J. Inst. Eng. India Ser. A (December 2017) 98(4):429–435
DOI 10.1007/s40030-017-0232-0

ORIGINAL CONTRIBUTION

Design Optimization of Reinforced Concrete Beams

Sonia Chutani[1] · Jagbir Singh[2]

Received: 21 December 2013 / Accepted: 17 October 2017 / Published online: 26 October 2017

Abstract This work presents a typical optimization technique i.e. Particle Swarm Optimization (PSO) to achieve optimal design of Reinforced Concrete (RC) beams. Optimal cross-sectional sizing of an RC beam results in cost saving, but it (optimal sizing) cannot be standardized for the various factors that influence a given design. An algorithm has been developed to search for a minimum cost solution that satisfies Indian codal requirements for RC beams. The objective function consists of the cost of concrete and rebars as prevalent at the place of construction. Successful implementation of the algorithm clearly establishes PSO's ability of performance in the case of RC beams. A number of examples have been presented to show the effectiveness of this formulation for achieving optimal design.

Keywords Optimum design · Reinforced concrete beam · Particle swarm optimization · Minimum cost

Introduction

Constantly increasing need for economical structures has enhanced the interests of designers in developing superior methodologies for optimum design of structural members. The structural design codes do not primarily dwell on the optimization front and this factor is mostly based on the experience of a particular designer—which in any case cannot be considered a substitute for the tested and validated principles of optimization techniques. This paper considers the provisions of IS 456-2000 and intelligent search technique to identify the optimum solution of RC beams. Undoubtedly, optimization of RC beams in smaller projects may not be financially viable due to unconventional cross-sectional size, however in large scale projects where the same design may be used several times, the savings compound and the optimization is viable [1].

Weight and cost are the two objective criterions commonly employed for structural optimization, but for RC structures 'weight minimization' may not necessarily lead to 'cost minimization' [2]. As regards to RC structures, the pioneer application of heuristic algorithm includes the work of previous researchers [3] who used Genetic Algorithm (GA) for economic optimization of RC beams. Furthermore, many more evolutionary methods have been developed during last many years for solving linear and non linear optimization problems such as genetic algorithm, simulated annealing, harmony search, particle swarm optimization and ant colonies, to explore solutions for constrained problems. Among all, GA is an artificial intelligent method, inspired by biological phenomenon has been widely used for structural design problems. The researchers have [4] efficiently used GA for optimizing RC continuous beams. Cost optimization models for RC and PC beams using GA have also been proposed by several investigators [5]. An Artificial Neural Network (ANN) with GA for optimum design of singly and doubly reinforced beams has been presented by some of the previous investigators [6, 7] who recommended the optimum steel ratios for beams and columns in their work of optimization of RC

✉ Sonia Chutani
soniachutani_annie@yahoo.co.in

1 I.K.G Punjab Technical University, Kapurthala 144603, Punjab, India

2 Department of Civil Engineering, Guru Nanak Dev Engineering College, Ludhiana, Punjab, India

flat slab using GA. Many researchers have tried to use GA to carry out optimization process of RC frames [8–10]. Some of the previous studies for optimizing RC frames—based on Indian specifications—have also implemented the capability of GA [11, 12].

PSO has mostly been studied for steel structures [13–15] and has found limited application for reinforced concrete structures [16–18]. PSO is widely acknowledged for its simplicity and convergence speed. The features of PSO which make it attractive for use are the adjustment of only a few parameters of the algorithm, as compared with other algorithms, and its applicability for non-differentiable, non-convex and highly nonlinear problems. Also it is considered to be a relatively powerful tool with high search speed for exploring optimal solution [19–21]. Recently, performance of PSO has been evaluated by combining it with other algorithms for optimum design of RC frame structures [22, 23]. This work is concerned with optimum cost of the simply supported rectangular beams using standard PSO technique and is organized as follows.

Particle Swarm Optimization

Particle swarm optimization is basically a ‘population based’ stochastic optimization technique [19] with the traits of simplicity and fair search potential. The interaction between different particles to determine their best positions is the crux of PSO. All particles communicate with each other in search of best position and adjust their velocities accordingly.

