The Preliminary Study on Multi-Swarm Sharing Particle Swarm Optimization
Applied to UAV Path Planning Problem

Chih-Li Huo, Tzu-Ying Lai
Department of Electrical Engineering
National Dong Hwa University
Hualien, Taiwan
{d9823004, u9723029}@ems.ndhu.edu.tw

Tsung-Ying Sun, Member, IEEE
Department of Electrical Engineering
National Dong Hwa University
Hualien, Taiwan
sunty@mail.ndhu.edu.tw

Abstract—This paper presents a preliminary study on multi-swarm sharing scenario for particle swarm optimization, MSSPSO, to deal with uncertain-dimension factor space optimization problems. The proposed MSSPSO can provide more wide capability for unknown solution space exploration. In this paper, the MSSPSO is applied to UAV path planning problem. Based on characteristic number of different paths, it has to use different number of control point to produce varied flight paths. In order to explore suitable solution within suitable characteristic number interval, MSSPSO is employed to explore better solution and the variable-length crossover concept is used to share information among different dimension swarms.

The simulation is show that MSSPSO has the ability to explore suitable solution and determine suitable characteristic for flight path. On the other hand, swarm crossover helps swarm to avoid falling local optimal position; swarm manager is applied to enhance computing efficiency and prune the helpless swarms.

Keywords— swarm intelligent; variable-length crossover; path planning;

I. INTRODUCTION

In recent years, evolutionary computation (EC) has been used as a popular and powerful optimization technique for complex solution space searching in many applications. EC is a population-based stochastic technique for emulation the process of natural selection in a search procedure. The more fit individuals have the opportunity to mate most of the time, leading to the expectation that the offspring have a similar, or better fitness to improve the ability of individuals to survive. In general, the complexity of a real-world problem lies in the complexity of solution search space. Since the complexity arises due to size of the problem domain, non-linear interactions between various elements, domain constraints, performance measure with dynamics and many independent and codependent elements, and incomplete, uncertain, and imprecise information, etc. Therefore, while the dimension of factor space is clearly known based on a priori model, a typical EC could obtain optimal solution in such fixed dimension problems. That is to say, it is significant issue that the factor space is clear enough to model such problem or not. Example of blind signal separation (BSS) problem, the sources cannot be separated by neural network unless the sources number (variable factor) is identified, and there are usually used the mapping method based on principal component analysis (PCA) [1]. Further, the characteristic based problems are also usually used such method to extract suitable characteristic number, or called principal component. In short, those unknown variables problem use these preprocess, such as mapping method, extraction or transform to determine suitable fixed variables that are used to solve following problem.

The variable-length crossover is developed to solve this problem that has ability to determine suitable dimension. It is useful to search the numbers of variable by designed crossover that main principle is use two different crossover point to produce diverse dimension offspring [2]. The structure of neural network that increase and decrease by variable crossover is applied successfully to satisfy its requirement as time [3]. However, variable-length crossover operator is less impotent to explore solution space in detail because it only combines offspring with varied dimension without has ability of learning swarm intelligence that guide swarm to arrive better position.

For path planning problem, the skeletonization, preprocess is used to extract suitable characteristic, i.e. control points for B-spline curve in our previous research [4]. It is powerful to plan suitable path but has some defects. The first is that skeletonization has to predefine several parameters that affect the performance of skeletonization. Then, the whole terrain has to analysis in detail that it is waste computing resources although it can get enough characteristic to model the suitable path in given terrain. Another defect is that the characteristic is enough but not suitable, it usually uses too many characteristic to model flight path lead the path to be complicated. Therefore, this paper addresses a preliminary study on multi-swarm sharing scenario for particle swarm optimization (MSSPSO) for dealing with uncertain-dimension factor space optimization problems. The proposed MSSPSO can provide more wide capability on searching suitable dimension of solution space, i.e. control point of in UAV path plan, and explore better position using swarm intelligence simultaneously.

The remainder of this paper is organized as follows: Section II presents the multi-swarm sharing PSO, including swarm crossover and swarm manager. Section III describes the production of flight path and related restriction of path production of flight path and related restriction of path...
planning for UAV. The experiment results of path planning are
demonstrated and compared with standard PSO in Section IV. The
last section, Section V contains a brief conclusion.

