Optimizing Laying Hen Diet Using Particle Swarm Optimization with Two Swarms

Gusti Ahmad Fanshuri Alfarisy[1], Wayan Firdaus Mahmudy[1], and Muhammad Halim Natsir[2]
[1] *Faculty of Computer Science, Universitas Brawijaya, Malang, Indonesia.*
[2]*Faculty of Animal Husbandry, Universitas Brawijaya, Malang, Indonesia.*
gusti.alfarisy@gmail.com

***Abstract*—The highest cost production of the poultry industry is the feed that given to the poultry on daily basis. Unfortunately, manual formulation of poultry diet becomes difficult task when several nutritional requirements with fluctuating price are accounted. Several evolutionary approaches have been employed to solve this complex problem such as particle swarm optimization (PSO). However, in order to prevent premature convergence, PSO highly depends on the diversity of particles that influenced by acceleration component. This study presents a strategy to improve diversity in PSO using two swarms with migration and learning phase (PSO-2S). Numerical experimental results show that swarm size of 20 for each swarm, total iteration of migration phase of 42,000, and total iteration of learning phase of 40,000 are the good choice parameter of PSO-2S. While comparison experimental results show that PSO-2S can provide good solutions with the lowest cost and standard deviation than genetic algorithm, canonical PSO, and another migration strategy in multi-swarm PSO.**

***Index Terms*—Feed Formulation; Laying Hens Diets; Least Cost; Multi-Swarm Optimization; Particle Swarm Optimization.**

I. INTRODUCTION

Chicken egg has become worldwide consumption that has many uses as valuable foodstuffs. It uses as staple food and main ingredient for various types of cake and other meals. It contains high-quality protein, a good source of antibodies, and affect the mental development of children with low prices [1]. Thus, it plays an important role in human health [1].

An increasing world population will rise the egg intake and farmer should be able to increase the production to meet these demands. Apart from the aspect of management, cleanliness of cages, temperature, humidity, and others, the fulfillment of nutrients is the pre-requisite to increase egg production. The nutrient deficiency of laying hen may exhibit typical symptoms that make embryos die in the oviduct and may decrease egg production [2]. Thus, feed intake by laying hen is the critical aspect in order to increase its productivity. However, the highest costs production to be incurred is in the feed that takes approximately 65-70%. So that farmer may save massively if the cost of feed formulation is minimized [3].

In order to formulate laying hen diet with least cost, the robust and scientific method is required. Classical methods such as trial and error, pearson square, and algebra have the limited way in formulating the feed mixture that meets the nutritional requirements with least cost. They became complicated and time-consuming when a lot of nutrients and cost of price are accounted [4]. To overcome the drawbacks of classical methods, linear and nonlinear programming is employed to optimize animal diet formulation [5]. Finding the best proportion of ingredients can be used in two different way and nonlinear approach provides the better formulation. It shows us that the relationship between rate of ingredients can be nonlinear. However, both linear and nonlinear programming techniques only have one objective and highly likely the unfeasible formulation is obtained.

In recent years, an evolutionary algorithm is employed to overcome the drawbacks of these methods. A Study conducted by Akif Şahman et al. [6] employed genetic algorithm (GA) to find the feed formulation that satisfies several constraints with objective function to minimize the price of feed formulations. However, the results of their studies, GA encounter difficulty to find the optimum formulation for poultry. Another study by Wijayaningrum [7] employed numerical method to generate initial population for GA and show better results. Random injection technique can also be utilized to repair the negative solution for evolution strategies in Fatyanosa study [8]. To enhance the GA ability in finding global optima, Wijayaningrum propose hybridization approach between adaptive GA and simulated annealing. It can provide a better solution than real-coded GA with little additional computation time [9]. The study conducted by Rahman et al [10] proposed evolutionary model with hard constraint such as a number of ingredients, total weight, and range of protein that must be satisfied exactly. While penalty is given to soft constraint which determined by the expert for several nutrients. However, the expert is needed to formalize the soft constraint. Several animals may require a different kind of nutrient penalty. Their approach also provides high penalty with the unstable formulation.

Particular swarm intelligence approach like particle swarm optimization show promising result for animal diet formulation. It provides robust and better formulation than real-coded GA and linear programming for cattle, sheep and rabbits diet in Altun and Şahman study [11]. Therefore, PSO becomes the best choice to be investigated since it can solve complex constrained programming and complex nonlinear problem in multidimensional space efficiently [12].

However, premature convergence highly likely occurs in solving multi-modal problem when there is loss of diversity [13]. To overcome this issue, multi-swarm PSO can be used to maintain the diversity of the swarm in order to generate better solutions, to consistent and to prevent premature convergence.

Over a decade, many studies are being conducted to improve the performance of PSO by using multi-populations.

