

Coupling Particle Swarm Optimization for Multimodal Electromagnetic Problems

Minh-Trien Pham*, Min Ho Song* and Chang Seop Koh[†]

Abstract - Particle swarm optimization (PSO) algorithm is designed to find a single global optimal point. However, the PSO needs to be modified in order to find multiple optimal points of a multimodal function. These modifications usually divide a swarm of particles into multiple subswarms; in turn, these subswarms try to find their own optimal point, resulting in multiple optimal points. In this work, we present a new PSO algorithm, called coupling PSO to find multiple optimal points of a multimodal function based on coupling particles. In the coupling PSO, each main particle may generate a new particle to form a couple, after which the couple searches its own optimal point using *non-stop-moving* PSO algorithm. We tested the suggested algorithm and other ones, such as clustering PSO and niche PSO, over three analytic functions. The coupling PSO algorithm was also applied to solve a significant benchmark problem, the TEAM workshop problem 22.

Keywords: Particle swarm optimization, Multimodal optimization, Electromagnetic problems

1. Introduction

Stationary electromagnetic design with small design sensitivities is preferred to an optimum design with big design sensitivities because such designs, when exposed to mass production process, may perform poorly due to manufacturing error. Therefore, finding all optimal points of an objective function is necessary so that a designer may select a robust design, which considers constraints and other performances that are not considered in the construction of objective function. There have been many attempts in recent years to improve particle swarm optimization (PSO) so that it may locate not only a global optimal point, but also all possible optimal points including local optimum [1], [2]. Among these, the clustering PSO and niche PSO are popular.

The clustering PSO is based on clustering algorithm, which classifies particles into different groups and allows each group to move not towards a global best point but to a group best point [1]. On the other hand, the niche PSO basically follows the conventional PSO at the beginning of iterations, and then divides the whole group of particles into several groups, before following the idea of clustering PSO [1], [2]. The niche PSO is known to be more effective than the clustering PSO; however, it often fails to find all the optimal points, or it requires a huge number of objective function calculations when applied to engineering optimization problems [3].

In this paper, a robust and efficient PSO algorithm, called coupling PSO, is developed to locate all optimal points of a multimodal objective function with relatively

less number of objective function calculations. The proposed algorithm is applied to some analytic functions to test its effectiveness; it is also applied to TEAM problem 22 [4].

2. PSO Algorithm

2.1 Conventional PSO

PSO is one of the popular algorithms used to find a global optimal point in numerical optimization problems. This algorithm can be easily implemented and has proven to be very effective and quick when applied to many different sets of optimization problems [5].

PSO was originally developed by Kennedy and Eberhart in 1995 [5]. The origin of PSO is population-based search process where individuals referred to as particles are grouped into a swarm. Each particle in a swarm represents a candidate solution to the optimization problem. Each particle moves toward a global optimal point over the multi-dimensional design space, adjusting its position according to its own experience and that of its neighboring particles. Therefore, the particles move towards a global optimal point utilizing the information about the best position encountered by themselves and its neighboring particles. In a d -dimension design space, the i -th particle maintains its current position, $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})$; current velocity, $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{id})$; and its best position (i.e., personal best position) $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{id})$ $i = 1, 2, \dots, N$; where N denotes the total number of particles in a swarm. The best position in the swarm is represented by $\mathbf{g} = (g_1, g_2, \dots, g_d)$, which is called the global best position. If f denotes the objective function to be minimized, then the personal best position of a particle and the global best position at time

[†] Corresponding Author: School of Electrical & Computer Engineering, Chungbuk National University, Korea. (pmtrien@chungbuk.ac.kr)

* School of Electrical & Computer Engineering, Chungbuk National University, Korea.

step t are updated, respectively, as follows:

$$\mathbf{p}_i(t) = \begin{cases} \mathbf{p}_i(t-1) & \text{if } f(\mathbf{x}_i(t)) \geq f(\mathbf{p}_i(t-1)) \\ \mathbf{x}_i(t) & \text{if } f(\mathbf{x}_i(t)) < f(\mathbf{p}_i(t-1)) \end{cases} \quad (1)$$

$$\mathbf{g}(t) = \arg \min_i \{f(\mathbf{p}_1(t)), f(\mathbf{p}_2(t)), \dots, f(\mathbf{p}_N(t))\} \quad (2)$$

$i = 1, \dots, N.$

At each time step t in the simulation, the velocity of the particle is adjusted in the search space as:

$$\mathbf{v}_i(t) = \omega \cdot \mathbf{v}_i(t-1) + \alpha_1 \cdot r_1 \cdot (\mathbf{p}_i(t-1) - \mathbf{x}_i(t-1)) + \alpha_2 \cdot r_2 \cdot (\mathbf{g}(t-1) - \mathbf{x}_i(t-1)) \quad (3)$$

where ω is the inertia weight; α_1 and α_2 are cognitive and social coefficients, respectively, and; r_1 and r_2 are uniform random number within $[0, 1]$.