Each ith particle vector from a set of moving particles represents a potential solution based on a fitness function, and has a position Pos_i^k and velocity Vel_i^k at the kth iteration in the problem space. Each ith vector keeps a track of its individual best position Pos_i^k, which is related with its own best fitness achieved so far at kth step in the iteration procedure. This value is identified as ‘pbest’. Likewise, the optimum position obtained so far in the swarm is stored as the global best position $gbest^k$ and identified as ‘gbest’. The new velocity of the particle is modernized as follows:

$$Vel_i^{k+1} = w^k Vel_i^k + c_1 rand_1 \left(pbest_i^k - \overline{X}_i^k \right) + c_2 rand_2 \left(gbest^k - \overline{X}_i^k \right) \quad (1)$$

$$Pos_i^{k+1} = Pos_i^k + Vel_i^{k+1} \quad (2)$$

where w^k is inertia weight at kth iteration in the first part and represents the memory of a particle during search. The inertia weighting function at each iteration is given as:

$$w^k = w_{max} - (w_{max} - w_{min}) \times k^{th}\_iter/iter_{max} \quad (3)$$

w_{max} and w_{min} represent maximum and minimum values of w respectively, where $iter_{max}$ represents maximum number of iterations and k^{th}_iter represents current iteration number. The first right hand term in (1) helps each particle to perform a global search by exploring a new search space, whereas the last two terms represent cognitive and social parts respectively in which c_1 and c_2 are the learning factors illustrating the weights of the acceleration terms that guide each particle toward the personal best and the global best positions respectively. $rand_1$ and $rand_2$ are uniformly distributed random numbers in the range 0–1, and N represents number of particles in the swarm. Each particle decides its position based on the updated velocity according to (2) which is known as flight formula. In this way, ‘velocity updating’ (1) and ‘flight formula’ (2) help the particles to locate the optimum solution in the search space. Figure 1 explains the algorithm as developed for the current problem.

In order to keep the particles within the search space, their velocities have been constrained by restricting the maximum velocity of each particle. Normally, the value of maximum velocity is selected empirically as per the characteristics of the given problem. When the value of this parameter is high, the particles start moving erratically and thereby go past a good solution, whereas when the value is small, the particle’s movement is restricted and they fall well short of the optimal solution. In the current optimization problem, the search space is bounded by $[Pos_{min}, Pos_{max}]$, and Vel_{max} has also been limited to 4.

Optimal Design Model

In the current optimization problem some of the parameters are considered as pre-assigned or fixed while others are variable. The design variables are determined such that the cost (objective function) becomes minimum. Some restrictions—called design constraints—limit the values of these design variables.

Objective Function

The total cost of the material used, which includes the cost of reinforcement (longitudinal and shear) and concrete, is taken as the objective function. Since the proposed algorithm is pertinent to an unconstrained and continuous optimization problem, the formulation of penalized objective function—including imposed penalties due to violation of constraints—is done to translate the constrained problem into an unconstrained one.

