right] \]  
(14)

\[ r_i(t_o) = |x_o - P_i(t_o)| \]  
(15)

where \( q \) is the amount of odor released, \( R_i \) is the parameter controlling the size of the \( i \)-th filament; and \( P_i \) is changing positions of the \( i \)-th filament. (For further explanation on this model, see [11], section two and three.)

This model generates plumes that meander; in addition, the meander is coherent with the flow fields in the sense that downwind odor distribution from the source is the result of advection by the flow. Therefore, we extend the original equations from [11] to incorporate the obstacles in the environment. As a result, the environment becomes more realistic and complicated.

### 3.2. Robot Behavior

The gas source localization algorithm used in this work can be divided into three subtasks: plume finding, plume traversal and source declaration. Random search is employed until one robot encounters the plume. After finding the plume, the second task of the plume traversal proceeds. Particle Swarm concept will be applied to following the cues determined from the sensed gas distribution toward the source. The last task is the source declaration based on the certainty that the gas source has been found. If a robot senses the gas density that is beyond a certain threshold value, it means that the gas source location is specified; and hence, the searching behavior is terminated. Moreover, the search is terminated if the swarm robots fail to localize the odor source by the maximum iteration time step.

To ensure that the performance of proposed strategies is applicable to the hardware experiments, the simulation must contain the key features of the hardware setup. Firstly, the robot has a maximum velocity at which it can move. Hence, the value of velocity vector can be restricted to the range \([-V_{\text{max}}, V_{\text{max}}]\). In this simulation, the maximum velocity is set to \(0.05 \text{ (m/s)}\), by following definition:

\[ V_j(t) = \min(V_j(t), V_{\text{max}}) \]  
(16)

Secondly, in order to incorporate a collision avoidance mechanism, which is not considered in the standard PSO algorithm, we assume that infrared sensors are equipped on each robot. Then the parameters of sensor noise and threshold value are added to model sensor responses. Assume that iteration time \( t \) of the robot in eq. (1) to (6) and iteration time \( t_o \) in eq. (13) to (15) is different time step resolution. Time correlation between time step \( t \) and time step \( t_o \) is explained as follow: The time scale of \( t \) has higher resolution than that of time step \( t_o \) and count up is represented as:

\[ t_o + 1 = t_o + \Delta t \]  
(17)

\( \Delta t \) is the interval time step \( t_o \) in terms of time step \( t \). Hence; \( t_o \) is represented with \( t \) by:

\[ t_o = \left\lfloor \frac{t}{\Delta t} \right\rfloor \]  
(18)

where \([\cdot]\) is the Gauss’s symbol. The sensor response is defined by:

![Fig. 1. Demonstration of ability of parallel MPSO with closing-spread method solves multiple odor sources.](image-url)
\[ S(t) = \begin{cases} C \left( \frac{t}{\Delta t} \right) + e(t) & \text{if } C > \tau \\ 0 & \text{otherwise} \end{cases} \]  

is the sensor’s response, \( C \) is the gas concentration, \( e \) is the random sensor.

