Computers and Concrete, Vol. 10, No. 3 (2012) 295-306
DOI: http://dx.doi.org/10.12989/cac.2012.10.3.295

Optimum design of a reinforced concrete beam using artificial bee colony algorithm

H.T. Ozturk*, Ay. Durmus and Ah. Durmus

Department of Civil Engineering, Karadeniz Technical University, Trabzon, Turkey

(Received May 10, 2011, Revised January 18, 2012, Accepted March 27, 2012)

Abstract. Optimum cost design of a simply supported reinforced concrete beam is presented in this paper. In the formulation of the optimum design problem, the height and width of the beam, and reinforcement steel area are treated as design variables. The design constraints are implemented according to ACI 318-08 and studies in the literature. The objective function is taken as the cost of unit length of the beam consisting the cost of concrete, steel and shuttering. The solution of the design problem is obtained using the artificial bee colony algorithm which is one of the recent additions to metaheuristic techniques. The artificial bee colony algorithm is imitated the foraging behaviors of bee swarms. In application of this algorithm to the constraint problem, Deb's constraint handling method is used. Obtained results showed that the optimum value of numerical example is nearly same with the existing values in the literature.

Keywords: optimization; metaheuristic; artificial bee colony algorithm; reinforced concrete beams.

1. Introduction

Since reinforced concrete structures are used extensively in the world, they are very important for civil engineers. In traditional design procedure shown in Fig. 1, the designer must verify problem requirements by mathematical analysis. If such requirements are not satisfied, then dimensions and/or reinforcement of RC elements are changed and a new solution is performed based on engineering perception. This repeated process consumes considerable time, until a suitable section is found. However optimal design procedure which consists of changing the design by minimizing an object function under some constraints is an alternative to the traditional design method (Fig. 2) (Coello Coello *et al.* 1997).

Haug and Kirmser (1967) used an iterative method based on generalized Newton's algorithm to solve statically determinate beams. The study was one of the first modern undertakings to use a digital computer as an optimal design tool. A method based on an energy criterion and a search algorithm based on constraint gradient values was developed by Venkayya (1971) for the design of structures under static loading. In the study the parameter to be minimized is the weight of the structural elements and his method is also applicable to the design of trusses, frames and beams. Karihaloo (1979) presented a solution of the problem of minimizing the maximum deflection of a simply supported beam under a transverse concentrated load. Sauma and Murad (1984) developed a method for minimizing the cost design of simply supported uniformly loaded partially prestressed

* Corresponding author, Ph.D., E-mail: htozturk@ktu.edu.tr

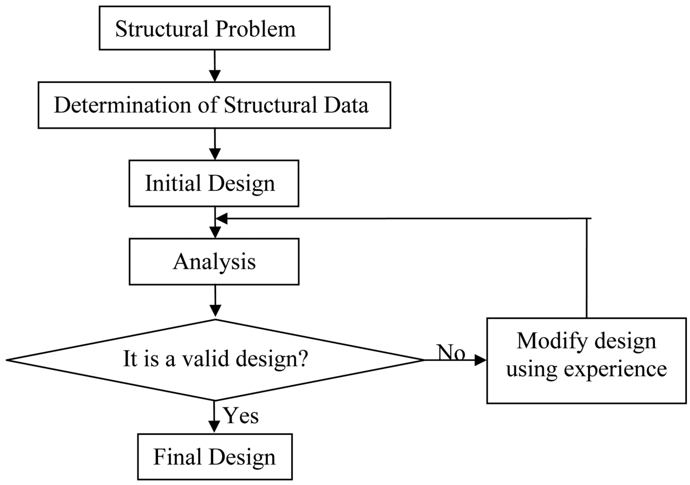

Fig. 1 Traditional design process (Coello Coello *et al.* 1997)