The study conducted by He and Wang [14] modify co-evolutionary technique in genetic algorithm and incorporate it into PSO to solve constrained engineering design problem. Two kinds of swarm are defined. One is used for finding a good solution and other used for finding suitable penalty factors. The effective and efficient solution is obtained by using co-evolutionary PSO. Liang and Suganthan [15] introducing multi-population PSO which operates dynamically. Periodically, sub-population filled by other particles that selected randomly and self learning is performed which learn from its own and another personal best position. Lai [16] conduct an experiment with different strategies through various migration strategy between subpopulations. The results of their study, the migration of the gBest particle in source sub-population to the worst particle in destination subpopulation with or without mutations show better results in the complex problem. Influenced by that study, Peng *et al.*, [17] use this migration strategy without mutations with multiple learning strategies. Each sub-swarm have different strategies and combined into one population after the movement process of particles in all sub-swarm is finished 80%.

In this study, we extend the canonical PSO for optimizing laying hen diet using two swarms with learning phase (PSO-2S). The migration strategy that we use is different from the above studies and discussed in the next section. We also investigate the optimum swarm size and the optimum number of iterations.

II. Data and Methodology

A. Data Source

Laying hen diet optimization is a process to determine the best proportion of each ingredient to fulfill the nutritional requirements with the minimum cost. For laying hen diet, 11 different nutrients are accounted. They are crude protein (CP), lysine (Lys), methionine (Met), methionine + cystine (Met+Cys), tryptophan (Tryp), threonine (Thre), crude fat (F), crude fiber (CF), calcium (Ca), total phosphorus (P), and metabolizable energy (ME). The nutritional requirement is vary based on the age of laying hen which is described in detail in Table 1 [18]–[23]. While data on feed ingredients was obtained from Faculty of Animal Husbandry, Universitas Brawijaya, East Java, Indonesia. It consists of the nutrient value and price for each ingredient (see Appendix II for more detail).

B. Canonical Particle Swarm Optimization

Nature inspiration from bird flocking and fish schooling leads to the initiation of particle swarm optimization. The particle flies toward its best personal experience and all particles best experience. It is a simple and effective algorithm to find global optima [24]. The addition of inertia weight is proposed by Shi and Eberhart [25] to control the personal and global best position and it becomes a canonical PSO as we know today.

PSO start with random initialization of N particles with D dimensions which contain velocity and position under a feasible domain. Then particle flies in search space by changing its position based on updated velocity. Both position and velocity are updated by using Equation (1) and (2). Let assume that particle i at particular iteration that contain position (x) and velocity (v) of each dimension denoted as $P_i(t) = \{ (x_{i,1}, v_{i,1}),\ (x_{i,2}, v_{i,2}), \ldots, (x_{i,D}, v_{i,D})\}$. Inertia weight is denoted as w and c_1 and c_2 are acceleration coefficient for cognitive and social component respectively. While r_1 and r_2 are two different random real number between 0 and 1.

$$v_{i,j}(t+1) = w \,.\, v_{i,j}(t) + c_1 \,.\, r_1 \ \left(pbest_i - x_{i,j}(t)\right) + c_2 \,.\, r_2 \ \left(gbest - x_{i,j}(t)\right) \tag{1}$$

$$x_{i,j}(t+1) = x_{i,j}(t) + \ v_{i,j}(t+1) \tag{2}$$

C. PSO with Two Swarms

The proposed PSO-2S is consist of two swarms, let say A and B. Each swarm contains N number of particles. This algorithm has two phases. First is the migration phase, the best particle (*gBest*) of swarm will migrate to other swarm and replace the destinated swarm best particle (*gBest*). If the source swarm is A, then the best particle in its swarm migrate to swarm B and vice versa. After the migration phase is ended, the whole swarm will enter the learning phase. Swarm A will learn from gBest of swarm B by updating the velocity that attracted to gBest B and vice versa. After all phases have ended, the best gBest of A and B will be the final solution.

At a glance, the both phases look similar however they work differently. After migration, both gBest are swapped and all particle will move toward the new gBest. If gBest from swarm A has higher fitness than gBest in swarm B, then all particles in swarm A will fly toward new gBest and will not be attracted anymore to the source gBest. This strategy may increase diversity in swarm A. Since all particles fly through the worse gBest, it has little chance to improve the new gBest. Because the selection of pBest is strict which the value of pBest may have higher fitness than new gBest. This situation may not apply in standard PSO. After the next migration is taking place, the gBest from swarm B will fly to its home, swarm A. Since diversity has been increased, it may lead swarm A to find a better solution. Therefore, the first migration may increase diversity for swarm A and for the next migration, when gBest fly back to its home, they may enhance the original gBest.

If swarm A has worse fitness than swarm B, then all particles in swarm A will fly toward new better gBest and may enhance the new gBest and pBest in swarm A. However, when the gBest back to its home and original gBest from swarm A is not increased, it may not increase the original gBest. This issue is resolved in learning phase.

