

Cost optimization of mixed feeds with the particle swarm optimization method

Adem Alpaslan Altun · Mehmet Akif Şahman

Received: 4 April 2011 / Accepted: 5 July 2011 / Published online: 21 July 2011
© Springer-Verlag London Limited 2011

Abstract In this study, the best mixed feed was prepared by using the algorithm of particle swarm optimization (PSO) and by taking into account the breeding type and method of the poultries and various farm animals (cattle, sheep, rabbit), their needs, ages, and feeding costs and optimizing them all. Results obtained through PSO were compared through linear programming and real-coded genetic algorithm. According to the results that were obtained, PSO produces more rapid, more stable, and optimum values.

Keywords Mixed feed · Particle swarm optimization · Real-coded genetic algorithm · Linear programming · Cost optimization

1 Introduction

Compound that is obtained after some products that can not be consumed as human food such as starch, flour, oil, and side products of the alcohol industry, herbal products like grains and legumes, and some of the animal products that pass through screeving, crushing, mixing processes is called as mixed feed. As per the description specified in the law on fodders, they are organic and inorganic substances or their compounds that are not harmful to animal health

when they are given to animals within specific limits and under certain conditions and with the purpose of meeting the animals' living and production needs in terms of substances and energy. In another description, mixed feed is the mixture of the organic and inorganic substances consumed through mouth which has a guaranteed structure and enables the domesticated animals to yield qualified products in large quantities [1].

The selection of a good mixed feed is the leading problem for a producer. This is because of the fact that the cost of feeds is the most important expense of the livestock farming enterprises. The basic requirement for an economical breeding is to keep this expense at a minimum level. In the feed industry, a linear programming approach has been used to determine the cheapest blend of available raw materials that meet certain nutritional requirements. During cost optimization, needs of the animals and the compounds of the feed raw materials should be recognized sufficiently and the appropriate mixtures should be prepared accordingly [2]. Therefore, the basic requirement of a good mixed feed is first to obtain the feed mixtures that will meet the needs of the animals and then to choose the one with minimum cost among these mixtures. The use of an optimization method is inevitable in order to fulfill all these objectives [3]. Linear and nonlinear programming techniques are used in the solution of many problems. These are as follows: sharing the milk resources during cheese production [4], animal diet formulation [5], assessing the life span [6], assessment of the nitrogen tax scenarios [7], optimization of performance according to energy level in the broiler diet formulation [8], and determining the cost and benefits of the separation of GM mixed feeds [9].

A superiority of a genetic algorithm to standard nonlinear programming techniques is that it can calculate global minimum instead of local minimum. Its superiority

A. A. Altun (✉)
Department of Electronic and Computer Science Education,
Technical Educational Faculty, Selcuk University,
Selcuklu/Konya, Turkey
e-mail: altun@selcuk.edu.tr

M. A. Şahman
Güneşsınır Vocational High School, Selcuk University,
Konya, Turkey

can be that it does not need derivation which, occasionally, can be obtained difficultly in mathematical functions [10]. Genetic algorithms can solve complex optimization problems quickly and can be applied in an easy and simple manner [11]. The use of standard, nonlinear programming techniques for the solution of nonlinear problems in the mixed feed cost optimization can be time-consuming and insufficient [12]. However, particle swarm optimization (PSO) has an algorithm structure that is quite simple and successful in solving many optimization problems [13]. It operates more rapidly than the other optimization methods in its category. It was designed for the solution of nonlinear problems. It does not require derivation knowledge like genetic algorithms.

In this study, an approach finding least-cost feed mixes, which satisfy the nutritional requirements for poultry and different types of animals (cattle, sheep and rabbits, etc.), is proposed by using PSO. The results obtained are compared with both linear programming model and genetic algorithm. The overall results show that the PSO yields more rapid results with fewer penalties when compared with the other optimization algorithms.