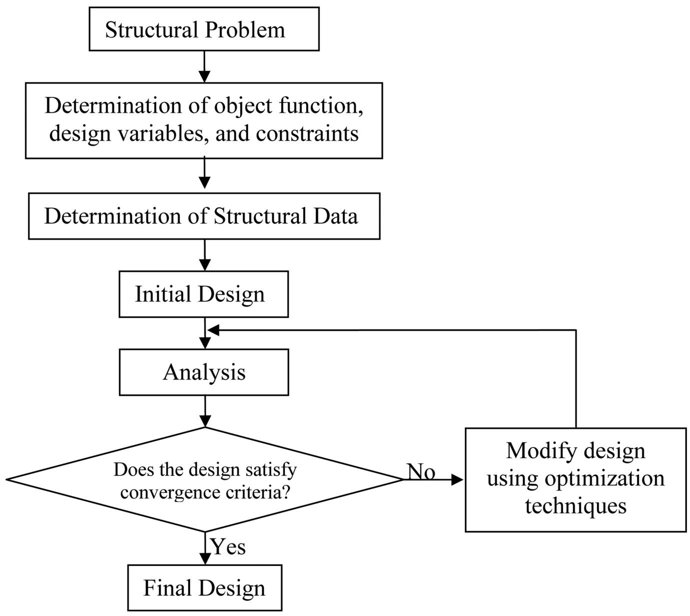

Fig. 2 Optimum design process (Coello Coello *et al.* 1997)

concrete beams using the penalty functions method coupled with quasi-Newton unconstrained optimization techniques. Zielinski (1995) used the internal penalty function for minimum cost design of reinforced concrete short tied rectangular columns based on Canadian standard specifications. Govindaraj and Ramasamy (2007) used genetic algorithm to optimize RC columns and compared the results with obtained in Zielinski's (1995) study. In these optimization problems strength, serviceability, ductility and side limitations are considered as the design constraints and they are implemented from Indian code of practice. Leroy Friel (1974) proposed an equation for RC beams to obtain optimum steel percentage and used moment strength constraints. Chou (1977) used

Lagrange multiplier method to obtain minimum cost design of reinforced T-beam sections based on ACI requirements. Kirsch (1983) presented three level iterative optimization procedure for multi span RC continuous beams. Prakash *et al.* (1988) were presented a model for optimal design reinforced concrete sections using Lagrangian and simplex methods. In their optimal design cost of steel, concrete and shuttering were taken into account. A similar but also detailed model developed by Chakrabarty (1992) and using geometric programming and Newton-Rapson method to minimize the cost. Al-Salloum and Siddiqi (1994) presented a closed form solution for steel area and depth of beam section to minimize the cost of RC beams. Coello Coello *et al.* (1997) presented a model using genetic algorithm for optimal design to minimize the cost of a rectangular reinforced concrete beam based on strength design procedures, but also considered the cost of concrete steel and shuttering. This work is one of the pioneering works on optimum design of RC structures using metaheuristic methods. Also in this paper we follow Coello Coello *et al.*'s and Chakrabarty's models to examine compatibility of our algorithm. Govindaraj and Ramasamy (2005) presented an optimum design of RC continuous beams using genetic algorithm based on Indian standard specifications.

Optimum design of RC frames were investigated by many researchers and they used such methods as linear programming (Krishnamoorthy and Munro 1973), optimality criteria method (Fadaee and Grierson 1996), direct search method (Choi and Kwak 1990, Kwak and Kim 2008), simulated Annealing (Balling and Yao 1997), harmony search (Akın 2010) and genetic algorithm (Rajaev and Krishnamoorthy 1998, Camp *et al*. 2003, Lee and Ahn 2003, Kwak and Kim 2009).

This paper focuses on the use of a metaheuristic artificial intelligence technique called artificial bee colony (ABC) algorithm (Karaboga and Basturk 2007). The main purpose of this study is to present the optimal design with artificial bee colony algorithm that minimizes the cost of singly reinforced rectangular reinforced concrete beams, considering the cost of concrete, steel and shuttering.

2. Simple reinforced concrete beams

When a beam is subjected to bending moments, bending strains are produced. Under positive moment, compressive strains are produced in the top of the beam and tensile strains are produced in the bottom. The following basic assumptions are made when using strength design:

- Plane section before bending remain plane after bending.
- At ultimate capacity, strain and stress are not proportional.
- Strains in the concrete is proportional to the distance from the neutral axis and the ultimate concrete strain is 0.003
- The modulus of elasticity of the reinforcing steel is 200000 MPa.
- The average compressive stress in the concrete is 0.85 f_c'. f_c' is indicates the specific compressive strength of the concrete.
- The average tensile stress in the reinforcement doesn't exceed f_y. f_y is indicates the specific yield strength of reinforcement.