In the learning phase, the update position is attracted to neighbor *gBest* that is described in Equation (3) and (4). The velocity of particle i in swarm A and B for dimension j respectively denoted as $vA_{i,j}$ and $vB_{i,j}$. While pBest for particle i in swarm A and B denoted as $pbestA_i$ and $pbestB_i$. Global best position for swarm A and B denoted as $gBestA$ and $gBestB$ respectively. By using this strategy, it may helps all particles learn from neighbor global best and may lead to better result.

$$vA_{i,j}(t+1) = w \,.\, vA_{i,j}(t) + c_1 \,.\, r_1 \ \left(pbestA_i - xA_{i,j}(t)\right) + c_2 \,.\, r_2 \ \left(gbestB - xA_{i,j}(t)\right) \tag{3}$$

$$vB_{i,j}(t+1) = w \,.\, vA_{i,j}(t) + c_1 \,.\, r_1 \ \left(pbestB_i - xB_{i,j}(t)\right) + c_2 \,.\, r_2 \ \left(gbestA - xB_{i,j}(t)\right) \tag{4}$$

Table 1
Nutrient requirements for each laying hen stage

No	Nutrient	Unit	Boundary	Layer Pre Starter (1 - 4 Weeks)	Layer Starter (5 - 10 Weeks)	Layer Grower (11 - 16 Weeks)	Pre Layer (17 - 18 Weeks)	Layer (19 - 50 Weeks)	Layer Post Peak (> 50 Weeks)
1	Crude Protein (CP)	%	Min	20.00	19.00	15.50	16.00	16.50	16.00
2	Lysin (Lys)	%	Min	1.00	0.90	0.70	0.75	0.80	0.75
3	Methionine (Met)	%	Min	0.50	0.40	0.30	0.35	0.40	0.35
4	Methionine + Cystine (Met+Cys)	%	Min	0.80	0.70	0.60	0.63	0.67	0.65
5	Tryptophan (Tryp)	%	Min	0.20	0.18	0.17	0.17	0.18	0.17
6	Threonine (Thre)	%	Min	0.75	0.65	0.50	0.52	0.55	0.50
7	Crude Fat (F)	%	Min	3.00	3.00	3.00	3.00	3.00	3.00
8	Crude Fiber (CF)	%	Max	6.00	7.00	8.00	8.00	7.00	8.00
9	Calcium (Ca)	%	Range	0.80 - 1.20	0.80 - 1.20	0.80 - 1.20	2.00 - 2.70	3.25 - 4.25	3.50 - 4.50
10	Total Phosphorus (P)	%	Min	0.60	0.55	0.46	0.50	0.55	0.50
11	Metabolizable Energy (ME)	Kcal/Kg	Min	2900.00	2800.00	2700.00	2700.00	2700.00	2650.00

The step of PSO-2S is shown in the following :

Step 1: Initialize velocity and position randomly for N particles in Swarm A and B in feasible domain.
Step 2: Evaluate all particles using fitness function, initialize pBest and update gBest for both swarms.
Step 3: Update velocity and position by using Equation (1) and (2) respectively.
Step 4: Evaluate particle in swarm A and B.
Step 5: Update personal best if the current particle is better and update gBest if the current particle is better for both swarm.
Step 6: if t mod $migrationPeriod$ is equal to zero, then swap global best particle in both swarms.
Step 7: Repeat step 3-6 if the stop condition is not meet.
Step 8: Update velocity for swarm A and B using Equation (3) and (4) respectively.
Step 9: Update position for both swarms.
Step 10: Update personal best if the current particle is better and update gBest if the current particle is better for both swarm.
Step 11: Repeat step 8-10 if the stop condition is not meet.

D. PSO Application for Laying Hen Diet Formulation

In employing PSO for laying hen diet formulation problem, the main issues are how to encode the particle, how to measure the good particles and what is the good parameters choice to produce optimum formulation. They will be discussed in detail in particle representation, fitness function and good parameters choice.

a. Particle Representation

Each dimension in particle represents the ingredient that being optimized. If a particle has D dimension then it will optimize D number of ingredients. In feed formulation, each ingredient is represented by the percentage and the summation of all ingredients that have to satisfy 100%. For the accuracy, we use real-coded particle. This particle is described in Figure 1. *x* denotes the position of the particle, *i* denotes the particular particle, and *j* denotes the particular dimension. During the movement of particles, the summation of all ingredients may not satisfy 100%, thus Equation (5) is used to adjust the percentage. The negative value may also appear during the movement. This issue is handled in the fitness function.

$Feed_1$	$Feed_2$	...	$Feed_j$	...	$Feed_D$	Total Percentage
$x_{i,1}$	$x_{i,2}$	...	$x_{i,j}$	...	$x_{i,D}$	$\sum_{j=1}^{D} x_{i,j} = 100$

Figure 1: Particle representation

$$x_{i,j} = \frac{x_{i,j}}{\sum_{j=1}^{D} x_{i,j}} \, x \, 100\% \quad (5)$$

b. Fitness Function

In this study, the accounted nutrients are crude protein (CP), lysine (Lys), methionine (Met), methionine + cystine (Met+Cys), tryptophan (Tryp), threonine (Thre), crude fat (F), crude fiber (CF), calcium (Ca), total phosphorus (P), and metabolizable energy (ME). All nutrients use percentage unit except for the ME that use Kcal/Kg. Each age of laying hens has a different nutrient requirement which is shown in detail in Table 1 [18]–[23].