2 Material and method

Database used in this study was received from Selcuk University, Faculty of Veterinary Science, Department of Animal Nutrition and Nutritional Diseases [14]. This database was designed in line with the norms specified by American National Research Council (NRC) [15], Turkey’s Mixed Feeds Norms, and the norms certified by various firms that keep animals for breeding.

Necessary information regarding the food raw materials that will be included in the feed mixture to be prepared in order to meet the needs of the animals was entered into the database completely by forming two groups as poultries and the other animals. Poultries (chicken, goose, ostrich, etc.) and the other animals (rabbit, heifer, cattle, etc.) constitute these two animal groups. Need constraints of each animal group differ from each other. Constraints to be taken into consideration when preparing feed mix for poultries include crude protein (CP), metabolizable energy (ME), calcium (Ca), utilizable phosphor (uP), sodium (Na), methionine + cystine (met + sis), lysin (Li), threonine (Tre), tryptophane (Tri), and linoleic acid (LA). As for the other animals fed with mixed feeds, crude protein (CP), metabolizable energy (ME), calcium (Ca), phosphor (P), sodium (Na), crude cellulose (CC), and crude ash (CA) should be taken into account [16]. There is a flexibility of change in these foodstuff needs specified here.

Raw materials that can be found easily and can provide animals with the sufficient nutrient and are appropriate to

Table 1 The nutrients contents

Nutrients	The nutrients contents										
	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11
	CP (%)	ME (Mcal/kg)	Ca (%)	P (%)	Na (%)	HS (%)	HK (%)	KM (%)	Cost (K₺)	Lower limit (kg)	Upper limit (kg)
x1	Barley	13.5	3.11	0.05	0.38	0.02	88	2.9	30	0	60
x2	Com, 2950	10.1	2.95	0.02	0.29	0.02	88	1.5	25	0	70
x3	Cottonseed meal, solvent extracted, 26% CP	29.0	2.2	0.20	1.00	0.05	90	7.3	25	0	30
x4	Soybean meal, 44% CP	46.8	3	0.32	0.71	0.04	91	6.6	50	0	40
x5	Alfalfa meal, %15	17.3	2.06	1.37	0.24	0.08	90	9.9	30	0	20
x6	Dicalcium phosphate	0	0	24	18	0.03	100	100	55	0	3
x7	Limestone	0	0	35	0	0.06	100	100	2.5	0	7
x8	Sodium chloride	0	0	0	0	39.34	100	100	10	0.25	1
x9	Vitaminmineral premix	0	0	0	0	0	100	0	250	0.25	0.35

Exchange rate about 1 US penny = 1.6 Kurush (K₺)

Table 2 The nutritional requirements in rabbit (constraints)

Animal species	Rabbit			
	Rabbit grower compound feed			
Nutrients	Min		Max	
Crude protein [CP (%)]	$z_{1_{\min}}$	16	$z_{1_{\max}}$	–
Metabolizable energy [ME (kcal/kg)]	$z_{2_{\min}}$	2.3	$z_{2_{\max}}$	–
Calcium [Ca (%)]	$z_{3_{\min}}$	0.8	$z_{3_{\max}}$	1.4
Phosphorus [P (%)]	$z_{4_{\min}}$	0.6	$z_{4_{\max}}$	–
Sodium [Na (%)]	$z_{5_{\min}}$	0.3	$z_{5_{\max}}$	0.4
Crude fiber [CF (%)]	$z_{6_{\min}}$	–	$z_{6_{\max}}$	10
Ash [Ash (%)]	$z_{7_{\min}}$	–	$z_{7_{\max}}$	10

their digestion systems are chosen in order to be included in the compound of mixed feeds. Amounts of the nutrient available in the content of these raw materials had been detected beforehand through chemical analyses and presented to the user as tables [17]. Estimated price and minimum and maximum amounts to be included in the mixed feed were also added to the raw material tables. These are not fixed; they can vary by the age and physiological state of the animal, raw material stock, and the current prices. Moreover, the amounts of the nutrient in the raw materials can be different in different animal groups. In this study, two different animal groups were used. These are the animal group of poultries and the group of other animals. Nutrient ingredients of some of the basic raw materials used in the group of other animals are indicated in the Table 1.