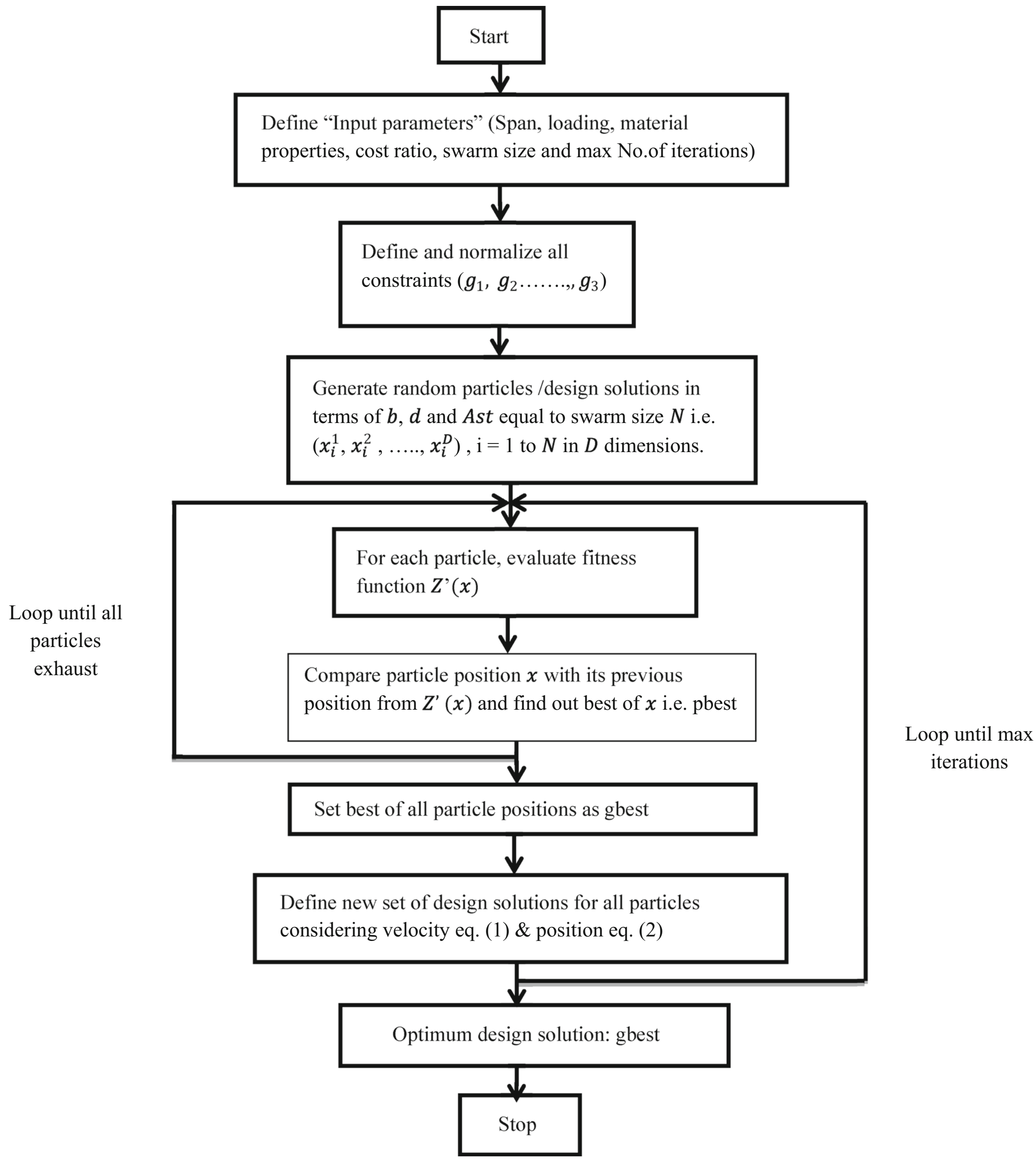

Fig. 1 Flow chart for design optimization of RC beam

Development of Objective Function

The cost of RC beam is given as:

$$C = C_{st}V_{st} + C_C V_C \tag{4}$$

where, C = total cost of beam, C_{st} = cost of steel per unit volume of steel (rate of steel), V_{st} = total volume of steel, C_C = cost of concrete per unit volume of concrete (rate of concrete), V_C = total volume of concrete.

Divide Eq. (4) by C_C,

$$\frac{C}{C_C} = \frac{C_{st}}{C_C} V_{st} + V_C$$

and substitute

$$\frac{C}{C_C} = Z, \quad \frac{C_{st}}{C_c} = \alpha \quad (\text{cost ratio})$$

and

$$V_C = V_G - V_{st}$$

where, V_G is the gross volume of beam. Thus, objective function Z is defined as:

$$\text{Minimize Z} = (\alpha - 1)\ V_{st} + V_G \quad (5)$$

Volume of steel V_{st} depends upon area of steel and its provided length. Similarly gross volume of concrete depends upon cross sectional area and length of beam.

Fixed Parameters

In the present model, all input design parameters have been considered fixed. These include span of beam, grade of reinforcement and concrete, intensity of dead and live loads, effective cover of concrete and cost ratio (ratio of unit cost of reinforcement to unit cost of concrete).