The total nutrients of all ingredients should satisfy the nutritional requirements. The different nutrient may require different nutritional requirements which can be described with different nutritional constraint. The penalty is given when the total nutrient requirements violate the boundary. While the total cost should be minimized during minimizing the penalty. However, the range between the total cost of ingredients and nutrient value may also far which depends on the currency of a particular country. Therefore, The fitness function that has to be maximized can be described as 1 divided by the summation of normalized cost of all ingredients and the summation of the penalty of nutrient constraint which is described in Equation (6).

$$fitness(P_i(t)) = \frac{1}{normalizedCost(P_i(t)) + penalty(P_i(t))} \quad (6)$$

The minimum and maximum cost can be identified through the combination of ingredients. Let say that a lot of ingredients is prepared for laying hen formulation. Then we identify the lowest (*minCost*) and highest (*maxCost*) price among those ingredients in one kg. Since the total rate of each ingredients should be 100%, thus we can say that the minimum and maximum cost of the formulation that can be obtained are 100 multiply by *minCost* and *maxCost* particularly. Thus, Equation (7) is used to normalized the cost.

$$normalizedCost(P_i(t)) = \frac{totalCost(P_i(t)) - 100 \, .minCost}{100 \, .maxCost - 100 \, .minCost} \quad (7)$$

While the total cost is the summation of ingredients rate multiple by the cost of that ingredient which is described in detail in Equation (8) where c_j denotes the cost of particular ingredient *j*.

$$totalCost(P_i(t)) = \sum_{j=1}^{D} x_{i,j} \; x \; c_j \tag{8}$$

Each nutrient has a different constraint to be satisfied. It may have maximum, minimum or range boundary. In order to identify particular nutrient that has a particular nutrient constraint, we add max and min property. Let say a is a particular nutrient like CP or Ca. If a_{max} is equal to zero, then it indicates that it has a minimum boundary. While if a_{min} is equal to zero, it indicates that it has a maximum boundary. If both properties are not equal to zero, than it indicates that it has absolute value in defined range for nutritional requirements. Thus, the penalty for a particle is the summation of penalty from nutritional constraint which is described in detail in Equation (9) where TN_a is the total nutrient of a.

$$penalty(P_i(t)) = \sum_{a=1}^{A} \begin{cases} f_{min}(TN_a(P_i(t))), & a_{max} = 0 \\ f_{max}\left(TN_a(P_i(t))\right), & a_{min} = 0 \\ f_{range}(TN_a(P_i(t))), & otherwise \end{cases} \tag{9}$$

Total nutrient a of formulation is the summation of each proportion of ingredient of nutrient a which is described in Equation (10). Nutritional constraint that has minimum boundary is described in Equation (11). Nutritional constraint that has maximum boundary is described in Equation (12). Nutritional constraint that has range boundary is described in Equation (13).

$$TN_a(P_i(t)) = \sum_{j=1}^{D} \frac{x_{i,j}}{100} \; x \; nutrient_{a,j} \tag{10}$$

$$f_{min}(TN_a(P_i(t))) = \begin{cases} 0, & TN_a(P_i(t)) \geq a_{min} \\ a_{min} - TN_a(P_i(t)), & TN_a(P_i(t)) < a_{min} \end{cases} \tag{11}$$

$$f_{max}(\,TN_a(P_i(t))\,) = \begin{cases} TN_a(P_i(t)) - a_{max}, & TN_a(P_i(t)) > a_{max} \\ 0, & TN_a(P_i(t)) \leq a_{max} \end{cases} \tag{12}$$

$$f_{range}(\,TN_a(P_i(t))) = \begin{cases} a_{min} - TN_a(P_i(t)), & TN_a(P_i(t)) < a_{min} \\ 0, & a_{min} \leq TN_a(P_i(t)) \leq a_{max} \\ TN_a(P_i(t)) - a_{max}, & TN_a(P_i(t)) > a_{max} \end{cases} \tag{13}$$

III. EXPERIMENTAL SETUP AND RESULT

We experiment with swarm size and a number of iteration in order to get the best parameter. We also compare 2SL-PSO with other algorithms. Scala programming language is chosen to develop all algorithms with the same environment to make sure that all algorithms are adequately comparable. All algorithms were run ten times because of stochastic optimization and the average of fitness, penalty, cost and standard deviation was compared. Swarm size and iteration experiment use A11 as test ingredient (please see Appendix I for more detail).