Amounts that should be included in the mixture to be prepared are calculated in percentages by taking into account the animals' daily nutrient needs determined with respect to their growth period, productivity status and their daily feed consumptions [18]. In the calculation of the nutrient needs of the animals, formulas including factors such as live weight of the animals, their productivity status

(gaining weight, producing egg), feed consumptions, and the ambient temperature are used. Values obtained at the end of these calculations constitute our constraints. The first constraint to be paid ultimate attention during the preparation of the mixture is to achieve these constraints precisely or to keep them at acceptable levels. Values of the rabbit growth feed to be prepared for the rabbit that is included in the group of other animals are presented in the Table 2.

In this study, PSO method was used in the cost optimization of the mixed feeds. In today's world, many methods designed by being inspired by biological systems are used in solving the optimization problems. For instance, artificial neural networks are simplified models of human brain, and the genetic algorithm was created by being inspired by human evolution. Social systems that are other forms of biological systems particularly examine the interaction of individuals with their social environment and their collective (common) behaviors. These behaviors are named as "swarm intelligence" [19].

PSO was designed in order to solve nonlinear problems. They are used in order to find solution to multiparameter and multivariate optimization problems. PSO shows many similarities with evolutive calculation techniques like genetic algorithms. System is launched by a population including random solutions, and it searches the most optimum solution by updating the population with each circle. Probable solutions called as particle in PSO wander around the problem space by following the Global Best and the Local Best that it obtained through its own experiences. The most important difference of PSO than the classical optimization techniques is that it does not need any information other than objective function. Applying PSO to different optimization problems is considerably simple thanks to the limited number of parameters to be adjusted. PSO can be applied successfully in many fields such as function optimization, blurry system control, artificial neural network training, and estimation problems [20–24].

Pseudo code of the algorithm is shortly as follows:

```

For
    Set the initial conditions for each particle
End
Do
    For
        Calculate the conformity value for each particle
        If this conformity value is better than pbest;
        Set the current value as the new pbest.
    End
    Set the best of the pbest values that all the particles have found as the gbest of all particles
    For for each particle,
        Calculate the particle velocity according to (8) equation
        Update the particle position according to (9) equation
    End
While Continue until the maximum iteration number and until minimum error condition is ensured

```

3 Application of PSO algorithms to feed mixes

In this study, PSO method is modeled like below and it is applied to the problem of the mixed feed preparation (Fig. 1). Firstly, animal and feed data are entered to the system. The content of the entered animal and feed data and the number of the data can be changed. Therefore, entered data should be modeled according to PSO each time. The detailed information about the used algorithm is given at the titles below.

3.1 Initial values of particles

Dimensions of particles show difference depending on the number of the food raw material that is chosen. Each dimension in the particle takes a random value that must remain between minimum and maximum amount limits of the relevant food raw material (Table 1). $x = (x_1, x_2, \dots, x_n)$ is the food raw materials that will compose the feed mixture and n shows the number of food raw materials and

accordingly the number of dimensions belonging to each particle. $y = (y_1, y_2, \dots, y_n)$ displays the contents of feeds to be included in the mixture.