The current velocity is the sum of three terms: the previous velocity and two terms proportional to the distances to the personal best and the global best positions from the current position (Fig. 1). The velocity of each particle is limited to keep the particle searching in a limited range by defining a maximum velocity V_{\max} as:

$$\mathbf{v}_i \in [-V_{\max}, V_{\max}]. \quad (4)$$

By limiting the range of velocity, each particle will move in limited space. However, the ranges of velocity are controlled so that the whole design space may be covered by all particles. The initial weight was introduced by Shi and Eberhart to control the influence of the velocity vector on searching the optimal point [6]. When $\omega < 0.8$, the PSO algorithm usually finds the optimal point quickly, but in some cases it fails to find a global optimal point. The PSO algorithm always tries to exploit the new area when $\omega > 1.2$. The PSO algorithm, hence, will require more iteration to find a global optimal point. When ω is medium ($0.8 < \omega < 1.2$), the PSO will have the best chance to find the global optimal point; however, this will also take a moderate number of iterations.

The cognitive and social coefficients are discussed in [7].

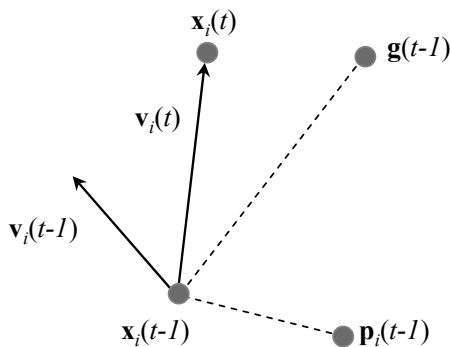


Fig. 1. Updated velocity for the i -th particle

If the cognitive coefficient value is much greater than the social one, the PSO tends toward the personal best position, whereas, it tends toward the global best position if the social coefficient value is much greater than that of cognitive one. A recommended choice for constant α_1 and α_2 is integer 2 since it generally makes the weights for “social” and “cognition” parts to be one [5], [6].

The current position of the particle is computed by adding the current velocity with previous particle position as:

$$\mathbf{x}_i(t) = \mathbf{x}_i(t-1) + \mathbf{v}_i(t) \quad (5)$$

The PSO algorithm repeatedly updates velocity and position until a specified number of maximum iteration has been exceeded, or until the updated velocities are close enough to the global optimal point.

2.2 Neighborhood Structure

As discussed in the previous section, the velocity of a particle is updated by using three terms, namely: its own information described by previous velocity and personal best, and the neighbors’ information represented by the global best position. The global best position of a group is updated by neighborhood topology, which plays a role in regulating the information flow among particles [8]. In this case, it is called *gbest topology*, which is equivalent to a social network where every individual particle is connected to every other one as shown in Fig. 2(a). The *cognition-only* model [7] is at the other extreme. In this case, the particle has no relation with the others.

A more common topology is called *lbest topology*, where each individual particle only interacts locally [9]. The *lbest topology* offers the advantage of allowing subpopulations to search diverse regions of the design space. Some individual particles in one part of the population influences one another to focus on one local optimal point;

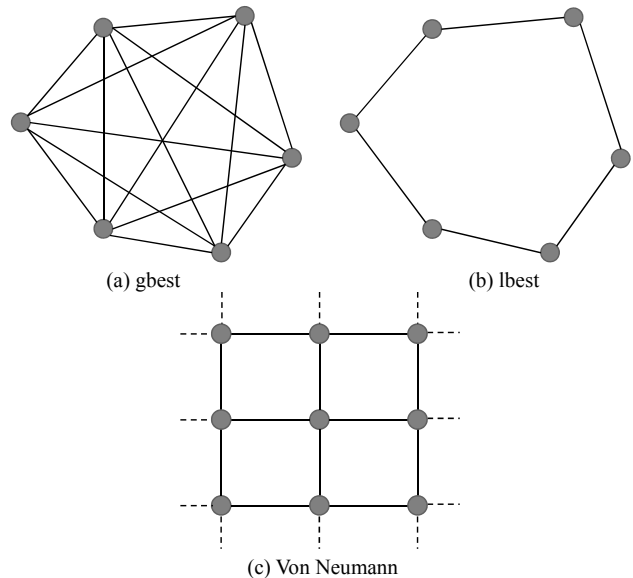


Fig. 2. Swarm of particles with different social networks.

the other part of the population could search around another one. The topology structure is a regular ring lattice with k edges per vertex, so that each particle is influenced by k neighbors. The most common lbest when $k=2$ results in individual particles being affected by only their immediate adjacent neighbors as shown in Fig. 2(b).