Design Variables

Independent design variables considered in the present model are width (b) and effective depth (d) of the beam. Cross-sectional area of longitudinal reinforcement (Ast) and shear reinforcement (Asv) have been calculated as dependent design parameters.

Constraints

Designs constraints considered in the present model not only considers Indian codal provisions for RC beam design (IS 456: 2000), but also few other publications [24, 25] (Table 1).

Constraint Normalization

All the constraint functions have been normalized—to speed up convergence and to prevent undue dominance of any particular constraint—as follows:

$$g_1 = 1 - \frac{b}{b_{min}} \le 0$$

$$g_2 = \left\{ \frac{x_u}{d} \Big/ \frac{x_{u,max}}{d} \right\} - 1 < 0 \quad \text{where} \quad \frac{x_u}{d} = \frac{0.87 f_y A_{st}}{0.36 f_{ck} bd}$$

$$g_3 = \frac{A_{st}}{A_{st}, max} - 1 \le 0$$

$$g_4 = 1 - \frac{A_{st}}{A_{st}, min} \le 0$$

$$g_5 = 1 - \frac{M_r}{M_n} \le 0$$

$$g_6 = \frac{l - \frac{D}{2}}{\min\left(60b, \frac{250b^2}{d}\right)} - 1 \le 0$$

$$g_7 = \frac{l}{20d} - 1 \le 0 \quad (\text{when span} \le 10\,\text{m})$$

$$g_8 = \frac{l^2}{200d} - 1 \le 0 \quad (\text{when span} > 10\,\text{m})$$

The constraint function values have been kept negative so that all constraints meet at optimal point. In case of any violation, a penalty has been imposed according to the "Constraint Handling Approach".

Constraint Handling Approach

A non-linear constrained optimization problem defined below has been converted to an unconstrained one by the use of dynamically modified penalty function approach, where penalties imposed are not stationary but gets modified during the process.

Minimize $Z(x)$

Subjected to inequality constraints:

$$g_i(x) \le 0 \quad i = 1, 2, 3\ldots m$$

and equality constraints:

$$h_i(x) = 0 \quad i = m + 1, \ldots l$$

$Z(x)$ is the objective function, $g_i(x)$ represents inequality constraints, and x is a 'n' dimensional vector of design variables.

In the PSO algorithm, objective function value indicates the favorability of two positions (old and new).

The penalized objective function (fitness function) $Z'(x)$ has been considered as in [21]:

$$Z'(x) = Z(x) + h(k)H(x), \quad x \in S \subset R^n \quad (6)$$

where, $H(x) = r \times \sum_{i=1}^{m} \theta(q_i(x)) q_i(x)^{\gamma(q_i(x))}$, $q_i(x) = \{0, g_i(x)\}$, $i = 1, 2\ldots m$, $h(k) = k\sqrt{k}$, $Z'(x)$ = penalized objective function, $Z(x)$ = original objective function, $h(k)$ = dynamically modified penalty value, k = algorithm current iteration number, r = penalty multiplier, $H(x)$ = penalty factor.

Function $q_i(x)$ is a relative violated function of the constraints, $\theta[q_i(x)]$ is a multi-segment assignment function, $\gamma\ [(q_i(x))]$ is a power of the penalty function, and $g_i(x)$ are the constraint functions.

If $q_i(x) < 1$, then $\gamma[q_i(x)] = 1$, otherwise $\gamma[(q_i(x))] = 2$.

Moreover,

$$\begin{aligned}
&\text{if } q_i(x) < 0.001 \text{ then } \theta[(q_i(x))] = 10\\
&\text{else, if } 0.001 < q_i(x) < 0.1 \text{ then } \theta[(q_i(x))] = 20\\
&\text{else, if } 0.1 < q_i(x) < 1, \text{then } \theta[(q_i(x))] = 100\\
&\text{otherwise } \theta[(q_i(x))] = 300.
\end{aligned}$$

Parameter r is problem dependent, which shall be a suitably large constant. In the current study, value of r has been set to 10^{12}).

Optimal Design Solution

The design procedure coded in C++ gets the design solution through conventional limit state method as well as through proposed PSO algorithm. The constant parameters used in Standard Particle Swarm Optimization (SPSO) are given in Table 2.