II. MULTI-SWARM SHARING PARTICLE SWARM
OPTIMIZATION

This section introduces MSSPSO in detail with three parts.
The scenario of particle swarm optimization with multi-swarm
introduces first. Next, the swarm crossover is submitted to
search the solution in other dimension and solve the problem
that falling local optimal solution. And then, the swarm
manager is employed to manage each swarm in whole
population in order to get lower computing cost. At last in this
section, the process of proposed algorithm is described in detail.

A. Particle Swarm Optimization with Multi-Swarm

The PSO is a population based optimization technique
proposed by Kennedy and Eberhart [5] in 1995, it has also
been used to solve various engineering problems [6]. The
population is referred to as a swarm. The particles examine
fast convergence to local optima and/ or global optimal position(s)
over a small number of generations.

A swarm in PSO consists of a number of particles. Each particle
represents a potential solution to the optimization task.
All of the particles iteratively discover the probable solution.
Each particle moves to a new position according to the new
velocity which includes its previous velocity, and the moving
vectors according to the past best and global best solution. The best solution is then kept; each particle
accelerates in the directions of not only the local best solution
but also the global best position. If a particle discovers a new
probable solution, other particles will move closer to it in
order to explore the region with more depth [7].

Let $sz$ denote the swarm size. In general, there are three
attributes, the particles’ current position $p_i$, current velocity $v_i$
and local best position $Pb_i$ for particles in the search space to
present their features. Each particle in the swarm is iteratively
updated according to the aforementioned attributes. Assuming
that the cost function $J$ is to be minimized so that the particles
contain $N$ dimensions, the new velocity of every particle is
updated by

$$v_i(g+1)=v_i(g)+c_1 \cdot r_{i,j}(g)\left[Pb_i(g)-p_{i,j}(g)\right]$$
$$+c_2 \cdot r_{i,j}(g)\left[Gb_i(g)-p_{i,j}(g)\right]$$

For all $j \in 1...N$, $v_{i,j}$ is the velocity of the $j$th dimension of the
ith particle, the $c_1$ and $c_2$ denote the acceleration coefficients,
$r_1$ and $r_2$ are elements from two uniform random sequences in
the range (0, 1), and $g$ is the number of generations. The new position of a particle is calculated as follows:

$$p_{i,j}(g+1)=p_{i,j}(g)+v_{i,j}(g+1)$$

The local best position of each particle is updated by

$$Pb_i(g+1)=\begin{cases} Pb_i(g), & \text{if } J(Pb_i(g+1)) \geq J(Pb_i(g)) \\ Pb_i(g+1), & \text{otherwise} \end{cases}$$

and the best position $Gb$ found from all particles in its search
dimension during the previous three steps is defined as:

$$Gb(g+1)=\arg \min_{Pb_i} J(Pb_i(g+1)), 1 \leq i \leq sz$$

In (3) and (4), the cost function $J$ of path planning problem is
defined as (16) and (17).
Now, consider $R$ particle swarms that have different
dimension $N$ in solution space, it is described as follows:

$$N=[N^1, N^2, ..., N^R], \quad M=1, 2, ..., R$$

where $M$ is $Mth$ swarm index, and the equation (2) is redefined
as follows:

$$v_{i,j}^{M}(g+1)=v_{i,j}^{M}(g)+c_1 \cdot r_{i,j}^{M}(g)\left[Pb_i^{M}(g)-p_{i,j}^{M}(g)\right]$$
$$+c_2 \cdot r_{i,j}^{M}(g)\left[Gb_i^{M}(g)-p_{i,j}^{M}(g)\right]$$

and the new position of a particle in $Mth$ swarm is redefined as:

$$p_{i,j}^{M}(g+1)=p_{i,j}^{M}(g)+v_{i,j}^{M}(g+1)$$

Then, the local best position and global best position in $Mth$
swarm are updated respectively as follows:

$$Pb_i^{M}(g+1)=\begin{cases} Pb_i^{M}(g), & \text{if } J(Pb_i^{M}(g+1)) \geq J(Pb_i^{M}(g)) \\ Pb_i^{M}(g+1), & \text{otherwise} \end{cases}$$

$$Gb_i^{M}(g+1)=\arg \min_{Pb_i} J(Pb_i^{M}(g+1)), 1 \leq i \leq sz^M$$

where $sz^M$ represent each swarm that has different particle
number, and it has $M$ best positions in different dimensions in
the proposed algorithm, the global best position $G_i$ is selected
among $Gb_i^{M}$:

$$G_i(g+1)=\arg \max_{Pb_i} J(Gb_i^{M}(g+1))$$

the minimum dimension of $G_i$ is selected if global best positions $G_i$ that have same fitness value are more than two,
because it has the lowest complexity in those best positions, i.e.
flight paths.

Furthermore, because each particle movement is carried out
according to the past best and global best solution, it will result
in the inevitable confinement of particles in the local optimal
solution [8]. The swarm crossover could help those particles
avoid the event where a particle would fall into the local
optimum.

B. Swarm Crossover

In traditional Genetic Algorithm (GA), the crossover is
applied to generate the offspring by cut and splice process.
The selected crossover point is used to “combine” offspring from
two selected parents that change their information behind
crossover point, i.e. the dimension of offspring is fixed in
assigned dimension. However, the suitable characteristic
number is often unknown lead it to define the dimension of
search space hardly. The variable-length crossover provides a
possibility that discovery other dimensions.

The major capabilities of the proposed swarm crossover is
to explore other dimension randomly, it is useful to search
suitable dimension about flight path characteristic by m-GA’s
variable-length crossover concept. Then, the swarm crossover
is described as follows:

First, two parents are selected from the global best positions $Gb_i^{M}, M=1, 2, ..., R$. Then, two crossover points are
chosen randomly from selected parents, their information that
is behind selected crossover points are changed and the offspring that extend or shrink search dimension are produced. Two offspring are compared by fitness function and the better one is selected to generate new swarm. The new swarm is generated by the same initial process of PSO.

Further, this crossover not only generates new swarm in different dimension, but also has the ability to avoid falling local optimal solution. If this crossover generates an offspring which dimension is the same as existing swarm, it will be a particle and add to this swarm. As an added bonus, this added particle is generated from other swarm’s best position, it will enhance the diversity of this swarm bring the ability that avoid falling local optimal solution.

Unfortunately, swarm crossover only has the ability to increase swarms or particle. So that, it will generate too many swarms lead PSO to have heavy computing cost. In this paper, the swarm manager is employed to overcome this drawback.

**C. Swarm Manager**

The objective of swarm manager is to prune unhelpful swarms. The main concern is which unhelpful swarms are. This is embarrassed to determine swarm’s contribution. In this paper, a particle with worst fitness is deleted in the whole swarm in each generation. It is moderate to prune the unhelpful particle because the worst swarm may has chance of arrive a better position in next generation.

Additionally, the worst swarm is death when the worst particle is deleted lead the swarm to be single member. The death swarm will record the best solution that is different than pruning particle, and this record is considered to be initial $G_{bmax}$ when the new swarm that has the same dimension is produced. This manager is useful to keep the better swarm and prune the unhelpful swarm lead algorithm to enhance computing preference.

**D. Process of planning algorithm**

The complete flowchart of the MSSPSO is shown in Fig. 1. First, two initial swarms are given in order to combine different dimension offspring by swarm crossover. After initial process, the PSO process is used to estimate fitness function and get $G_{bmax}$. The $G_{bmax}$ are select to combine two offspring by swarm crossover. And then, the better offspring is picked to produce a new swarm in different dimension. If the dimension of better offspring is the same with previous swarm, it will be added to this swarm to enhance the diversity of population. Next, the swarm manager is used to prune worst particle and the swarm with single particle, it will save computing cost and avoid falling local optimal position. The last, the $G_{bmax}$ the best solution among those swarms is evaluated. Above process is repeated until terminal condition is met, such as given generation or larger enough fitness.