The good choices of inertia weight (w), cognitive coefficient (c_1), and social coefficient (c_2) are 0.6, 1.8, and 2.1 respectively which was used in the all following experiment. These parameters have been tested in our unpublished work.

A. Swarm Size

The swarm size was tuned from 5 until average fitness converges by 5 for both swarms. The number of iteration of 100,000 was set as maximum iteration, migration performed after 5,000 iterations periodically, and 100,000 number of iteration was set for learning phase. The optimum swarm size then was drawn from the experimental result.

The obtained result from swarm size experiment is depicted in Figure 3. The average fitness increased start from 5 to 20 for both swarm and there is no improvement over 20. Therefore, 20 swarm size for each swarm is considered to be the good value for optimum swarm size.

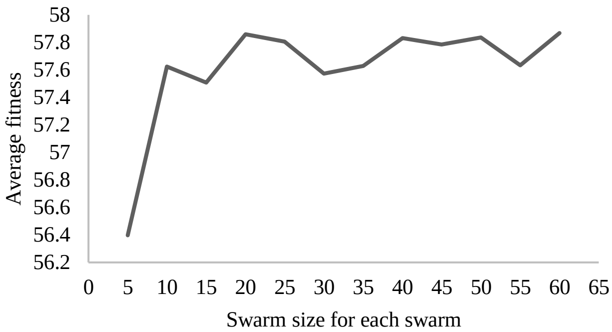

Figure 2: The average fitness for different swarm size

B. Number of Iterations

2SL-PSO use two phases for seeking process. The stopping criteria may independently be determined for both migration and learning phase. The present study uses an iterative method to stop the seeking process of 2SL-PSO. Thus, we tune a number of iteration for both phases and the optimum values were drawn from the experimental result. The optimum swarm size from previous experimentation was used for this experiment.

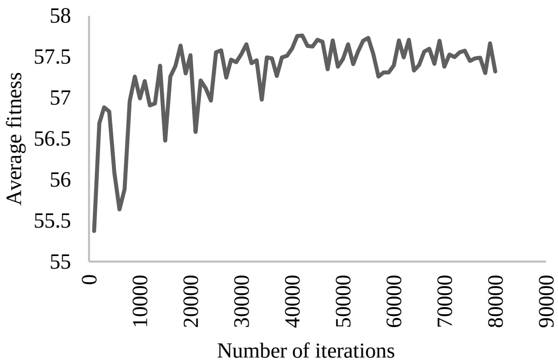

Figure 3: The average fitness for different number of iterations in migration phase

A number of iterations were tuned from 1,000 until the average fitness converges by 1,000 step. The adjustment of this experiment does not involve the learning phase in order to see the optimum value obtained in migration phase. As shown in Figure 3, the average of fitness value was increased from 1,000 to 42,000. The improvement of average fitness is not significant over 42,000. Therefore, 42,000 considered being the good value for total iterations of PSO in migration phase (*maxIteration*).

In order to perceive the effectiveness of learning phase, the different number of iteration should be tuned through iteration parameter. Hence, we introduce a parameter called learning iteration (*learningIteration*) which is similar to PSO iterative method to stop the seeking process but applied in the learning phase. *learningIteration* was tuned from 10,000 until the average fitness converges with 10,000 step. The same swarm was utilized after migration phase. It was run ten times and the average fitness was drawn from the obtained result.

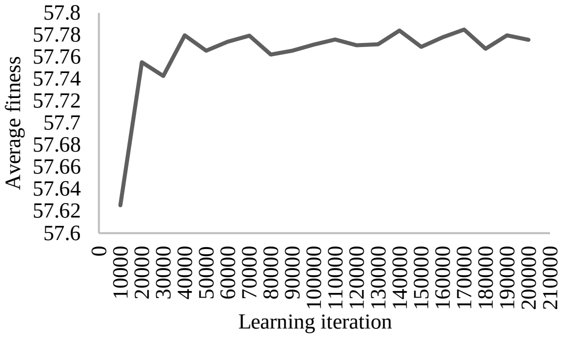

Figure 4: The average fitness of different learning iteration

As shown in Figure 4, the average fitness is sensitive to the learning iteration. The average fitness is gradually increased from 10,000 to 40,000 and does not provide significant result more than 40,000 iterations. Therefore, we choose 40,000 as the good starting value for learning iteration.

C. Comparison

In this study, we compare our proposed algorithm with canonical PSO and real-coded genetic algorithm (GA). We also compare it with multi-swarm PSO with BW strategy using two swarms by Lai [16] (for simplicity we call this algorithm as PSO-BW) in order to see the effectiveness of PSO-2S with another migration strategy. The parameters of each comparison algorithm are set by considering equity comparison. The parameters of each algorithm are described in detail in Table 2.