Particles that enter the first iteration are composed through the following equation:

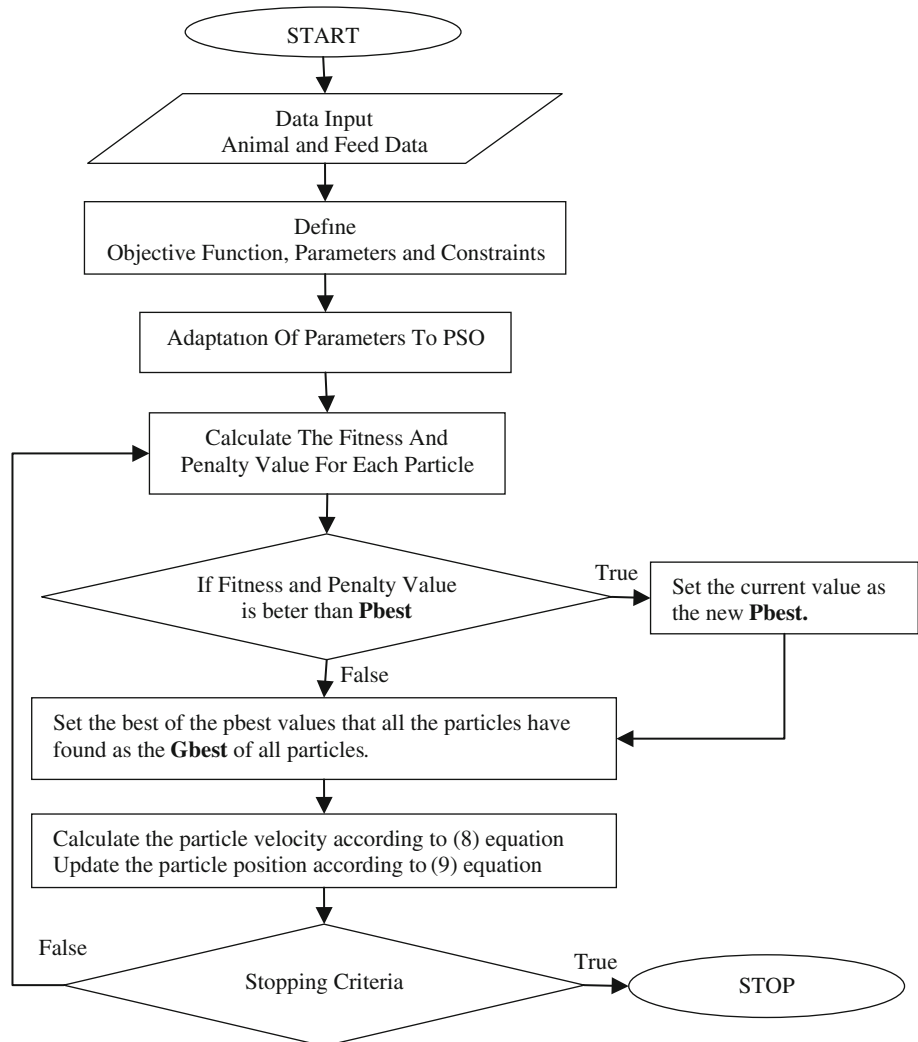
$$\text{Particle}_{ij} = y_{10j} + \text{rand}() \times (y_{11j} - y_{10j}) \tag{1}$$

Particle_{ij} refers to the value of j th dimension belonging to i th particle. y_{10j} and y_{11j} show the lower and upper limit values belonging to j th feed. As for rand() statement, it is a random number produced in the closed interval of [0, 1]. Random initial values are introduced within the requested limit intervals without any overflow by using the (1).

3.2 Constraints

The primary objective to be paid attention during the preparation of feed mixtures is to meet nutrient needs. These needs constitute our constraints. Constraint values of

Fig. 1 Flow chart for cost optimization of the feed mixes



each particle should be calculated in order to enable the control of constraint values.