In the first systematic study on the performance of different topologies, Kennedy and Mendes recommended the Von Neumann's architecture [8], [10], which makes up a square. The population is arranged in a rectangular matrix, in which each individual particle is connected to the individual particles above, below, and on both sides as shown in Fig. 2(c).

3. Multimodal Function Optimization

A function is multimodal if it has more than one optimal point. There are some problems where multimodal function is optimized using the conventional PSO algorithm for locating a single optimal point. Although the function has one global optimal point, it may have multiple local optimal points. Thus, the conventional PSO may miss locating local optimal point as a solution because this algorithm may not have enough number of particles in swarm or "social" and "cognition" coefficients are not compatible. Furthermore, if the function has multiple global optimal points that have the same objective function value, the conventional PSO may not locate any solution at all. Thus, an optimization algorithm, which can find not only a global optimal point but also all optimal points including local optima, is required.

The two foregoing effective optimization algorithms, i.e., clustering particles swarm optimization and niche particles swarm optimization, are summarized in this section.

3.1 Clustering PSO algorithm

Clustering PSO algorithm classifies the particles into a series of different groups or clusters according to the distance between particles. There are some clustering algorithms that may be classified as exclusive and overlapping clustering algorithm. In the former algorithm, each particle is allowed to belong to only a single cluster, whereas it can belong to several clusters at the same time in the latter. One of the most popular clustering algorithms is k -means clustering, which belongs to the exclusive clustering. Passaro and Starita developed the k -means clustering PSO algorithm using k -mean clustering [1]. In this algorithm, a swarm is divided into clusters or subswarms, and each particle can only communicate with particles in the same subswarm. Therefore, each cluster tends to explore different regions of the search space. The whole process of the algorithm can be described in the proceeding section below.

Step 1. Initial particles

Initialize particles with random positions and velocities.

Step 2. Clustering

In this step, a clustering is performed 10 times, after which the optimum clustering is selected by minimizing the following error E :

$$E = \sum_{j=1}^k \sigma_j \quad (6)$$

where k is the number of clusters, and σ_j is the variance of the j -th cluster defined as:

$$\sigma_j = \frac{1}{N_j - 1} \sum_{i=1}^{N_j} \|\mathbf{x}_i - \mathbf{m}_j\|^2 \quad (7)$$

In the above, N_j is the number of particles belonging to the j -th cluster, and \mathbf{x}_i and \mathbf{m}_j are the k -th particle and the center of the j -th cluster, respectively. The number of particles in a cluster usually depends on the number of clusters. It should be noted that the number of cluster, k , strongly affects the performance of the optimization algorithm because the number of optimal points that can be found by this algorithm is limited by it.

Step 3. Each c steps of iterations

The average number of particles per cluster, N_{avg} , will be calculated from the numbers of particles and clusters. The exceeding particles in a cluster will be removed if the cluster has more particles than the average number of particles per cluster. In this process, the particles with the worst fitness value are removed, and these removed particles are regenerated randomly. The regenerated particles will form a new cluster having the Von Neumann lattice neighborhood. They start performing a more efficient search for new solutions.

Step 4. Perform the conventional PSO for each cluster

When the conventional PSO is applied for a specific cluster, the global best means the best particle of that cluster. Thus, in that cluster, the velocity and position of a particle are updated as follows:

$$\mathbf{v}_i^j(t) = \omega \cdot \mathbf{v}_i^j(t-1) + \alpha_1 \cdot r_1 \cdot (\mathbf{p}_i^j(t-1) - \mathbf{x}_i^j(t-1)) + \alpha_2 \cdot r_2 \cdot (\mathbf{c}^j(t-1) - \mathbf{x}_i^j(t-1)) \quad (8)$$

$$\mathbf{x}_i^j(t) = \mathbf{x}_i^j(t-1) + \mathbf{v}_i^j(t) \quad (9)$$

where \mathbf{c}^j is the best particle in j -th cluster.

Step 5. Stopping criterion

Repeat Steps 3 and 4 until the maximum number of iteration is met.