**III. EXAMPLE OF PATH PLANNING**

In this section, an example for MSSPSO, UAV path planning is introduced, includes flight restriction, it is used to meet UAV flight ability and envelope restriction, and flight path description, that is how to use smaller data to produce complicated 3D flight path.

**A. Flight Restriction**

It is important that the path have to meet some restriction about UAV envelop, such as climbing and gyration abilities. In this paper, the dynamic model of flight vehicle is referred to define some parameters about flight restriction [9].

For above reason, the variation of path angle $\psi_{max}$, climbing head and diving head angle $\gamma_{max}$ that limit the path’s gyration ability, climbing and diving ability are defined respectively. In proposed algorithm, each path is considered those restrictions and the UAV envelop is considered to be a point. The suitable paths is adjusted by the proposed algorithm until meet those restrictions. Two limit parameters of angle $\psi_{max}$ and $\gamma_{max}$ are used to formulate the fitness function in section III.

**B. Flight Path Description**

In order to produce flight path, the B-spline curve is employed to produce flight path. Its advantage is easy and fast to calculate complex curve by several control points and smooth enough to flight in 3D environment [10].

In a 3D environment, the B-spline curve for degree K is defined by $n+1$ control points with coordinates from $(x_0, y_0, z_0)$ to $(x_n, y_n, z_n)$ respectively and blending functions $B_{i,k}(t)$ to smooth the line that connects the control points:

\[
\begin{align*}
X(t) &= \sum_{i=0}^{n} x_i B_{i,k}(t) \\
Y(t) &= \sum_{i=0}^{n} y_i B_{i,k}(t) \\
Z(t) &= \sum_{i=0}^{n} z_i B_{i,k}(t)
\end{align*}
\]
where symbol $t$ is an adjustable parameter set between 0 and $n-K+2$. $B_{i,k}(t)$ are defined in a set of Knot value. $Knot(i)$ is given with (12), where $0 \leq i \leq n+K$.

$$Knot(i) = \begin{cases} 0, & \text{if } i < K \\ i-K+1, & \text{if } K \leq i \leq n \\ n-K+2, & \text{if } n < i \end{cases} \tag{12}$$

where $Knot(i)$ varies between 0 and $n-K+2$, i.e., $0 < K \leq n+1$. Thus, blending functions $B_{i,k}(t)$ are defined recursively, written as:

$$B_{i,k}(t) = \frac{(t-Knot(i))}{Knot(i+K-1)-Knot(i)}B_{i,k-1}(t)$$

$$+ \frac{Knot(i+K)-t}{Knot(i+K)-Knot(i+1)}B_{i+1,k-1}(t) \tag{13}$$

and when $K=1$, the blending function $B_{i,k}(t)$ is defined:

$$B_{i,k}(t) = \begin{cases} 1 & \text{if } Knot(i) \leq t < Knot(i+1) \\ 1 & \text{if } t = Knot(i+1) \text{ and } t = n-K+2 \\ 0 & \text{Otherwise} \end{cases} \tag{14}$$

where integer value $K$ is the degree of B-spline curve, different $K$ values could produce differently fluid curves.

This curve is useful to produce varied path, which adapt any environment, but the suitable control point number $n+1$, can’t determine clearly although the path in complex environment may need larger control points to avoid obstacles; Some paths, however, even don’t need any control points in simple place, such as straight path.

In this paper, the initial particles of PSO are obtained from the control points of the appropriate candidate path. In the proposed method, a matrix $P_{\text{candidate}}$ is used to represent the control points of B-spline curve for the appropriate candidate path:

$$P_{\text{candidate}} = \begin{bmatrix} x_1 & x_2 & \ldots & x_{n+1} \\ y_1 & y_2 & \ldots & y_{n+1} \\ z_1 & z_2 & \ldots & z_{n+1} \end{bmatrix} \tag{15}$$

where size of $P_{\text{candidate}}$ is $3 \times (n+1)$ at initial process, and each column represents a coordinate of control point.