For purpose of simplification and practical application, an equivalent rectangular concrete stress distribution was proposed by Whitney (1942) and subsequently adopted by the ACI Code. With respect to this equivalent stress distribution as shown in Fig. 3, the average stress distribution is taken as 0.85 f_c', acting over upper area of the beam cross section defined by the width b and depth a. The value of a is determined using a coefficient β_1 as

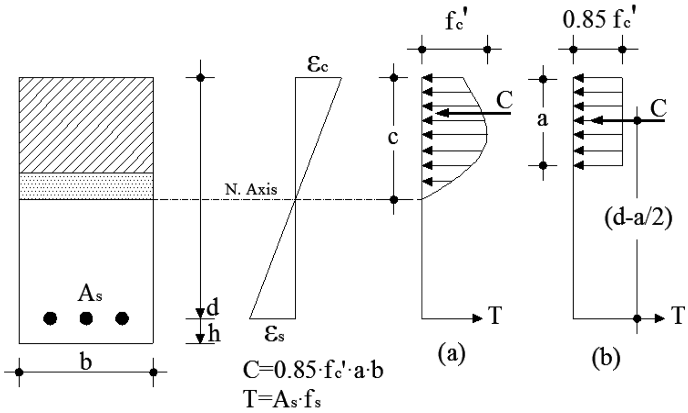

Fig. 3 Strain and stress distribution across beam depth: (a) stress block and (b) equivalent stress block

$$a = \beta_1 \cdot c \tag{1}$$

With this assumption the moment resistance of the section, as the nominal strength M_n, can be expressed as

$$M_n = A_s \cdot f_y \cdot \left(d - \frac{a}{2}\right) \tag{2}$$

where, A_s is area of reinforcement and d is distance from extreme compression fiber to the centroid of reinforcement (ACI 318 2008).

3. Formulation of optimum design problem

General three phases are considered in the optimum design of a structure or structural element. These are structural modeling, optimum design modeling and the optimization algorithm. For optimum design modeling, design variables, objective function, constraints and constraint handling method are decided in this section.

3.1 Design variables

The design variables chosen for the formulation are related to cross sectional dimensions of the beam and steel reinforcement area (Fig. 4). Three design variables were taken into consideration like Chakrabarty's (1992) and Coello Coello *et al.*'s (1997) study to compare the results. These include width of the beam (b), depth of the beam (d) and area of steel reinforcement (A_s).

3.2 Objective function

Objective function is the total cost per unit length of the beam consisting of cost components due to concrete, steel and shuttering. The cost per unit length of the beam is calculated by the following expression (Chakrabarty 1992, Coello Coello *et al.* 1997)

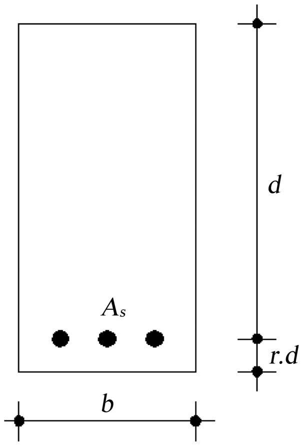

Fig. 4 Schematic cross section of a singly reinforced rectangular beam and design variables

$$f_{cost}(x) = C_1 \cdot A_s + C_2 \cdot b \cdot d + C_3 \cdot d + C_4 \cdot b \tag{3}$$

where C_1 is the cost coefficient due to volume of tensile steel reinforcement ($/cm^3), C_2 is the cost coefficient due volume of concrete in the beam ($/cm^3), C_3 is the cost coefficient due to shuttering along the vertical surfaces of the beam ($/cm^2) and C_4 is the cost coefficient due to shuttering along the bottom horizontal surface of the beam ($/cm^2). These coefficients are determined by following expressions

$$C_1 = W_s \cdot C_s \quad (\$/\mathrm{cm}^3) \tag{4}$$

Where, W_s is the unit weight of steel ($W_s = 0.00785$ kg/cm^3), C_s is unit cost of steel ($/kg).

$$C_2 = (1+r)C_c \cdot 10^{-6} \quad (\$/\mathrm{cm}^3) \tag{5}$$

Where, C_c is the unit cost of concrete and r is the cover ratio.

$$C_3 = 2(1+r)C_r \cdot 10^{-4} \quad (\$/\mathrm{cm}^2) \tag{6}$$

Where, C_r is the unit cost of shuttering.