3.2 Niche PSO algorithm

The first PSO algorithm-based niching technique, called

nbest PSO, was introduced by Brits *et al.* [11]. The nbest PSO was aimed at locating multiple solutions and used lbest topology. The same authors subsequently proposed a new niche PSO approach, which used subswarms to locate multiple optimal points in multimodal function optimization problems [12]. Niches are identified by monitoring the changes in the fitness of individual particles in a swarm. The initial swarm is known as the main swarm. When a particle finds a niche, it forms a subswarm with its closest neighbor, and the subswarm is separated from the main swarm. This process continues until all particles in the main swarm are deleted. If two or more subswarms tend toward the same niche, they will be merged. Moreover, if a particle in the main swarm locates an existing niche found by any other subswarm, it is added to that subswarm. The whole process of the niche PSO can be summarized in the section below.

Step 1. Initialize the main particle swarm

In this algorithm, the positions of initial particles are generated by using Faure-sequences that are distributed with high uniformity within a d -dimensional unit cube [13].

Step 2. Movement of the particles in main swarm

When the particle moves according only to its own history and experience in the form of the personal best, it converges on a local optimal point [14]. Kennedy referred to it as the *cognition-only* model [7]. Thus, the velocity of a main particle is updated using (3), with the social coefficient α_2 being equal to zero.

Step 3. Update fitness of each main swarm particle

Step 4. For each subswarm

(a) Movement of the particles in subswarm

The Guaranteed Convergence PSO is used [15]. Updating the velocity and position of the global best particle are redefined as:

$$\mathbf{v}_\tau(t) = -\mathbf{x}_\tau(t) + \mathbf{g}(t) + \omega \cdot \mathbf{v}_\tau(t-1) + \rho(t)(1-2r_2) \quad (10)$$

$$\mathbf{x}_\tau(t) = \mathbf{g}(t) + \omega \cdot \mathbf{v}_\tau(t-1) + \rho(t)(1-2r_2) \quad (11)$$

where τ is the index of the global best particle, $-\mathbf{x}_\tau$ resets the particle's position to the global best position $\mathbf{g}(t)$, and $\rho(t)(1-2r_2)$ adds a random search. The parameter, $\rho(t)$, is dynamically adapted to control the size of bounding area around $\mathbf{g}(t)$ within which a local search is conducted to force a change in the value of $\mathbf{g}(t)$ in (3).

(b) Update each particle's fitness

(c) Update swarm radius

Step 5. Merging subswarms

While the subswarms perform local searching, some of the subswarms may tend toward the same local optimal point with another subswarm. In this case, these subswarms will merge into one. Formally, any two subswarms

S_l and S_m intersect when

$$\|\mathbf{g}_{S_l} - \mathbf{g}_{S_m}\| < (R_l + R_m) \quad (12)$$

is satisfied, where \mathbf{g}_{S_l} and R_l are the best position and swarm radius of the l -th subswarm, respectively, and \mathbf{g}_{S_m} and R_m are the best position and swarm radius of the m -th subswarm, respectively. With any k -th subswarm, the R_k is defined as:

$$R_k = \max \left\{ \|\mathbf{g}_{S_k} - \mathbf{x}_{S_k,i}\| \right\} \quad (13)$$

where $\mathbf{x}_{S_k,i}$ represents all particles in subswarm S_k subject to $\mathbf{g}_{S_k} \neq \mathbf{x}_{S_k,i}$. The subswarms will be merged when they satisfy:

$$\|\mathbf{g}_{S_l} - \mathbf{g}_{S_m}\| < \mu \quad (14)$$

where μ is a small number, such as $10^{-3}\%$ of design space, ensuring that two subswarms are sufficiently close to each other.

Step 6. Subswarm absorbing particles from the main swarm

When any main particle moves into a subswarm, the particle will be absorbed by that subswarm. That is, the i -th particle is absorbed by a subswarm S_j when it satisfies the following equation:

$$\|\mathbf{x}_i - \mathbf{g}_{S_j}\| \leq R_j \quad (15)$$

where \mathbf{g}_{S_j} and R_j are the best position of the subswarm and the radius of the subswarm S_j , respectively.

Step 7. Creation of a subswarm

If the particle's fitness gives very little change over a small number of iteration, it creates a subswarm together with its closest neighbor particle. The closest neighbor to particle \mathbf{x}_i is simply the particle \mathbf{x}_k defined as:

$$\mathbf{x}_k = \arg \min_{\mathbf{x}_k} \left\{ \|\mathbf{x}_i - \mathbf{x}_k\| \right\} \quad (16)$$

where k is the index of any particle in the main swarm, with $k \neq i$.