C. fitness function for Path Planning Problem

The fitness function can describe this problem which is to be solved, i.e. path planning problem. That is, fitness represent the merits of solution that using magnitude of values to describe distance of the optimal solution. Larger fitness means approaching the optimal solution with much more probabilities, and it will guide its swarm to arrive better position. However, a difficulty is how to design appropriate fitness function. The haphazard design of fitness function could induce over-complicated or steep solution space cause PSO to falling local optimal solution or bad explore ability.

In this paper, two fitness functions are considered, including restricted function $J_{\text{res}}$ that is used to find path meet flight ability $\dot{\psi}$ and $\dot{\gamma}$, and is feasible to flight; task function $J_{\text{task}}$, the other one is used to assign varied task for UAV. They are defined as follows:

$$J_{\text{res}} = \frac{J_1 + J_2}{2} \tag{16}$$

$$J_{\text{task}} = \frac{w_1J_1 + w_2J_2 + w_3J_3}{w_1 + w_2 + w_3} \tag{17}$$

where $w_1, w_2, w_3$ are the weights of task assignment, and $J_{\text{task}}$, $i=1, 2, \ldots, 5$ are sub function that is explained as follows:

1) $J_1$ – the sub function of collisions with obstacles:

Now assume the flight path is constructed by $M_{bs}$ sampling points $SP$ after B-spline curve process, the collision parts of path are selected from sampling points, and $J_1$ is defined as follows:

$$J_1 = \frac{M_{bs} - M_{bs}^*}{M_{bs}} \tag{18}$$

where $M_{bs}^*$ is the number of collision parts from sampling points. When there is no any discrete sampling points collide with obstacles, $M_{bs}^*$ is zero, and i.e. $J_1$ is 1.

2) $J_2$ – the sub function of path angle limit

Considering the boundary of variation of path angle and heading angle, the $\psi_{\text{max}}$ and $\gamma_{\text{max}}$ are used to estimate flight path angle and heading angle is feasible, as Fig. 2, the feasible path $\psi^*$ and the feasible heading angle $\gamma^*$ are described as follows:

$$\psi^* = \{\psi_i | \psi_i > \psi_{\text{max}}, i = 1, 2, \ldots, M_{bs} - 2\} \tag{19}$$

$$\gamma^* = \{\gamma_i | \gamma_i > \gamma_{\text{max}}, i = 1, 2, \ldots, M_{bs} - 2\} \tag{20}$$

Let $N_{\psi}$ and $N_{\gamma}$ are the number of the feasible path and the feasible angle respectively. Therefore, $J_2$ is defined as follows:

$$J_2 = \frac{N_{\psi} + N_{\gamma}}{2(M_{bs} - 2)} \tag{21}$$

when the all angle of path are met $\psi_{\text{max}}$ and $\gamma_{\text{max}}$, $J_2$ is 1.

Above function $J_1$ and $J_2$ are bases of generating a feasible path to flight. The following explain the function about tasks, which make the PSO adjust according to the assigned task requirement.

3) $J_3$ – the sub function of path length

Assume the sampling point is enough to fit curve, then the
approximate length $L$ can be calculated by

$$L = \sum_{t=1}^{M-1} \sqrt{\dot{x}^2(t+1) + \dot{y}^2(t+1) + \dot{z}^2(t+1)}$$  \hspace{1cm} (22)$$

where $\dot{x}$, $\dot{y}$ and $\dot{z}$ is described as follows:

$$\begin{align*}
\dot{x}(t+1) &= x(t+1) - x(t) \\
\dot{y}(t+1) &= y(t+1) - y(t) , t = 1,2,...,M_{bc} - 1 \\
\dot{z}(t+1) &= z(t+1) - z(t)
\end{align*}$$  \hspace{1cm} (23)$$

However, the length is varied with flight behavior, it is inappropriate to be fitness function because it has no boundary lead the proportion of each sub function to be unbalanced. For this reason, $J_3$ is normalized as follows:

$$J_3 = \frac{L_{\text{min}}}{L}$$  \hspace{1cm} (24)$$

where $L_{\text{min}}$ is the shortest path between start point and endpoint. If the path is shortest, then $J_3$ is 1.