Constraint values are calculated through the following equation:

$$\begin{aligned}
 z_1 &= \sum_{i=0}^n (x_i * y_1) / 100 \\
 z_2 &= \sum_{i=0}^n (x_i * y_2) / 100 \\
 &\vdots \\
 z_n &= \sum_{i=1}^j (x_i * y_n) / 100
 \end{aligned}
 \tag{2}$$

In this equation, $z = 1, 2, \dots, n$ shows the value of n item constraint(s) belonging to one particle. $x = 1, 2, \dots, j$ indicates the raw material amount in j item dimension(s) in one particle. $y = 1, 2, \dots, n$ refers to the total value of n item constraint(s) of each food raw material. Constraint values are available for each value (Table 2), and the feed mixture that will be formed at the end of feed mix should not exceed these value limits. Constraint intervals are as follows (3).

$$\begin{aligned}
 z_{1 \min} &\leq z_1 \leq z_{1 \max} \\
 z_{2 \min} &\leq z_2 \leq z_{2 \max} \\
 &\vdots \\
 z_{n \min} &\leq z_n \leq z_{n \max}
 \end{aligned}
 \tag{3}$$

3.3 Penalty values

In this study, total penalty value is calculated for each particle by punishing the overflows of minimum and maximum values of constraints. If there is not any overflow in the constraint, value of that constraint is taken as 0. Total penalties that are calculated constitute our conformity values.

Penalty values are calculated for each particle as follows:

$$\begin{aligned}
 &\text{If } z_1 \text{ value exceeds } z_{1 \max} \text{ constraint, } G(1) = ((z_1 - z_{1 \max}) * 100 / z_{1 \max}) \\
 &\text{If } z_1 \text{ value exceeds } z_{1 \min} \text{ constraint, } G(2) = ((z_{1 \min} - z_1) * 100 / z_{1 \min}) \\
 &\text{If } z_2 \text{ value exceeds } z_{2 \max} \text{ constraint, } G(3) = ((z_2 - z_{2 \max}) * 100 / z_{2 \max}) \\
 &\text{If } z_2 \text{ value exceeds } z_{2 \min} \text{ constraint, } G(4) = ((z_{2 \min} - z_2) * 100 / z_{2 \min}) \\
 &\vdots \\
 &\text{If } z_n \text{ value exceeds } z_{n \max} \text{ constraint, } G(2n - 1) \\
 &\quad = ((z_n - z_{n \max}) * 100 / z_{n \max}) \\
 &\text{If } z_n \text{ value exceeds } z_{n \min} \text{ constraint, } G(2n) = ((z_{n \min} - z_n) * 100 / z_{n \min})
 \end{aligned}
 \tag{4}$$

Amount limit value is also added to the total penalty apart from the constraints. At the end of this formulation, total feed mixture is demanded to be 100 kg. If there is a difference resulting from this value, this is also added to the total penalty.

$$Ap_j = \left(100 - \sum_{i=1}^n (x_{j,i}) \right)^2
 \tag{5}$$

In the (5), difference of all dimensions total of j th particle out of 100 is squared.

Conformity (total penalty) value of the target particle is calculated by determining the total of penalties to be detected as in the (6).

$$\psi_j(x) = \sum_{i=1}^k [g_i^j(x)]^2 + Ap_j
 \tag{6}$$

In the (6), j value refers to the j th particle. k value shows the number of the detected penalties.

3.4 Costs

In this study, the particle with the lowest cost was chosen out of the particles whose penalty values are zero or acceptable values. Cost of a particle is calculated as follows:

$$f(x) = \sum_{i=1}^n (x_i * c_i) / 100
 \tag{7}$$

Cost is calculated by multiplying the amount value of food raw material kept in dimensions of each particle with the price value of that feed and then dividing it with 100.

3.5 Local and global bests

Penalty and cost values calculated for each particle are kept in different areas where local and global best values are available. Local best values are the optimum solutions belonging to that relevant particle beginning from the iteration. As for global best, it is the optimum solution that all particles reach to until the existing iteration.

Values determined in the primary iteration are taken both as local and global best values. In the following iterations, if neither of the particles is within the conformity interval, the one with the lower penalty score is chosen; if one of these chromosomes is within the conformity interval, that particle is chosen; if both chromosomes are within the conformity interval, the one with the lower cost is chosen. In accordance with this selection method, local and global best values are readjusted at every iteration.