Step 8: Stopping criteria

Repeat from Step 2 until stopping criteria are met. If no significant change occurs in the particles' position over a number of iterations or a maximum number of training iterations is met, the algorithm will be stopped.

4. Coupling PSO Algorithm

In coupling PSO, all the main particles at the beginning perform local search to detect a direction, which leads to a local optimal point using the *cognition-only* model [7]. When a particle finds a direction which gives a lower fitness value, it creates a new particle beside the main particle with which to form a couple. A couple with two particles performs the conventional PSO. However, the conventional PSO has an undesirable property, that is, the velocity of particles could easily be zero in its local searching or in its conventional PSO. Thus, we propose the *non-stop-moving* PSO, which will be discussed in the proceeding section. During local searching, some of the couples may tend toward the same optimal point. In this case, only the effective couple survives while the other is eliminated. Finally, each couple will locate its own optimal point. The detail flow of the proposed coupling PSO is summarized below.

Step 1. Initial particles

Initially, N main particles are randomly generated with their position and velocity; these particles are allowed to move according to the *cognition-only* model. Thus, each main particle is expected to move not towards a global optimal point but to its nearest local optimal point.

Step 2. Coupling

When a main particle updates its personal best position, i.e., when $F(\mathbf{x}_i(t)) < F(\mathbf{p}_i(t-1))$ is satisfied, it generates a new particle near itself (Fig. 3) to form the i -th couple together with that particle. The position of the new particle is given randomly as:

$$\mathbf{c}_i(t) = \mathbf{x}_i(t) + r_i \cdot (\mathbf{x}_i(t) - \mathbf{p}_i(t-1)) \quad (17)$$

where r_i is a uniform random number within $[0,1]$.

Step 3. Movement of a couple

The movement of a couple is similar to that of a cluster in the clustering PSO. The two particles in the i -th couple move according to the following rule:

$$\begin{aligned} \mathbf{v}_k(t) = & \omega \cdot \mathbf{v}_k(t-1) + \alpha_1 \cdot r_1 \cdot (\mathbf{p}_k(t-1) - \mathbf{x}_k(t-1)) \\ & + \alpha_2 \cdot r_2 \cdot (\mathbf{c}_i(t-1) - \mathbf{x}_k(t-1)) \\ & + \delta \cdot r_3 \cdot \boldsymbol{\varepsilon}, \quad k = 1, 2. \end{aligned} \quad (18)$$

where $\mathbf{c}_i(t)$ is the couple best position of the i -th couple, r_3 is a uniform random number within $[0,1]$, and $\boldsymbol{\varepsilon}$ is a very small number representing $10^{-3}\%$ of the design space. The last term is for *non-stop-moving* and δ becomes 1 only when the condition, $\mathbf{c}_k(t) = \mathbf{p}_k(t) = \mathbf{x}_k(t)$, $k = 1, 2$, is satisfied for 3 successive iterations; otherwise, δ becomes 0. *Non-stop-moving* PSO is proposed because the conventional PSO algorithm has its own unwanted property when $\mathbf{x}_i = \mathbf{p}_i = \mathbf{g}$ for any i -th particle, the updated velocity in (3) depends only on the $\omega \mathbf{v}_i(t)$ term. As a particle approaches an

optimal point, its velocity approaches zero, and all particles eventually stop moving without locating the exact solution [13], [16], [17]. After the velocity is updated following (18), the position of the particles is then updated by (5).

Step 4. Elimination of a couple and main particle

When a couple or a main particle is very close to another couple, the couple or the main particle is eliminated (Fig. 4). A main particle \mathbf{x}_i will be eliminated if it moves into a couple, i.e., it satisfies the following equation:

$$\|\mathbf{x}_i - \mathbf{c}_k\| \leq R_k \quad (19)$$

where \mathbf{c}_k is the best position of the k -th couple, and R_k is the radius of the k -th couple which is defined as:

$$R_k = \|\mathbf{x}_k - \mathbf{c}_k\| \quad (20)$$

If two couples, \mathbf{c}_k and \mathbf{c}_j , are overlapped, i.e., when the following equation is satisfied:

$$\|\mathbf{c}_k - \mathbf{c}_j\| < (R_k + R_j) \quad (21)$$

the couple which gives lower fitness value is kept, and the other one is eliminated. There is one more circumstance in which the radii become very small when the couples tend toward the optimal point closely. The condition in (21) is not easily satisfied. Therefore, if the distance between two couples is smaller than μ as given by:

$$\|\mathbf{c}_k - \mathbf{c}_j\| < \mu \quad (22)$$

then one of them is eliminated. In this algorithm, μ is set to $10^{-2}\%$ of the design space.