$$C_4 = C_r \cdot 10^{-4} \quad (\$/\mathrm{cm}^2) \tag{7}$$

3.3 Constraints

Considered constraints are expressed in a normalized form as given below

$$g_1(x) = \frac{M_u}{\phi M_n} - 1 \le 0 \tag{8}$$

$$g_2(x) = \frac{0.25 \cdot d}{b} - 1 \le 0 \tag{9}$$

$$g_3(x) = \frac{b}{0.60 \cdot d} - 1 \le 0 \tag{10}$$

The first constraint is the strength constraint of the beam and here M_u is the ultimate applied bending moment at cross section, M_n is the nominal moment strength and ϕ is the strength reduction factor. Constraints $g_2(x)$ and $g_3(x)$ are the weight-height ratio constraints and given as $0.25 \leq b/d \leq 0.60$ by Coello Coello *et al.* (1997). This expression is divided into two parts as $g_2(x)$ and $g_3(x)$ and normalized. These limits allow us to have a reasonable amount of reinforcement steel in our designs, so that we can quarantee a good adherence between steel and concrete and we can provide a good control of beam's deflection (Coello Coello *et al.* 1997).

3.4 Constraint handling method

Most of the optimization problem; in science and engineering involve a number of constraints by which the optimal solution must be satisfied. A constraint optimization is usually written as

$$\begin{aligned} &\text{optimize } f(x) \\ &\text{Subject to} \quad g_j(x) \leq 0 \qquad i = 1, \ldots, J \\ &\qquad\qquad\quad h_k(x) = 0 \qquad k = 1, \ldots, K \\ &\qquad\qquad\quad x_i^{\min} < x_i \leq x_i^{\max} \quad i = 1, \ldots, n \end{aligned} \tag{11}$$

In these expressions there are n variables, J is greater than and equal to type inequality constraints and K equality constraints. If a solution in the algorithm is not satisfy these constraints, is required that using a constraint handling method to make a decision about choosing a new evaluated solution instead of old solution or not. Constraint handling methods used with evolutionary algorithm can be classified into some categories such as: (1) Methods based on preserving feasibility of solution, (2) methods based on penalty functions, (3) methods making distinction between feasible and infeasible solutions, (4) methods based on decoders and (5) hybrid methods (Deb 2000, Coello Coello 2002).

In this paper Deb's constraint handling method, which belongs to category (3), is used. This method proposes to use a tournament selection operator, where two solutions are compared at a time and three criteria are always enforced such as: (1) Any feasible solution is preferred to any infeasible solution, (2) among two feasible solutions, the one having better objective function value is preferred and (3) among two infeasible solutions, the one having smaller constraint violation is preferred (Deb 2000).

4. Artificial bee colony algorithm

Behavior of real bees was modeled by Tereshko (2000). This model consists of three essential components such as; food sources, employed and unemployed bees. Bees select a food source according to its closeness, richness and taste of the nectar, ease of extracting this nectar. Employed bees employed at a specific food source which is discovered before. They carry information about distance, the direction and profitability of the source and share it with the other bees in the hive. Unemployed bees are divided into two groups. One of the groups is called scout bees who search the environment randomly and the other group called onlookers who try to find a food source by means of the information given by the employed bees.

These foraging behaviors of bee swarms are imitated by several algorithms such as; Bee colony optimization (Teodorovic 2003, Teodorovic and Orco 2005), virtual bee (Yang 2005), bee (Pham *et* *al.* 2006) and artificial bee colony (Karaboga and Basturk 2007, Singh 2009, Sonmez 2011) algorithms. Although all the algorithms are basically similar, there are some differences between them (Karaboga and Basturk 2008).

In artificial bee colony (ABC) algorithm, each food source corresponds to a possible solution of a given optimization problem. Quality and the location of the food source represent fitness of solution and design variables respectively. First half of all bees consist of the employed bees and second half includes onlooker bees. At the beginning, algorithm generates random solutions for all bees. This operation can be defined as

$$x_{ij} = x_j^{\min} + \mathrm{rand}(0,1)(x_j^{\max} - x_j^{\min}) \tag{12}$$

where, $i = 1,2\ldots SN$ and $j = 1,2\ldots D$, D is the number of design variables and SN is the number of employed or onlooker bees.