Table 2
Parameters value of all algorithms

Algorithm	Parameters
2SL-PSO	N = 40 (20 for each swarm), *maxIteration* = 42,000 , $w = 0.6$, $c_1 = 1.8$, $c_2 = 2.1$, *migrationPeriode* = 5,000, *learningIteration* = 40,000
PSO-BW	N = 40, *maxIteration* = 82,000, $w = 0.6$, $c_1 = 1.8$, $c_2 = 2.1$, *migrationPeriod* = 5,000
PSO	N = 40, *maxIteration* = 82,000 , $w = 0.6$, $c_1 = 1.8$, $c_2 = 2.1$
GA	N = 40, $maxIteration = 82{,}000$, $pc = 0.8$, $pm = 0.3$

The comparison results are shown in Table 5. B11 is the abbreviation of the different ingredients combination. It averages that the number 11 denotes the eleven different ingredients that used as test ingredients (please see Appendix I for more detail). As shown in Table 5, PSO-2S can provide the highest average fitness than other comparison algorithms with near zero penalty. PSO-2S also can produce the formula that has lowest average cost and the lowest standard deviation.

As shown in Table 5, the formula obtained by PSO-BW that use the migration strategy of BW in two swarms decrease the average fitness compared to canonical PSO. Replacing the worst particle in destinated swarm with the best particle source swarm may not fully maintain diversity in both swarms which may lead swarm to converge faster. However, 3 of 5 formula (B11, B12, and B15) is more stable than canonical PSO as shown in average standard deviation results. Using more than one swarm may lead particle to move with different direction based on the acceleration component and inertia weight. It may produce different global best position and by using more than one swarm, the probability of stable formulation that obtained by PSO-BW may higher than canonical PSO. These results also show the effectiveness of migration phase and learning phase in PSO-2S to maintain diversity and to make particles in different swarm learn each other which may lead to better solution.

All average fitness of obtained formula from canonical PSO are lower than all average fitness from PSO-2S as shown in Table 3. These average fitness results associate with the average cost which canonical PSO produce higher cost than PSO-2S. The migration strategy by swapping the global best position periodically lead particle to move that influenced by different global best position which may lead the better solution is found. It also show us that migration phase may maintain or increase diversity in PSO with two swarms. Moreover, the formula obtained by PSO-2S also produce more stable than canonical PSO.

Unfortunately, GA produces the lowest average fitness than other comparison algorithms. It is not beneficial for GA since it uses evolution operator like crossover and mutation in evolution process that require more computation time. The evolution operator in GA can not compete with particle movement in PSO. It may lead GA need an improvement by using hybridization or other evolution operator technique to enhance the formula. These results show the effectiveness and efficiency of PSO-2S compared to GA.

Table 3
Comparison of PSO-2S, PSO-BW, PSO, and GA

Formula	Algorithm	Average Fitness	Average Penalty	Average Cost	Average Standard Deviation
B11	PSO-2S	**12.73639142**	1.11E-17	**465,916.73**	**0.000592567**
	PSO-BW	12.71228524	**0**	466,326.62	0.019289156
	PSO	12.72276442	**0**	466,148.98	0.028552415
	GA	12.20509985	**0**	475,602.80	0.452420014
B12	PSO-2S	**20.86232487**	**0**	**362,775.34**	**0.019915684**
	PSO-BW	20.49487767	**0**	365,161.50	0.142064225
	PSO	20.80160571	**0**	363,170.47	0.164848918
	GA	16.65377807	**0**	397,526.26	1.420332937
B13	PSO-2S	**28.24232863**	**0**	**270,830.17**	**0.422434364**
	PSO-BW	27.68565396	**0**	272,292.60	0.791162508
	PSO	27.77125775	**0**	272,060.02	0.71010195
	GA	21.03616908	**0**	295,507.15	1.481661054
B14	PSO-2S	**28.8428542**	**0**	**326,039.47**	**0.131852677**
	PSO-BW	28.65813022	**0**	326,669.97	0.350891842
	PSO	28.66292177	**0**	326,645.19	0.209546701
	GA	26.5589369	**0**	334,531.75	1.285590364
B15	PSO-2S	**35.48962362**	6.66E-11	**317,769.82**	**0.104165946**
	PSO-BW	34.68139583	2.85E-14	319,612.91	0.71610071
	PSO	35.14186506	**5.40E-15**	318,583.50	0.857415918
	GA	31.00731739	2.18E-07	329,215.75	1.560244467

IV. CONCLUSION

This paper presents how two swarms can be used effectively to produce optimum laying hen diet by using migration and learning phase. All formula obtained from all comparison algorithms can produce near zero penalty that shows the quality of the formula. Furthermore, PSO-2S can produce the highest average fitness with the lowest average cost and average standard deviation. It shows that by using PSO-2S can enhance the formula that satisfies the nutritional requirement of laying hen with the lowest cost and stable results. It also shows that the migration strategy used in migration phase in PSO-2S is better than BW strategy that replacing the worst particle with best particle periodically. The obtained results show the effectiveness and robustness of PSO-2S.

In the future study, more than two swarms may be employed with different strategy. By using multiple swarms, the particles in different swarm have different directional movement and may lead to better solution.