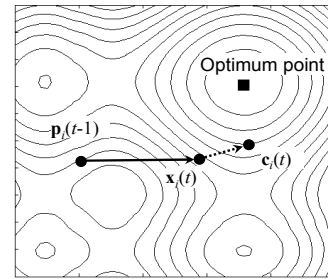


Fig. 3. Generation of a new particle.

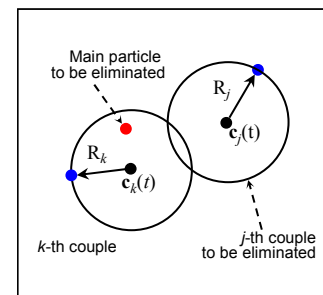


Fig. 4. Elimination.

The elimination process of a couple and main particle is expected to increase the numerical efficiency of the proposed coupling PSO because two close couples locating the same optimal point result in wasted computation time. Thus, eliminating the redundant couple could reduce the evaluation of objective function or computing time.

Step 5. Stopping criterion

The i -th couple stops moving when its movement is very small (less than $10^{-3}\%$ of the design space for 10 successive iterations) and its couple best position is the optimal point of that couple. The algorithm is terminated when all the couples stop moving.

5. Numerical Tests

5.1 Analytical Problems

To check the validity and numerical efficiency of the proposed coupling PSO, the following three analytic functions with two design variables were considered:

$$F_1(\mathbf{x}) = (x_1^2 + x_2 - 11) + (x_1 + x_2^2 - 7)^2 - 200, \mathbf{x} \in [-5, 5]^2 \quad (23)$$

$$F_2(\mathbf{x}) = \sum_{i=1}^2 \{x_i^2 - 10 \cos(2\pi x_i) + 10\}, \mathbf{x} \in [-1.3, 1.3]^2 \quad (24)$$

$$F_3(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^2 x_i^2 - \prod_{i=1}^2 \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \mathbf{x} \in [-9.5, 9.5]^2 \quad (25)$$

The three functions were tested 100 times. In each test, the number of initial particles was set to 50, and the maximum iteration was limited to 800. The performance of three methods (k -means clustering, niche, and coupling PSO) were compared and the results were summarized by their percentages of successful trials of finding all optima and average number of function calls (Table 1). It is clearly shown that the coupling PSO and niche PSO successfully located all optima, while the k -means clustering PSO cannot; results also demonstrated that the coupling PSO requires less number of function calls than the niche PSO. This could be attributed to the possibility that the coupling PSO eliminated the redundant couple and main particles that would locate the same local optimal point. Furthermore, the number of function calls in the proposed coupling PSO was shown to be proportional to the number of optima found. Fig. 5 shows that the coupling PSO gives better convergence speed to optimal points than the others for $F_3(\mathbf{x})$.

5.2 Optimization in Electromagnetic Device

In this work, TEAM problem 22 was chosen to demonstrate the application of coupling PSO in electromagnetic optimization. This problem is caused by the minimization

Table 1. Experiment Results

Function	Optima	% of all optima found			Average function calls	
		Clustering	Niche	Proposed	Niche	Proposed
$F_1(\mathbf{x})$	4	100	100	100	40,000	20,546
$F_2(\mathbf{x})$	9	83	100	100	40,000	26,144
$F_3(\mathbf{x})$	17	14	100	100	40,000	38,352

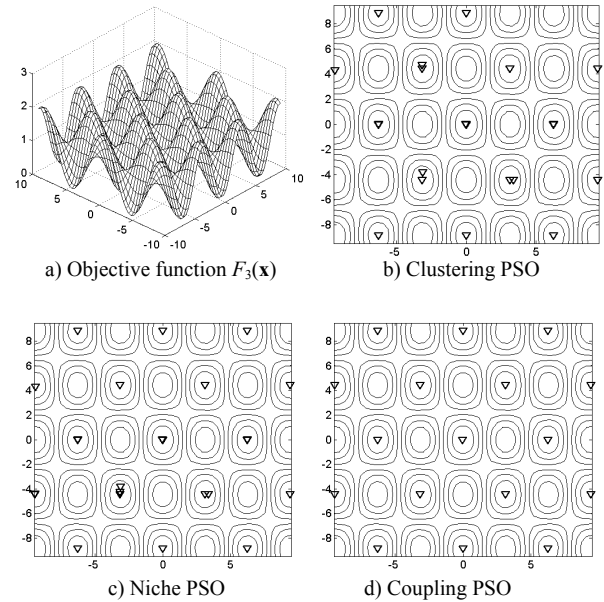


Fig. 5. Particles' position for function $F_3(\mathbf{x})$ after 800 iterations.