4) $J_4$ - the sub function of path variation along X-Y plane

Consider flight path is constructed by $n+1$ control points, and the path variation vector $v_{CP}$ is calculated by difference of control point as follows:

$$v_{CP}(i) = CP_{i+1} - CP_i , i = 0,1,\ldots,n$$  \hspace{1cm} (25)$$

The angle of path variation vector is calculated by inner product of vectors, as follows:

$$\theta(i) = \cos^{-1}\left(\frac{v_{CP}(i+1) \cdot v_{CP}(i)}{|v_{CP}(i+1)||v_{CP}(i)|}\right) , i = 0,1,\ldots,n-1$$  \hspace{1cm} (26)$$

The larger $\theta$ is, the larger variation of path is. For example in Fig. 3, pathA has smaller $\theta$ bring smooth path; on the other hand, the control point is linked with zigzagged connection, the $\theta$ could be larger than pathA. In fact, the larger $\theta$ represent the path maybe need heavy turn to avoid obstacles.

Further, $\theta$ is varied between 0 to $\pi$, it has to normalize and $J_4$ is defined as follows:

$$J_4 = 1 - \frac{\sum_{i=0}^{n-1} \theta_i}{\pi(n-1)}$$  \hspace{1cm} (27)$$

when path is perfectly straight, $\theta$ is zero and $J_4$ is 1. It is shown that security-oriented path reflects larger $J_4$ with less control points, as Fig. 3.

5) $J_5$ - the sub function of path variation along Z-axis

This sub function of path variation $J_5$ can show the behavior of UAV climbing or diving frequently or not. Let the $z$ coordinate of control points $CP_j$ denote as $z_j$. $z_j$ is normalized by the boundary between maximum and minimum of flight height, sign as $\dot{z}_j$, that is boundary between 0 and 1. Then, the variance of $\dot{z}_j$ is formulated as follows:

$$\sigma_j = \frac{1}{n+1} \sum_{j=1}^{n+1} (\dot{z}_j - \bar{z})^2$$  \hspace{1cm} (28)$$

where $\bar{z}$ is the mean of $\dot{z}_j$. Consequently, the sub function $J_5$ is defined as follows:

$$J_5 = 1 - \sigma_j$$  \hspace{1cm} (29)$$

The smaller variance $\sigma_j$ represents UAV is flight stably along the same level.

Above five sub functions ($J_1$ to $J_5$) are designed for flight restrictions ($J_1$ and $J_2$) and task requirement ($J_3$ to $J_5$) respectively. However, until restriction function $J_{\text{res}}$ has to be 1 bring feasible flight path, the task function $J_{\text{task}}$ is considered subsequently in PSO to optimize path to meet task requirement.

IV. SIMULATION

The simulations compare standard PSO with two task assignments in complicated environments, including the shortest path and the security-oriented path. Restrictions of UAV are set as follows: $\psi_{\text{max}}$ is 0.5 rad. and $\gamma_{\text{max}}$ is 0.3 rad., the boundary of flight height from 1.5 to 3.8 km.

1) Comparison with the shortest path

The parameters of MSSPSO are set as follows: The initial swarm size for each swarm is 15, the dimensions of swarm are 3 and 5 that represent the path is produced by 3 and 5 free moving control points without consider start point, direction point and end point. The initial swarm numbers are 2. The generation of MSSPSO is 200, $c_1$ and $c_2$ of MSSPSO are set both 0.09. The parameters of standard PSO are set the same MSSPSO except the swarm size is 150 that is the max total swarm size of MSSPSO, and the dimension of PSO is set 2 that is the best dimension by MSSPSO. All of simulation in this part is considered 50 independent runs to prove its performance.

In part 1, the start point set is (0, 0, 2), the direction point set is (10, 0, 2) that represent UAV flight toward east at the beginning, the endpoint is set (100, 100, 2). The weight of $J_{\text{task}}$ is set as follows: $w_1$ is 1, $w_4$ and $w_3$ are 0.5, this setting lead planning algorithm to search solution with high priority of shortest path.
The result is shown as Fig. 4. All paths are feasible flight in 50 runs. The most suitable flight path is shown by red color and the black paths are the results in other runs. It appears clearly that MSSPSO has better explore ability than PSO. MSSPSO avoid most of local optimal to explore similar paths. For dimension issue in MSSPSO, many suitable paths in 50 runs use 2 free moving control points that include the most suitable path. It also produces fewer path by others control point number, such as 1, 3, 4, 5, 6 and 8 control points, but their position of path is similar lead it to have the similar fitness.