3.6 Velocity and position updates

In PSO, data available in the dimensions of each particle are called as position information. Position information is reupdated at all iterations depending on the local and global best values. During this update process, firstly, the velocity value is calculated. Velocity values are calculated separately for all dimensions of a particle. This calculation is made as follows:

$$v_i^{k+1} = v_i^k + c_1 \cdot \text{rand}_1^k (\text{pbest}_i^k - x_i^k) + c_2 \cdot \text{rand}_2^k (\text{gbest}^k - x_i^k) \quad (8)$$

c_1 and c_2 are learning factors in the (8). Taking learning factors as 2 usually enable the users to obtain better results. Rand is a random number produced in the [0, 1] closed interval. pbest_i^k is the local best value of i th particle in the k th iteration. gbest^k is the global best value found up to the k th iteration. v_i^k is the velocity value in the previous iteration.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (9)$$

New position value is found by adding the calculated velocity value to the previous position value (9).

3.7 Convergence test

New penalty and cost values are calculated following the calculation of velocity information and the update of the position information. Local and global best values are updated by taking these new values into account and iteration process is kept going. Termination process is conducted only when a determined penalty value or the desired number of iteration(s) is reached to.

4 Experimental results

In the study, cost optimization process was conducted on the test animals selected out of two animal groups. Coşkun et al. [17] had analyzed the low-cost feed mixtures for these animal groups by using Ms Excel Solver that includes linear programming techniques in its structure. However, there are some shortcomings as regard to Office programs on the grounds that they need to be developed constantly, the user can have various versions and some macros available in older versions have to be readjusted for the new versions. Furthermore, MS Excel Solver add-in can not conduct convergence in the event that it can not produce exact results and leaves unsettled.

In our previous study, real-coded genetic algorithms were used in the cost optimization due to the fact that linear programming does not achieve enough success in the solution of nonlinear problems. It was indicated in the studies which were carried out that real-coded genetic algorithm was not as successful as linear programming in the solution of linear problems, and it yielded more effective results in the solution of nonlinear or unsolvable problems in which linear programming remained incapable to solve them [12].

In the study where cost optimization of feed mixtures was conducted through real-coded genetic algorithm, the

fact that the study was carried out more slowly than the linear programming and it yielded different results in each study shows that it can not operate in a steady manner. Thus, different methods were sought for so as to ensure steadiness. At the end of the researches, it was found out that PSO yielded effective and steady results in the solution of low parameter, nonlinear problems and yielded these results rapidly thanks to its simple algorithmic structure.

In this study, PSO was used as a method with which we thought we could cover the deficiencies of real-coded genetic algorithm.

In the study, real-coded genetic algorithm and PSO were coded in matlab program in order to test them under the same conditions. Tests were conducted in the same computer environment, and the aim was to reach to correct results. Algorithms were operated ten times for the same animal species, and certain conclusions were drawn from the recorded results.

The most outstanding finding of the study was that PSO behaved rather steadily in the solution of nonlinear problems. It was determined that genetic coded algorithm could not behave in a sufficiently steady way, and it stuck to local minimums (Table 3).

Table 3 Results (stability test)

Expr. number	RCGA (penalty values)	PSO (penalty values)
Arbor acres broiler breeding, starter (0–4 week)		
1	223.219	181.775
2	219.606	181.775
3	215.357	181.775
4	215.459	181.775
5	211.099	181.775
6	220.095	181.775
7	204.723	181.775
8	211.755	181.775
9	211.753	181.775
10	207.124	181.775
Rabbit grower compound feed		
1	839.755	838.079
2	841.461	838.079
3	846.963	838.079
4	840.632	838.079
5	839.891	838.079
6	840.546	838.079
7	841.533	838.079
8	844.289	838.079
9	843.841	838.079
10	843.734	838.079

Table 4 Results (linear prg, RCGA, PSO)