For investigating the performance of PSO, different design examples were considered and two of them are given here to compare the results with conventional Limit State Method (LSM). The necessary input parameters for these design examples of singly reinforced concrete beam carrying uniformly distributed load over the entire span are given in Table 3.

In the examples discussed below, population size of the swarm i.e. swarm size is taken as 25.

In the present study, the optimization procedure was terminated when one of the following two stopping criteria was met:

1. No. of iterations become equal to maximum specified number.
2. No significant improvement in the solution.

The objective function (Z) gets reduced from 2.96964×10^9 in conventional LSM to 2.77124×10^9 in optimum beam design and consequently, percentage saving in cost is achieved by 6.7% in beam design (Example 1, Table 4). Similarly, the objective function obtained in design example 2, gets reduced from 1.47696×10^9 to 1.35332×10^9 and 8.4% saving in cost has been achieved (Table 4). It has also been observed that the effective depth to width ratio gets increased during the process, from 1.87 to 2 in example 1 and 1.84 to 2.5 in example 2 which indicate that optimization of the section is associated with rise in depth to width ratio. Two design examples with different range of input parameter i.e. 'depth to width ratio' indicate that greater the d/b ratio, greater is the percentage saving in cost. Furthermore, rise in depth to width ratio was restrained by the constraint put on minimum width of beam.

Table 2 Constant parameters used in standard particle swarm optimizer

Parameters	Values for SPSO
Swarm size	25
Maximum iteration number	500
Vel_{max}	4
w_{max}	0.9
w_{min}	0.4
c_1	2
c_2	2

Table 3 Input parameters for design

Input parameters	Input values	
	Example 1	Example 2
Effective span, m	6	5
Load, kN/m	65	50
Grade of concrete	M20	M25
Grade of steel	Fe415	Fe500
Limits of d/b ratio	1.5–2.0	1.5–2.5
Effective cover, mm	40	40
Bearing of support, mm	300	300
Cost ratio $\frac{C_{st}}{C_c}$	100	100

To study 'convergence performance' of the algorithm, progress of design improvements has been illustrated in Figs. 2 and 3.

Parametric Study

Swarm Size

In PSO, best swarm size depends upon optimization problem. Different swarm sizes were considered to study their effect on performance of algorithm and optimum design solutions. Example 2 was optimized by taking

Table 1 Design constraints for optimal design of RC beam as per IS 456:2000

Constraint type	Particular constraint	Expression	Description
Geometric constraints	g_1	$b \geq b_{min}$	Minimum width constraint
	g_2	$x_u/d < x_u, max/d$	Ductility constraint
	g_3	$A_{st} < A_{st}, max$	Maximum tensile steel constraint
	g_4	$A_{st} > A_{st}, min$	Minimum tensile steel constraint
Behavior constraints	g_5	$M_r > M_n$	Moment capacity constraint
	g_6	$l - (D/2) < min(60b, 250b^2/d)$	Lateral stability constraint
	g_7	$l/d \leq 20$	Deflection constraint (for span up to 10 m)
	g_8	$l/d \leq 20 * (10/l)$	Deflection constraint (for span above 10 m)

Table 4 Simulation results

Beam design example	Method	*d*, mm	*b*, mm	*Ast*, mm^2	*Z* (10^9)	% Saving
Example 1	LSM	620	330	1958.45	2.96964	6.7
	PSO	616	308	1816.09	2.77124	
Example 2	LSM	460	250	1094.48	1.47696	8.4
	PSO	509	204	988.23	1.35332	