Then employed bees determine candidate food sources in the neighbourhood of the food sources in their memory and evaluate its nectar amount. When they produce a candidate food source, algorithm uses following expression

$$v_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) & R_j < \mathrm{MR} \\ x_{ij} & \text{otherwise} \end{cases} \tag{13}$$

where, $k \in \{1,2,\ldots,SN\}$ and $j \in \{1,2,\ldots,D\}$ are randomly chosen indexes, but k must be different from i. ϕ_{ij} is a random number between [-1,1]. This parameter controls the production of neighbor food sources around x_{ij}. R_j is a random number between [0,1] and MR is a control parameter between [0,1].

Employed bees share their information with onlooker bees in the hive and onlooker bees select one of the food sources depending on the information given by the employed bees. After this step onlooker bees produce candidate food sources, according to probability value of the old sources, in the neighbourhood of the food source chosen by them. In other words onlooker bees select a food source according to a probability proportional to the amount of nectar and constraint violations. Probability value calculated by

$$p_i = \begin{cases} 0.5 + \left(\dfrac{\mathrm{fitness}_i}{\sum_{j=1}^{SN} \mathrm{fitness}_j} \right) \cdot 0.5 & \text{if constraints are satisfied} \\ \left(1 - \dfrac{\mathrm{violation}_i}{\sum_{j=1}^{SN} \mathrm{violation}_j} \right) \cdot 0.5 & \text{if constraints are satisfied} \end{cases} \tag{14}$$

Where, $\mathrm{violation}_i$ is the constraint violation of ith food source (solution). Fitness of a food source is determined as

$$\mathrm{fitness}_i = 1/(1 + f_i) \tag{15}$$

Where, f_i is the value of objective function for ith solution. In these equations, probability values of

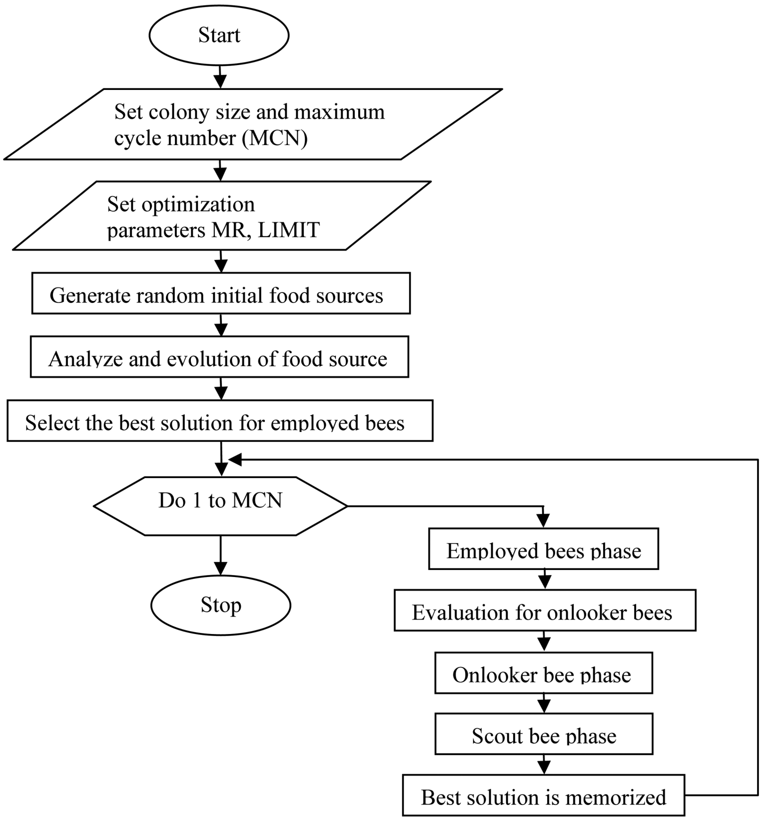

Fig. 5 Flow chart of artificial bee colony algorithm (Sonmez 2011)

infeasible solutions are between 0 and 0.5 while those of feasible ones are between 0.5 and 1. By a selection mechanism like roulette wheel, solutions are selected probabilistically proportional to their fitness values in case of feasible solutions and inversely proportional to their violation values in case of infeasible solutions. Thus, solutions in feasible region have a threshold value of 0.5 (Karaboga and Akay 2011).