Animal species and type	Hubbard (>1.5 kg) broiler, starter (0–10 days)		Arbor acres broiler breeding, starter (0–4 weeks)		Rabbit grower compound feed		Heifer compound feed	
	1	0	1	0	1	0	1	0
Linear programming								
Penalty values	0	464.92	0	1,634.53	0	932.79	0	181.25
Total cost (Kurus/kg)	55.92	59.83	49.48	46.73	26.73	30.17	36.08	28.28
Elapsed time (s)	1.53	1.48	2.01	1.36	1.42	1.35	1.23	1.33
Real code genetic algorithm								
Penalty values	1.16e–02	0.021	0.096	214.02	1.57e–010	842.27	0.035	109.42
Total cost (Kurus/kg)	70.24	72.09	54.18	86.62	32.04	33.77	33.42	29.7
Elapsed time (s)	1.78	1.73	1.65	1.76	1.59	1.63	1.72	1.71
Particle swarm optimization								
Penalty values	3.55e–13	3.84e–27	0.0077	181.76	9.15e–17	838.08	1.34e–20	109.31
Total cost (Kurus/kg)	70.19	77.91	52.87	102.00	35.69	34.09	40.79	29.66
Elapsed time (s)	1.37	1.34	1.44	1.23	1.23	1.13	1.34	1.19

1 Comparisons made for appropriately chosen ingredients, 0 comparisons made for inappropriately chosen ingredients

All methods were operated ten times in the study and the average of the results was found. When the results were examined, it was observed that low-cost feed mixes having fewer penalties were available in linear programming in the solution of linear feed mix problems. However, it is observed that RCGA and PSO produce better results in the solution of nonlinear feed mix problems. While PSO yields more rapid results with fewer penalties in the solution of nonlinear problems when compared with RCGA, its cost generally becomes too high (Table 4). Primary objective to be taken into consideration during the preparation of feed mixes is low penalty rate. PSO was accepted to produce better results than RCGA as cost was a secondary factor to observe.

5 Conclusions

RCGA and PSO that produce better results than linear programming in the solution of nonlinear feed mix problems which is one of the problems encountered in the optimization of feed mixtures were modeled in compliance with the problem and used. PSO is applied for the cost optimization of the feed mixtures and a software is developed by using Matlab environment, which provides flexible, extensible, and user-friendly framework for tuning the heuristic relevant parameters and improving the solution quality. According to results, PSO produces more rapid and steady results having low penalty rates but its cost is slightly higher than RCGA. It is thought that in the future studies, both steady and low-cost results will be able to be achieved through the use of different optimization

methods and hybrid methods. It has already been accepted that new generation optimization techniques have a great deal of future prospects as it has direct practical utility in the field of animal nutrition.

References

- Baran MS, Demirel R, Demirel DŞ, Şahin T, Yeşilbaş D (2008) Determination of the feeding values of feedstuffs and mixed feeds used in the Southeastern Anatolia region of Turkey. *Turk. J. Vet. Anim. Sci.* 32(6):449–455
- Oishi K, Kumagai H, Hirooka H (2011) Application of the modified feed formulation to optimize economic and environmental criteria in beef cattle fattening systems with food by-products. *Anim Feed Sci Technol* 165:38–50
- Dogan I, Dogan N, Akcan A (2000) Using goal programming in rational and economical animal nutrition. *Turk J Vet Anim Sci* 24:233–238
- Kerrigan GL, Norback JP (1986) Linear programming in the allocation of milk resources for cheese making. *J Dairy Sci* 69(5):1432–1440. doi:10.3168/jds.S0022-0302(86)80552-5
- Munford AG (1996) The use of iterative linear programming in practical applications of animal diet formulation. *Math Comput Simulat* 42(2–3):255–261. doi:10.1016/0378-4754(95)00115-8
- Azapagic A, Clift R (1998) Linear programming as a tool in life cycle assessment. *Int J Life Cycle Assess* 3(6):305–316. doi:10.1007/BF02979340
- Berntsen J, Petersen BM, Jacobsen BH, Olesen JE, Hutchings NJ (2003) Evaluating nitrogen taxation scenarios using the dynamic whole farm simulation model FASSET. *Agric Syst* 76:817–839. doi:10.1016/S0308-521X(02)00111-7
- Guevara VR (2004) Use of non-linear programming to optimize performance response to energy density in broiler feed formulation. *Poult Sci* 83:147–151
- Gryson N, Eeckhout M, Neijens T (2008) Cost and benefits for the segregation of GM and non-GM compound feed. 12th