APPENDIX I
Test Ingredients

Formula	Ingredients	Total Combination
A11	3, 8, 9, 19, 20, 21, 22 ,23, 24, 26, 27	11
B11	3, 7, 9, 10, 15, 19, 22, 23, 24, 26, 27	11
B12	0, 3, 4, 5, 6, 19, 20, 23, 25, 26, 27, 28	12
B13	1, 2, 3, 4, 5, 8, 17, 18, 21, 25, 26, 27, 28	13
B14	0, 6, 7, 8, 9, 18, 19, 21, 22, 23, 24, 26, 27, 28	14
B15	0, 1, 2, 3, 7, 8, 9, 10, 19, 25, 22, 23, 24, 26, 27	15

APPENDIX II
Ingredient List for Laying Hen Diet

Index	Ingredient	ME	CP	F	CF	Ca	P	Lys	Met	Tryp	Thre	Met +Cys	Price /Kg.
0	Bran	2860	10.2	7	3	0.04	0.16	0.71	0.27	0.09	0.57	0.64	3500
1	Brown Rice	2660	8	1.7	9	0.09	0.04	0.3	0.17	0.1	0.31	0.27	9000
2	White Rice	3100	7.5	0.4	0.4	0.03	0.01	0.27	0.17	0.09	0.36	0.26	10000
3	Fine Bran	1630	8	8	12	0.12	0.21	0.77	0.29	0.1	0.62	0.69	2500
4	Corn Barn	2950	10.6	6	5	0.04	0.15	0.5	0.17	0.27	0.37	0.37	4000
5	Yellow Corn	3370	8.54	2.61	4.76	0.02	0.1	0.2	0.18	0.1	0.4	0.36	5000
6	Pollard	1300	15	4	10	0.14	0.32	0.3	0.17	0.1	0.31	0.27	2300
7	Sorghum	3250	10	2.8	2	0.03	0.1	0.2	0.13	0.12	0.36	0.28	6000
8	Cassava Flour	2970	1.5	0.7	0.9	0.18	0.09	0.03	0.09	0.14	0.18	0.19	2400
9	Whey	1910	13	0.8	0	0.9	0.8	0.9	0.15	0.15	0.7	0.45	8000
10	Cotton Seed Meal	2100	41	4.8	12	0.18	0.33	1.6	0.6	0.5	1.4	1.6	2500
11	Soybean Meal	2240	42	0.9	6	0.29	0.65	2.9	0.65	0.6	1.8	1.32	5900
12	Coconut Meal	2200	18.5	2.5	15	0.2	0.57	0.64	0.29	0.2	0.65	0.59	3500
13	Sesame Meal	1910	45	5	5	2	0.3	1.3	1.4	0.76	1.6	2	6000
14	Sunflower Seed Meal	1760	31	2.5	21	0.4	0.3	1.3	0.5	0.6	1.5	0.83	5500
15	Peanut Meal	2200	42	1.9	17	0.2	0.2	1.8	0.5	0.5	1.4	1.3	3900
16	Dried Buttermilk	2730	32	5	0.4	1.3	0.9	2.4	0.7	0.5	1.6	1.1	2500
17	Foka	2700	14	1.8	10.1	2.25	1	0.71	0.27	0.09	0.57	0.64	2000
18	Pea	2200	22	1.1	6	0.15	0.1	1.6	0.31	0.24	0.94	0.48	22000
19	Soybean	3510	38	18	5	0.25	0.25	2.4	0.51	0.55	1.5	1.15	6500
20	MBM	2190	52	10	2.8	10	5.1	2.61	0.69	0.27	1.74	1.38	5000
21	Beer Yeast	1850	35	5	3	0.13	0.5	2.6	2.4	1.63	1.5	2.83	3500
22	Torula Yeast	1850	48	5	2	0.57	0.5	3.8	0.8	0.5	2.6	1.4	3000
23	Skimmed Milk	2510	33	0.9	0.2	1.3	1	2.3	1	0.45	1.7	1.42	30000
24	Fish Flour (Ancovetta)	2830	65	4	1	4	2.6	5.2	1.8	0.8	2.6	2.8	7500
25	Fish Flour (Herring)	2640	72	10	1	2	1.5	6.4	2	0.9	2.8	3.2	8000
26	Fish Flour (Menhaden)	2650	54	9	1	5.5	2.8	4	1.3	0.8	2.6	2.24	8500
27	Quill Flour	2310	85	2.5	1.5	0.32	0.32	1.5	0.5	0.5	0	3.5	5000
28	Blood Flour	2750	85	1.1	1	0.15	0.32	6.9	6.9	1.1	3.7	8.3	5000

REFERENCES

[1] K. Zaheer, "An Updated Review on Chicken Eggs: Production, Consumption, Management Aspects and Nutritional Benefits to Human Health," *Food Nutr. Sci.*, vol. 6, no. 6, pp. 1208–1220, 2015.