Produced new solutions performance is compared with that of its old one. If the new food source has an equal or better nectar than the old source, it is replaced with the old one in the memory. If there is no improvement in the amount of nectar from a source after predefined iteration number (LIMIT), this source is discarded and its employed bee becomes scout bee. A new food source is randomly generated by the scout bee. This process is repeated until the iteration is reached a predefined maximum cycle number (MCN) or a termination criterion is satisfied (Karaboga and Basturk 2009). Another optimization parameter called Scout Production Period (SPP) which is the period of scout bee production used in the algorithm. The flow chart of algorithm is shown in Fig. 5.

5. Numerical example

In this paper, Coello Coello *et al.*'s (1997) numerical example taken from Everard (1993) was chosen for comparing the results. Rectangular concrete simple supported beam has a span (s) of 10 m is subjected to a uniform dead load of 15 kN/m and a uniform live load of 20 kN/m. The specified steel yield strength is f_y = 300 MPa, and the specified compressive strength of concrete f_c' = 30 MPa. The unit cost of steel (C_s), concrete (C_c) and shuttering (C_r) are \$0.75/kg, \$64.5/m^3 and \$2.155/m^2 respectively. Cover ratio (r) is 0.10, capacity reduction factor is 0.90, and unit weight of concrete is 2323 kg/m^3. The ultimate uniform load is calculated as

$$W_u = 1.4 \cdot DL + 1.7 \cdot LL \tag{16}$$

Because each solution has a different dimensions, self weight of the beam is calculated and added to the dead load for each solution. Ultimate applied bending moment (M_u) is determined for simply supported beam as

$$M_u = W_u \cdot s^2 / 8 \tag{17}$$

The optimization software based on the artificial bee colony algorithm was developed using MATLAB version 7.6.0. It was run on a personal computer with a Pentium Dual Core 2.0 GHz processor and 3 GB memory under The Microsoft Windows Vista operating system. To investigate the effect of colony size on the convergence rate of the artificial bee colony algorithm, four different colonies consisting of 10, 20, 30 and 40 bees were used. Twenty independent runs were performed for the each colony sizes. The averages of 20 independent runs for each colony sizes were given in Fig. 6. Since the convergence rates for all sizes are very close to each other, the colony size may be set at any value between 10 and 40 for this problem.

Maximum cycle numbers (MCN) and colony size are selected as 500 and 20 respectively. Karaboğa and Basturk (2007) purposed useful ranges for optimization parameters MR, SPP and LIMIT such as between 0.3 to 0.8 for MR, 0.5·SN·D and SN·D for SPP, and 0.1·SN·D and 2·SN·D for LIMIT. In this study MR, SPP and LIMIT parameters were selected as 0.7, 30 and 60 respectively.

Comparison of geometric programming approach used by Chakrabarty (1992), genetic algorithm used by Coello Coello *et al.* (1997) and artificial bee colony algorithm are presented in Table 1. In this table optimum values of design variables, maximum and minimum value of cost function,

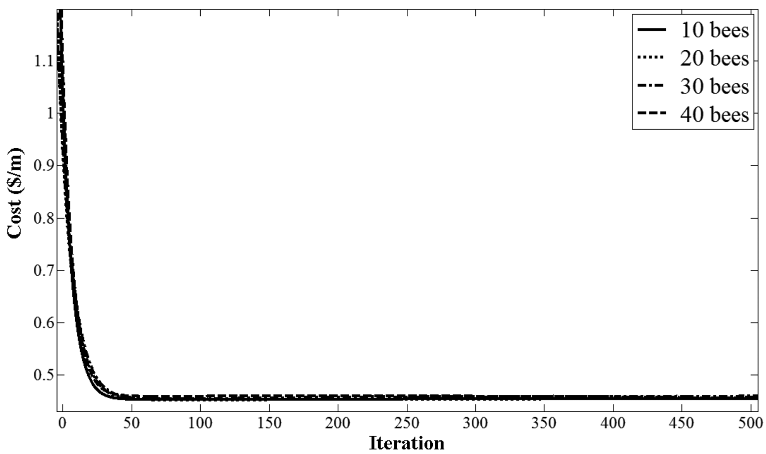

Fig. 6 Comparison of the convergence rate of four different colonies

Table 1 Comparison of geometric programming (Chakrabarty 1992), genetic algorithm (Coello Coello *et al.* 1997) and artificial bee colony algorithm

Design variables, max and min value of cost function, violation, evaluation number	Chakrabarty (1992)	Coello Coello *et al.* (1997)			In this study
	Geometric programming	GA (Binary coding)	GA (Gray coding)	GA (Floating point coding)	Artificial bee colony algorithm
As (cm^2)	37.6926	36.1893	41.5905	37.5205	37.4455
d (cm)	86.0629	89.5402	78.6177	86.6221	86.6221
b (cm)	30.0000	30.0162	30.0447	30.0022	30.0000
f_{cost} (max) ($/m)	-	-	-	-	0.4487
f_{cost} (min) ($/m)	0.4435	0.4442	0.4464	0.4436	0.4435
Evaluation number	-	-	-	-	1340
Violation*	None	None	None	None	None

*Violations of previous studies are calculated according to our formulations.

violation and number of objective function evaluations are presented.