of the magnetic flux density at a certain distance from a superconducting magnetic energy storage device. The problem is defined as the minimization of the value of stray field expressed by:

$$B_{\text{stray}}^2 = \frac{1}{22} \sum_{i=1}^{22} B_{\text{stray},i}^2 \quad (26)$$

subject to:

$$B_{\text{max}} \leq 4.92T \quad (27)$$

$$\frac{\text{Energy} - E_{\text{ref}}}{E_{\text{ref}}} \leq 0.05 \quad (28)$$

where the stray field B_{stray} is evaluated at 22 equidistant points along lines a and b (Fig. 6), B_{max} is the maximum magnetic flux density at the outer coil, defined from the critical curve of NbTi superconductor (Fig. 7), and Energy is the stored magnetic energy. In this paper, the deviation for the energy constraint was set at 5% of the reference value $E_{\text{ref}} = 180\text{MJ}$ as represented in (28). The range of the three design variables, which defines the size and position of the outer coil of the device and other parameters, are shown in Table 2.

The coupling PSO was applied to solve this problem with initial 50 particles, maximum iteration of 1,000. There were 12 optima found after 42,000 function calls, 6 of which satisfied both constraints (27) and (28) as shown in Table 3.

Among the candidates obtained by coupling PSO (Table 3), the Opt2 could be selected as the solution because its deviation from the reference value for the *Energy* is smallest and the small B_{stray} is acceptable. However, other candi-

Table 2. Variable ranges and values used

Variable [Unit]	R_1 [m]	$h_1/2$ [m]	d_1 [m]	R_2 [m]	$h_2/2$ [m]	d_2 [m]	J_1 [MA/m ²]	J_2 [MA/m ²]
Min	-	-	-	2.6	0.204	0.1	-	-
Max	-	-	-	4.5	1.1	0.6	-	-
Value	2.0	0.8	0.27	-	-	-	22.5	-22.5

Table 3. Optimization results

Optima	R_2 [m]	$h_2/2$ [m]	d_2 [m]	B_{stray} [mT]	<i>Energy</i> [MJ]	B_{max}	Deviation
Opt1	2.994	0.355	0.280	0.7818	172.01	4.198	0.044
Opt2	3.041	0.255	0.375	0.8998	177.07	4.438	0.016
Opt3	3.131	0.431	0.211	0.9558	176.18	3.326	0.021
Opt4	3.269	0.376	0.222	1.0891	183.99	3.213	0.022
Opt5	3.415	0.458	0.169	1.2234	186.30	2.573	0.035
Opt6	3.557	0.505	0.140	1.3548	187.94	2.099	0.044
TEAM[4]	3.080	0.239	0.394	0.8896	180.03	-	-

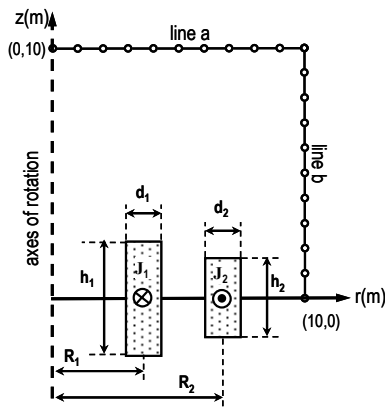


Fig. 6. Design variables.

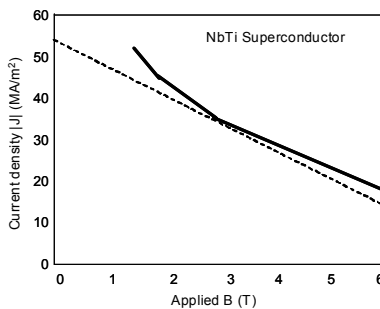


Fig. 7. Critical curve of the material.

dates could be chosen depending on the purpose of the designer.

6. Conclusion

The coupling PSO for multimodal optimization was proposed in this paper. The coupling PSO demonstrated its ability to locate all optima, including the global optimal point. Moreover, coupling PSO required less number of objective function evaluations than *k*-means clustering and niche PSO. The proposed optimization method was applied to solve TEAM 22 problem. After locating multiple optimum solutions by the coupling PSO, the designer may analyze these candidates with respect to other characteristics that have not been considered, such as critical condition or more subjective criteria, during the optimization process. In general, these characteristics are not easily described in mathematical form. These advantages make the proposed coupling PSO algorithm suitable for electromagnetic studies.