2) Task assignment with the security-oriented path

In part 2, the weight of task function is changed to assign the security-oriented path. \( w_3 \) is set 0 that represent it ignore the factor of path length. \( w_4 \) is set 1 lead algorithm to search the minimum \( \theta \) of path that represent the path is less turn. \( w_5 \) is also set 1 to minimum the variation of \( z \) direction of path. The start point, direction point and end point are set the same with part 1.

The result is shown as Fig. 5, the most suitable path has the red curve and the block curves are results in other runs. It is show that when the weight is changed, the path is focus on security-oriented path. Although they are located in different position, it is security because it avoids fewer obstacles. The mean of used control point is 3.16, the control point number of most suitable path is 6 that is concentrated at nearby corner region. Concentrated control point is help flight path with straight line and few turn to avoid obstacles.

3) Comparison with the ring obstacle terrain

In this part, a difficult terrain with ring obstacle is designed to test the performance of MSSPSO. Because it may need larger number of control point, the initial swarm is set 5 and 7 to employ swarm crossover with longer length. Other parameters set the same as part 1. The result is shown as Fig. 6.

The red curve is the most suitable path during 50 runs, the white paths represent the feasible but not best paths in 50 runs, and the purple curve is bad path that cannot avoid obstacles that occur 7 times. The mean number of control point is 9.82 that represent it need almost 10 control points to model this path. Moreover, the most suitable path uses 9 control points.

The mean of control point for bad path is almost 12 to 14, it is larger dimension for flight path lead MSSPSO to explore heavily. It has to note that the control points of bad path are chaotic in the ring notch because it is not yet completed exploration with larger dimension. Other 43 paths are finish to explore with better swarm crossover. However, two feasible paths at start point flight reversely that ignore initial direction point. They are also ascribed bad path during 50 runs. In short for this part, it is almost 80% to explore suitable flight path in the difficult ring terrain.

4) Discussion

Above simulation results are collated in TABLE I, case A is the result of shortest path, case B is the result of security-oriented path and case C is the result of ring terrain. It is shown that MSSPSO has better performance with larger fitness and smaller standard deviation (STD) than standard PSO in case A. Furthermore, MSSPSO also determines the fewer and suitable control point number to plan path. Case B has the same performance with case A (MSSPSO).

However, because there has some bad path is produced in case C led the mean fitness to be smaller than other cases. On
the other hand, the STD is larger than other cases because those bad path segments inside the ring area are chaos lead the fitness value and control point number to be varied. To sum up, MSSPSO can determine suitable control point number and explore solution space in detail.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean CPN</th>
<th>STD</th>
<th>Mean fitness</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (PSO)</td>
<td>2</td>
<td>0</td>
<td>0.9485</td>
<td>0.0067</td>
</tr>
<tr>
<td>A (MSSPSO)</td>
<td>2.5217</td>
<td>1.2620</td>
<td>0.9718</td>
<td>0.0062</td>
</tr>
<tr>
<td>B</td>
<td>3.0667</td>
<td>1.3551</td>
<td>0.9619</td>
<td>0.0138</td>
</tr>
<tr>
<td>C</td>
<td>8.7750</td>
<td>5.2010</td>
<td>0.8103</td>
<td>0.1954</td>
</tr>
</tbody>
</table>

*CPN is the abbreviation of control point number*

V. CONCLUSION

In this paper, the idea of multi-swarm is employed to solve the characteristic problem with unknown characteristic number. The novel swarm sharing algorithm, MSSPSO not only explore at each swarm in detail, but also share its information to other swarm bring a new offspring that is unusual to add in given swarm and avoid falling local optimal position. The simulation results show that MSSPSO has better performance with less variance than standard PSO, and has the ability to give suitable dimension in the characteristic unknown problem to explore solution in detail.

REFERENCES