Performed study shows that, obtained results with artificial bee colony algorithm are nearly same with Chakrabarty's and Coello Coello *et al.*'s results, and ABC algorithm is useful for structural optimization.

6. Conclusions

The artificial bee colony (ABC) algorithm, based on mimicking the behavior of honeybee swarms, is proposed as a method of optimization of singly reinforced concrete rectangular beams. Deb's method, which makes distinction between feasible and infeasible solutions, is used as the constraint handling method. The objective function is the cost of the beam for unit length. Constraints and material costs were taken from Coello Coello *et al.*'s and Chakrabarty's study. Obtained results were compared with their results. Bee colony size effect is investigated and it is presented that the colony size may be set at any value between 10 and 40. The design algorithm performs effectively in finding the optimum values of design variables. Because the results are nearly same with compared studies, the ABC algorithm is an effective tool for optimization of singly reinforced concrete beams.

Acknowledgements

This study is a part of Ph. D. Thesis of Hasan Tahsin ÖZTÜRK directed by Prof. Dr. Ing. Ahmet DURMUŞ, and it is supported by the Research Fund of Karadeniz Technical University. Project number: 2008.112.001.1.

References

ACI 318 (2008), *Building code requirements for structural concrete and commentary*, American Concrete

Institute, USA.
Akın, A. (2010), *Optimum design of reinforced concrete plane frames using harmony search algorithm,* Ph.D. Thesis, Graduate School of Natural and Applied Sciences of Middle East Technical University, Ankara, Turkey.
Al-Salloum, Y.A. and Siddiqi, G.H. (1994), "Cost optimum design of reinforced concrete beams", *ACI Struct. J.*, **91**(6), 647-655.
Balling, R.J. and Yao, X. (1997), "Optimization of reinforced concrete frames", *J. Struct. Eng.-ASCE*, **123**(2), 193-202.
Camp, C.V., Pezeshk, S. and Hansson, H. (2003), "Flexural design of reinforced concrete frames using a genetic algorithm", *J. Struct. Eng.-ASCE*, **129**(1), 105-115.
Chakrabarty, B.K. (1992), "Models for optimal design of reinforced concrete beams", *Comput. Struct.*, **42**(3), 447-451.
Choi, C.K. and Kwak, H.G. (1990), "Optimum RC member design with predetermined discrete sections", *J. Struct. Eng.-ASCE*, **116**(10), 2634-2655.
Chou, T. (1977), "Optimum reinforced concrete T-beam sections", *J. Struct. Div.*, **103**(8), 1605-1617.
Coello Coello, C.A. (2002), "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art", *Comput. Method. Appl. M.*, **191**(11-12), 1245-1287.
Coello Coello, C.A., Christiansen, A.D. and Santos Hernandez, F. (1997), "A simple genetic algorithm for the design of reinforced concrete beams", *Eng. Comput.*, **13**(4), 185-196.
Deb, K. (2000), "An efficient constraint handling method for genetic algorithms", *Comput. Method. Appl. M.*, **186**(2-4), 311-338.
Everard, N.J. (1993), *Theory and problems of reinforced concrete design*, Schaum's Outline Series, Third Edition, McGraw-Hill, USA.
Fadaee, M.J. and Grierson, D.E. (1996), "Design optimization of 3D reinforced concrete structures", *Struct. Multidiscip. O.,* **12**(2-3), 127-134.
Govindaraj, V. and Ramasamy, J.V. (2005), "Optimum detailed design of reinforced concrete continuous beams using genetic algorithms", *Comput. Struct.*, **84**(1-2), 34-48.
Govindaraj, V. and Ramasamy, J.V. (2007), "Optimum detailed design of reinforced concrete frames using genetic algorithms", *Eng. Optimiz.,* **39**(4), 471-494.
Haug, E.J. and Kirmser, P.G. (1967), "Minimum weight design of beams with inequality constraints on stress and deflection", *J. Appl. Mech.-ASME*, **34**(4), 999-1004.
Karaboga, D. and Akay, B. (2009), "A comparative study of artificial bee colony algorithm", *Appl. Soft. Comput.*, **214**(1), 108-132.
Karaboga, D. and Akay, B. (2011), "A modified artificial bee colony (ABC) algorithm for constrained optimization problems", *Appl. Soft Comput.*, **11**(3), 3021-3031.
Karaboga, D. and Basturk, B. (2007), "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems", *Proceedings of IFSA 2007*, Mexico, June.
Karaboga, D. and Basturk, B. (2008), "On the performance of artificial bee colony (ABC)", *Appl. Soft. Comput.,* **8**(1), 687-697.
Karihaloo, B.L. (1979), "Optimal design of multi-purpose tie-beams", *J. Optimiz. Theory App.*, **27**(3), 427-438.
Kirsh, U. (1983), "Multilevel optimum design of reinforced concrete structures", *Eng. Optimiz.,* **6**(4), 207-212.
Krishnamoorty, C.S. and Munro, J. (1973), "Linear program for optimal design of reinforced concrete frames", *Procced. IABSE*, **33**(1), 119-141.
Kwak, H.G. and Kim, J. (2008), "Optimum design of reinforced concrete frames based on predetermined section databese", *Comput.-Aided Civ. Inf.*, **40**(3), 396-408.
Kwak, H.G. and Kim, J. (2009), "An integrated genetic algorithm with direct search for optimum design of RC frames", *Comput.-Aided Civ. Inf.*, **41**(7), 490-500.
Lee, C. and Ahn, J. (2003), "Flexural design of reinforced concrete frames by genetic algorithm", *J. Struct. Eng.-ASCE,* **129**(6), 762-774.
Leroy Friel, L. (1974), "Optimum single reinforced concrete sections", *ACI J.*, **71**(11), 556-558.
Pham, D.T., Granbarzadeh, A., Koç, E., Otri, S., Rahim, S. and Zaidi, M. (2006), "The bees algorithm – a novel tool for complex optimization problems", *Proceedings of Innovation Production Machines and System Virtual*