- European association of agricultural economists congress. Ghent, Belgium
10. Hwang SF, He RS (2006) Improving real-parameter genetic algorithm with simulated annealing for engineering problems. *Adv Eng Softw* 37:406–418. doi:[10.1016/j.advengsoft.2005.08.002](https://doi.org/10.1016/j.advengsoft.2005.08.002)
 11. Cunkaş M (2010) Intelligent design of induction motors by multiobjective fuzzy genetic algorithm. *J Intell Manuf* 21:393–402. doi:[10.1007/s10845-008-0187-0](https://doi.org/10.1007/s10845-008-0187-0)
 12. Şahman MA, Çunkaş M, İnal F, İnal Ş, Coşkun B, Taşkıran U (2009) Cost optimization of feed mixes by genetic algorithms. *Adv Eng Softw* 40:965–974
 13. Kennedy J, Eberhart RC (1995) Particle swarm optimization. *Proc IEEE Int Conf Neural Netw IV*: 1942–1948
 14. Coşkun B, İnal F, İnal Ş (2007) Ration programs. <http://veteriner.selcuk.edu.tr/bolum/hbesleme/>, Accessed 14 June 2011
 15. Subcommittee on Poultry Nutrition (1994) NRC nutrient requirements of poultry, 9th Rev. ed edn. National Academy Press, Washington DC
 16. Şahman MA (2008) Cost optimization of feed mixes by using genetic algorithms. Dissertation, Selcuk University, Konya
 17. Coşkun B, Şeker E, İnal F (2000) Feeds and technology. Selcuk University, Veterinary Medicine Faculty Publication Unit, Konya
 18. Pesti GM, Miller BR (1993) Animal feed formulation: economics and computer applications. Springer, Berlin
 19. Eberhart R, Shi Y, Kennedy J (2001) *Swarm Intelligence*. Morgan Kaufmann, San Mateo, CA
 20. Chen D, Zhao C, Zhang H (2011) An improved cooperative particle swarm optimization and its application. *Neural Comput Appl* 20:171–182. doi:[10.1007/s00521-010-0503-4](https://doi.org/10.1007/s00521-010-0503-4)
 21. Wang J, Yin Z (2008) A ranking selection-based particle swarm optimizer for engineering design optimization problems. *Struct Multidisc Optim* 37:131–147. doi:[10.1007/s00158-007-0222-3](https://doi.org/10.1007/s00158-007-0222-3)
 22. Gao Z, Liao X (2011) Rational approximation for fractional-order system by particle swarm optimization. *Nonlinear Dyn* 1–9. doi:[10.1007/s11071-011-0075-6](https://doi.org/10.1007/s11071-011-0075-6)
 23. Chen L-F, Su C-T, Chen K-H, Wang P-C (2011) Particle swarm optimization for feature selection with application in obstructive sleep apnea diagnosis. *Neural Comput Appl* 1–10. doi: [10.1007/s00521-011-0632-4](https://doi.org/10.1007/s00521-011-0632-4)
 24. Das A, Bhattacharya M (2011) Affine-based registration of CT and MR modality images of human brain using multiresolution approaches: comparative study on genetic algorithm and particle swarm optimization. *Neural Comput Appl* 2:223–237. doi:[10.1007/s00521-010-0374-8](https://doi.org/10.1007/s00521-010-0374-8)