[2] National Research Council, *Nutrient Requeriments of Poultry*. Washington: National Academy Press, 1994.

[3] O. M. Bamiro and A. M. Shittu, "Vertical integration and cost behavior in poultry industry in Ogun and Oyo states of Nigeria," *Agribusiness*, vol. 25, no. 1, pp. 1–15, 2009.

[4] R. A. Rahman, C. Ang, and R. Ramli, "Investigating Feed Mix Problem Approaches : An Overview and Potential Solution," vol. 4, no. 10, pp. 750–758, 2010.

[5] P. Saxena, "Comparison of Linear and Nonlinear Programming Techniques for Animal Diet," *J. Appl. Math.*, vol. 1, no. 2, pp. 106–108, 2012.

[6] M. Akif Şahman, M. Çunkaş, Ş. Inal, F. Inal, B. Coşkun, and U. Taşkiran, "Cost optimization of feed mixes by genetic algorithms," *Adv. Eng. Softw.*, vol. 40, no. 10, pp. 965–974, 2009.

[7] V. N. Wijayaningrum and W. F. Mahmudy, "Fodder Composition Optimization using Modified Genetic Algorithm," *Int. J. Intell. Eng. Syst.*, pp. 1–6, 2017.

[8] T. N. Fatyanosa and W. F. Mahmudy, "(in press). Modified Evolution Strategies for Beef Cattle Feed Optimization," *Int. J. Intell. Eng. Syst.*, pp. 1–7, 2017.

[9] V. N. Wijayaningrum, W. F. Mahmudy, and M. H. Natsir, "Optimization of Poultry Feed Composition using Hybrid Adaptive Genetic Algorithm and Simulated Annealing," *J. Telecommun. Electron. Comput. Eng.*, pp. 1–5, 2017.

[10] R. A. Rahman, R. Ramli, Z. Jamari, and K. R. Ku-Mahamud, "Evolutionary Algorithm Approach For Solving Animal Diet Formulation," *Proc. 5th Int. Conf. Comput. Informatics, ICOCI 2015*, no. 32, pp. 274–279, 2015.

[11] A. A. Altun and M. A. Şahman, "Cost optimization of mixed feeds with the particle swarm optimization method," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 383–390, 2013.

[12] Y. Dong, J. Tang, B. Xu, and D. Wang, "An application of swarm optimization to nonlinear programming," *Comput. Math. with Appl.*, vol. 49, no. 11–12, pp. 1655–1668, 2005.

[13] Y. Tan, S. Yuhui, F. Buarque, A. Gelbukh, S. Das, and A. Engelbercht, *Advances in Swarm and Computational Intelligence*, vol. 9142. 2015.

[14] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, no. 1, pp. 89–99, 2007.

[15] J. Liang, B. Qu, P. N. Suganthan, and B. Niu, "Dynamic Multi-Swarm Particle Swarm Optimization for Multi-objective optimization problems," *Cec*, no. 60905039, pp. 1–8, 2012.

[16] X. Lai and G. Tan, "Studies on migration strategies of multiple population parallel particle swarm optimization," in *Proceedings - International Conference on Natural Computation*, 2012, no. Icnc, pp. 798–802.

[17] M. Peng, Y. Gong, J. Li, and Y. Lin, "Multi-Swarm Particle Swarm Optimization with Multiple Learning Strategies," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, no. 1, pp. 15–16.

[18] S. N. Indonesia and B. S. Nasional, “Laying Hen Feed - Part 1 (Layer Pre Starter) (Pakan ayam ras petelur - Bagian 1 : Sebelum masa awal (Layer pre starter)),” 2016.
[19] S. N. Indonesia and B. S. Nasional, “Laying Hen Feed - Part 2 (Layer Starter) (Pakan ayam ras petelur ─ Bagian 2 : masa awal (starter)),” 2016.
[20] S. N. Indonesia and B. S. Nasional, “Laying Hen Feed - Part 3 (Layer Grower) (Pakan ayam ras petelur (layer) Bagian 3: Layer Grower),” 2016.
[21] S. N. Indonesia and B. S. Nasional, “Laying Hen Feed - Part 4 (Pre Layer) (Pakan ayam ras petelur - Bagian 4 : Sebelum produksi (Pre layer)),” 2016.
[22] S. N. Indonesia and B. S. Nasional, “Laying Hen Feed (Layer) Part 5 : Production Period (Pakan ayam ras petelur (layer) Bagian 5: Masa produksi),” 2016.
[23] S. N. Indonesia and B. S. Nasional, “Laying Hen Feed Part 6 : Post Production Period (Pakan ayam ras petelur - Bagian 6 : Setelah puncak produksi),” 2016.
[24] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *MHS’95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995.
[25] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” *Evol. Comput. Proceedings, 1998. IEEE World Congr. Comput. Intell. 1998 IEEE Int. Conf.*, pp. 69–73, 1998.