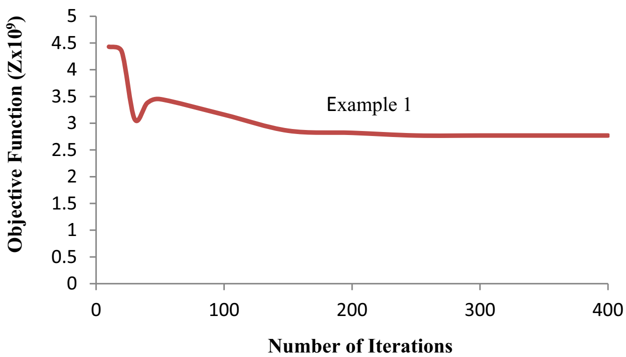

Fig. 2 Progress of design improvements for Example 1

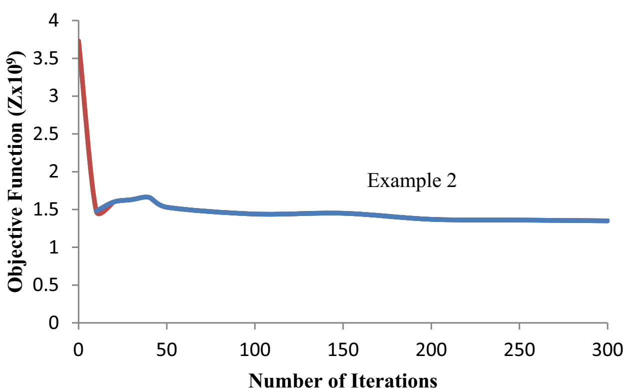

Fig. 3 Progress of design improvements for Example 2

swarm sizes equal to 5, 10 and 25. Effect of swarm size on required number of iterations for convergence has been shown in Fig. 4.

Although the optimum result largely remains unaffected by the swarm size, larger swarm size needs more iteration to get to the global optimum solution up to desired precision because the algorithm has to explore a greater area in each iteration resulting in more number of evaluations. Also the computational time is more for large swarm size. On the other hand, swarm sizes smaller than 5 have a risk of getting trapped into local minima.

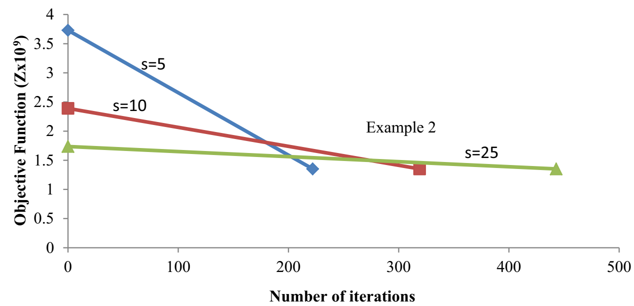

Fig. 4 Convergence curves with different swarm sizes for Example 2

Acceleration Coefficients

The relative values of acceleration coefficients c_1 (cognitive acceleration coefficient) and c_2 (social acceleration coefficient), when combined with the random numbers determine the exploratory nature of particles. The effect of acceleration coefficients has been studied for example 2 and shown in Table 5.

It has been shown in Table 5 that when cognitive acceleration coefficient has lower value than social acceleration coefficient, its convergence is slow. Even at higher values of c_1, particles tend to wander randomly. While if c_2 has much lower value than c_1 and much higher value than c_1, particles do not reach global optima for example 2. But both cognitive acceleration coefficient and social acceleration coefficient equal to 2 give fairly good results.

Concluding Remarks

The PSO has proved to be a relatively robust tool for exploring optimal solutions for reinforced concrete beams. Undoubtedly, this study is specifically carried out for simply supported beams, but the scope of proposed algorithm is wide enough to seek the optimum solution for other beams and structures. The algorithm has not proved to be very sensitive to the variation of parameters like swarm size and acceleration coefficients. A considerable percentage of saving in cost of RC beam has been found using proposed optimum design approach. As the entire optimum design algorithm has been coded in C++, time

Table 5 Convergence of algorithm for different acceleration coefficients for example 2

c_1/c_2	$Iter_{convergence}$	Z	c_2/c_1	$Iter_{convergence}$	Z
0.25/2.0	305	1.35×10^9	0.25/2.0	117	1.40×10^9
2.0/2.0	222	1.35×10^9	2.0/2.0	222	1.35×10^9
3.5/2.0	320	1.35×10^9	3.5/2.0	439	1.37×10^9

taken to get the optimum design values has almost become an insignificant dimension. The limitations and restrictions of the Indian code IS 456: 2000 have been considered as a series of constraints in the current optimization problem and applied as penalties on the fitness function of the PSO. Two design examples have been presented to demonstrate the effectiveness and efficiency of the procedure. It has been viewed that reduction in both steel area as well as concrete volume contributes towards optimization of reinforced concrete beams and cost optimization is directly proportional to the ratio of depth to width of a beam.