Conference, http://conference.iproms.org., July.
Prakash, A., Agarwala, S.K. and Singh, K.K. (1988), "Optimum design of reinforced concrete sections", *Comput. Struct.*, **30**(4), 1009-1011.
Rajeev, S. and Krishnamoorthy, C.S. (1998), "Genetic algorithm-based methodology for design optimization of reinforced concrete frames", *Comput.-Aided Civ. Inf.*, **13**(1), 63-74.
Saouma, V.E. and Murad, R.S. (1984), "Partially prestressed concrete beam optimization", *J. Struct. Eng.-ASCE*, **110**(3), 589-604.
Singh, A. (2009), "An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem", *Appl. Soft Comput.*, **9**(2), 625-631.
Sonmez, M. (2011), "Artificial bee colony algorithm for optimization of truss structures", *Appl. Soft Comput.*, **11**(2), 2406-2418.
Teodorovic, D. (2003), "Transport modeling by multi-agent systems: a swarm intelligence approach", *Transport. Plan. Techn.*, **26**(4), 289-312.
Teodorovic, D. and Orco, M.D. (2005), "Bee colony optimization-a comparative learning approach to computer transportation problems", *Adv. IA Method. Transport.,* 51-60.
Tereshko, V. (2000), "Reaction-diffusion model of a honeybee colony's foraging behaviour", *Lecture Notes in Comput. Sci.*, **1917**, 807-816.
Venkayya, V.B. (1971), "Design of optimum structures", *Comput. Struct.*, **1**(1-2), 265-309.
Whitney, C.S. (1942), "Plastic theory of reinforced concrete design", *Trans. ASCE*, **107**, 251-326.
Yang, X.S. (2005), "Engineering optimization via nature-inspired virtual bee algorithms", *Lecture Notes in Comput. Sci.*, **3562**, 317-323.
Zielinski, Z.A., Long, W. and Troitsky, M.S. (1995), "Designing reinforced concrete short tied columns using optimization technique", *ACI Struct. J.*, **92**(5), 619-626.

CC