**Table 1 Analysis for parameter relationship**

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>df</th>
<th>Mean Square</th>
<th>( F_{calc} )</th>
<th>( F_{table} )</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corrected Model</td>
<td>64804116.958(a)</td>
<td>191</td>
<td>339288.570</td>
<td>100.521</td>
<td>1.00</td>
<td>.000</td>
</tr>
<tr>
<td>Intercept</td>
<td>161867221.542</td>
<td>1</td>
<td>161867221.542</td>
<td>47956.378</td>
<td>3.84</td>
<td>.000</td>
</tr>
<tr>
<td>Area</td>
<td>21350598.277</td>
<td>3</td>
<td>7116866.092</td>
<td>2108.513</td>
<td>2.60</td>
<td>.000</td>
</tr>
<tr>
<td>Source</td>
<td>24577352.973</td>
<td>5</td>
<td>4915470.595</td>
<td>1456.306</td>
<td>2.21</td>
<td>.000</td>
</tr>
<tr>
<td>Robot (R)</td>
<td>2346056.539</td>
<td>2</td>
<td>1173028.269</td>
<td>347.533</td>
<td>3.00</td>
<td>.000</td>
</tr>
<tr>
<td>Niche (N)</td>
<td>6782592.603</td>
<td>2</td>
<td>3391296.302</td>
<td>1004.739</td>
<td>3.00</td>
<td>.000</td>
</tr>
<tr>
<td>Area * Source</td>
<td>4987334.722</td>
<td>15</td>
<td>332488.981</td>
<td>98.506</td>
<td>1.67</td>
<td>.000</td>
</tr>
<tr>
<td>Area * Robot</td>
<td>508385.225</td>
<td>6</td>
<td>84730.871</td>
<td>25.103</td>
<td>2.10</td>
<td>.000</td>
</tr>
<tr>
<td>Source * Robot</td>
<td>817757.164</td>
<td>8</td>
<td>102219.645</td>
<td>40.458</td>
<td>2.10</td>
<td>.000</td>
</tr>
<tr>
<td>Area * Source * Robot</td>
<td>795158.328</td>
<td>24</td>
<td>33131.597</td>
<td>45.973</td>
<td>45.973</td>
<td>.000</td>
</tr>
<tr>
<td>Area * Niche</td>
<td>931027.584</td>
<td>6</td>
<td>155171.264</td>
<td>45.973</td>
<td>2.10</td>
<td>.000</td>
</tr>
<tr>
<td>Source * Niche</td>
<td>1365569.769</td>
<td>10</td>
<td>136556.977</td>
<td>40.458</td>
<td>40.458</td>
<td>.000</td>
</tr>
<tr>
<td>Area * Source * Niche</td>
<td>705616.653</td>
<td>30</td>
<td>23520.555</td>
<td>9.816</td>
<td>9.816</td>
<td>.000</td>
</tr>
<tr>
<td>Robot * Niche</td>
<td>17432.426</td>
<td>4</td>
<td>4358.106</td>
<td>1.291</td>
<td>1.291</td>
<td>.271</td>
</tr>
<tr>
<td>Area * Robot * Niche</td>
<td>312197.735</td>
<td>12</td>
<td>26016.478</td>
<td>7.708</td>
<td>7.708</td>
<td>.000</td>
</tr>
<tr>
<td>Source * Robot * Niche</td>
<td>297325.213</td>
<td>16</td>
<td>18582.826</td>
<td>6.506</td>
<td>6.506</td>
<td>.000</td>
</tr>
<tr>
<td>Area * Source * Robot * Niche</td>
<td>703915.470</td>
<td>48</td>
<td>14777.406</td>
<td>4.378</td>
<td>4.378</td>
<td>.000</td>
</tr>
<tr>
<td>Error</td>
<td>583252.500</td>
<td>1728</td>
<td>3375.301</td>
<td>3375.301</td>
<td>3375.301</td>
<td>.000</td>
</tr>
<tr>
<td>Total</td>
<td>232503859.000</td>
<td>1920</td>
<td>12075.37</td>
<td>12075.37</td>
<td>12075.37</td>
<td>.000</td>
</tr>
<tr>
<td>Corrected Total</td>
<td>70636637.458</td>
<td>1919</td>
<td>3720.738</td>
<td>3720.738</td>
<td>3720.738</td>
<td>.000</td>
</tr>
</tbody>
</table>

\( a \) R Squared = .917 (Adjusted R Squared = .908)

4. **Experimental Example**

The main idea of PSO niching is to eliminate found solutions [8]. In multiple odor sources localization problem, one simple way to eliminate found solutions is closing found odor sources. When found odor source is closed, it will not spread any plume. Then, the searching agents can find other odor sources. Moreover spreading method was introduced [12], to
improve the performance of the system.

While in this paper to make searching time faster, we try using parallel PSO niching. Robots are grouped and the number of groups can be determined also the member of groups can also be determined as well. For example we can determine three neutral robots and three charged robots for each group. If we determine two groups, then there is total twelve robot used for multiple odor source localization.

Each group runs by itself. There is no connection between groups. Members of each group can only send and take information among their group. Each group has its global best information which is different and not connected to others. Detect and response mechanism is also run separately among each group. When one group is running in spread phase, other may run in PSO phase.

Conceptual idea of parallel search is showed in Fig 1. Parallel search logically makes searching time faster. Several groups of robot run and find odor sources separately. The comparison between single sub group and parallel are showed in Fig 2. Then the statistical analysis is showed in Table 1 and Table 2.

Table 2 Analysis for searching best combination (performance vs cost)

<table>
<thead>
<tr>
<th>Sources</th>
<th>Range of Searching Area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5 x 5 m²</td>
</tr>
<tr>
<td>5</td>
<td>N₃R₁(62,6)</td>
</tr>
<tr>
<td>10</td>
<td>N₃R₁(198)</td>
</tr>
<tr>
<td>15</td>
<td>N₃R₁(139,6)</td>
</tr>
<tr>
<td>20</td>
<td>N₃R₁(165,8)</td>
</tr>
<tr>
<td>25</td>
<td>N₃R₁(212,8)</td>
</tr>
<tr>
<td>30</td>
<td>N₃R₁(210,7)</td>
</tr>
</tbody>
</table>

5. Conclusion

The Niche MPSO was implemented for solving multiple odor source localization problems. These proposed approaches can solve such dynamic environment problems but in practical, for real natural environment, the robot will find various situations related with multi study from biology, physical chemistry, engineering and robotic. For example in parallel process Robot groups run separately and do not transfer any data to each other. It makes one group cannot find the others. Hence, one group will not know if it goes to the same odor source as the others. These groups compete with each other. It is very inefficient. One odor source can be tracked by more than one group. However, each group should track different odor sources. Unresolved problem still find in implementation phase. Most of those could be grouped into one of the following categories: Environments, Performance Analysis, Algorithm Optimization and Real Hardware Implementation. We also try to analyze the feasibility conjectures referred to above, in future work.

Acknowledgment

This work was supported in part by the Incentive Research Program 2008 by the Ministry of Research and Technology Republic of Indonesia, and Science and Technology 2007 by INDONESIA TORAY SCIENCE FOUNDATION 2007. The authors also are grateful to Prof. J. A. Farrell from the University of California, Riverside, U.S.A., for his support in advanced turbulent-environment source-code.

References