Acknowledgements

This work was supported by the research grant provided by the Chungbuk National University in 2008.

References

- [1] A. Passaro and A. Starita, "Clustering particles for multimodal function optimization," *Proceedings of ECAI Workshop on Evolutionary Computation*, Riva del Garda, Italy, August 2006.
- [2] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Scalability of niche PSO," *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 228-234, Indianapolis, IN, USA, April 2003.
- [3] J. H. Seo, C. H. Im, S. Y. Kwak, C. G. Lee, H. K. Jung, "An improved PSO algorithm mimicking territorial dispute between groups for multimodal function optimization problems," *IEEE Trans. on Magnetics*, Vol. 44, No. 6, pp. 1046-1049, 2008.
- [4] B. Brandstaetter, "SMES optimization benchmark, TEAM Problem 22, 3 parameter problem," http://www.igte.tu-graz.ac.at/archive/team_new/team3.php.
- [5] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks (ICNN '95)*, Vol. 4, pp. 1942-1948, IEEE Service Center, Perth, Western Australia, November-December 1995.
- [6] Y. Shi, R.C. Eberhart, "A modified particle swarm optimizer," *Proceedings of the IEEE World Conference on Computational Intelligence*, pp. 69-73, Anchorage, Alaska, May 1998.
- [7] J. Kennedy, "The particle swarm: social adaptation of knowledge," *Proceedings of IEEE Congress on Evo-*

- lutionary Computation (CEC '97), pp. 303-308, Indianapolis, IN, USA, April 1997.
- [8] J. Kennedy and R. Mendes. "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics*, Part C, 36(4): pp. 515-519, 2006.
- [9] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proceedings of the 6th IEEE International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39-43, Nagoya, Japan, October 1995.
- [10] J. Kennedy, R. Mendes, "Population structure and particle swarm performance," *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress*, pp. 1671-1676, May 12-17, 2002.
- [11] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "Solving systems of unconstrained equations using particle swarm optimization," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02)*, Vol. 3, pp. 100-105, Hammamet, Tunisia, October 2002.
- [12] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimizer," *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL '02)*, Vol. 2, pp. 692-696, Singapore, November 2002.
- [13] E. Thie ´ mard, "Economic Generation of Low-Discrepancy Sequences with a b-ary Gray Code," Department of Mathematics, Ecole Polytechnique F ´ ed ´ erale de Lausanne, Lausanne, Switzerland.
- [14] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1931-1938, July 1999.
- [15] F. van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimizer," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02)*, Vol. 3, pp. 96-101, Hammamet, Tunisia, October 2002.
- [16] F. van den Bergh, "An Analysis of Particle Swarm Optimizers." Ph.D. thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.
- [17] F. van den Bergh, A.P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Science*, pp. 937-971, August 2006.



gate model. He can be contacted at pmtrien@chungbuk.ac.kr.



He can be contacted at silver2012@chungbuk.ac.kr, (telephone) +82-43-274-2426, and (fax) +82-43-274-2426.



He can be contacted at kohcs@chungbuk.ac.kr, (telephone) +82-43-261-2426, and (fax) +82-43-274-2426.

Chang Seop Koh received his B.S., M.S., and Ph. D. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 1982, 1986, and 1992, respectively. He became a visiting Professor at the Department of Electrical and Computer Engineering, Florida International University (Florida, USA) and at the Department of Electrical and Computer Engineering, Texas A&M University (Texas, USA) from May 1993 to April 1994, and from February 2003 to January 2004, respectively. He was also a Senior Researcher at the Central Research Institute of Samsung Electro-Mechanics Co., Ltd., from May 1994 to August 1996. Since 1996, he has been a Professor with the School of Electrical and Computer Engineering at Chungbuk National University in Korea. His research interests include electric machine design, numerical analysis of electric machines using the finite element, and boundary element methods.

Minh-Trien Pham received his B.S. and M.S. degrees at the Department of Electronics of Vietnam National University in 2003 and 2007, respectively. He is currently working on his Ph.D. degree on Electric Machines. His research interests include mesh generation, optimization algorithm, and surrogate model.

Min Ho Song obtained his B.S. degree at the Electrical and Electronics Engineering of Chungbuk National University in 2009. He is currently working towards an M.S. degree on Electric Machines. His research interests include vector hysteresis modeling of electrical steel sheets, among others.