References

1. S. McCluskey, T.J. McCarthy, A particle swarm optimisation approach to reinforced concrete beam design according to AS3600, in *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering* (Civil-Comp Press, Madeira, 2009), pp. 1–14
2. P. Sharafi, M.N.S. Hadi, L.H. Teh, Geometric design optimization for dynamic response problem of continuous reinforced concrete beams. J. Comput. Civ. Eng. (ASCE) **28**, 202–209 (2014)
3. C.A. Coello, A.D. Christiansen, F. Santos, A simple genetic algorithm for the design of reinforced concrete beams. Eng. Comput. **13**(4), 185–196 (1997)
4. V. Govindaraj, J.V. Ramasamy, Optimum design of reinforced continuous beams by genetic algorithms. Comput. Struct. **84**, 34–48 (2005)
5. M. Alqedra, M. Arafa, M. Ismail, Optimum cost of prestressed and reinforced concrete beams using genetic algorithms. J. Artif. Intell. **4**(1), 76–88 (2011)
6. B. Saini, V.K. Sehgal, M.L. Gambhir, Genetically optimized artificial neural networks based optimum design of singly and doubly reinforced concrete beams. Asian J. Civ. Eng. (Build. Hous.) **7**(6), 603–619 (2006)
7. M.G. Sahab, Sensitivity of the optimum design of reinforced concrete flat slab buildings to the unit cost components and characteristics material strengths. Asian J. Civ. Eng. (Build. Hous.) **9**(5), 487–503 (2008)
8. C.V. Camp, S. Pezeshk, H. Hansson, Flexural design of reinforced concrete frames using a genetic algorithms. J. Struct. Eng. **129**(1), 105–115 (2003)
9. H. Kwak, J. Kim, Design an integrated genetic algorithm complemented with direct search for optimum design of RC frames. Comput. Aided Des. **41**(7), 490–500 (2009)
10. C. Lee, J. Ahn, Flexural design of reinforced concrete frames by genetic algorithm. J. Struct. Eng. ASCE **129**(6), 762–774 (2003)
11. V. Govindaraj, J.V. Ramasamy, Optimum detailed design of reinforced concrete frames using genetic algorithms. Eng. Optim. **39**(4), 471–494 (2007)
12. S. Rajeev, C.S. Krishnamoorthy, Genetic algorithm—based methodology for design optimization of reinforced concrete frames. Comput. Aided Civ. Infrastruct. Eng. **13**, 63–74 (1998)
13. S. Gholizadeh, Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization. Comput. Struct. **125**, 86–99 (2013)
14. G.C. Luh, C.Y. Lin, Optimal design of truss structures using particle swarm optimization. Comput. Struct. **89**, 2221–2232 (2011)
15. E. Dogan, M.P. Saka, Optimum design of unbraced steel frames to LRFD–AISC using particle swarm optimization. Adv. Eng. Softw. **46**, 27–34 (2012)
16. B.A. Nedushan, H. Varaee, Minimum cost design of slabs using particle swarm optimization with time varying acceleration coefficients. World Appl. Sci. J. **13**(12), 2484–2494 (2011)
17. M. Khajehzadeh, M.R. Taha, A. El-Shafie, M. Eslami, Economic design of retaining wall using particle swarm optimization with passive congregation. Aust. J. Basic Appl. Sci. **4**(11), 5500–5507 (2010)
18. S. Gharehbaghi, M.J. Fadaee, Design optimization of RC frames under earthquake loads. Int. J. Optim. Civ. Eng. **2**(4), 459–477 (2012)
19. J. Kennedy, R.C. Eberhart, Particle swarm optimization, in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
20. I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Process. Lett. **85**, 317–325 (2003)
21. Y.D. Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, R.G. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans. Evol. Comput. **12**(2), 171–195 (2008)
22. A. Kaveh, O. Sabji, Optimum design of reinforced frames using a heuristic particle swarm—ant colony optimization, in *Proceedings of Second International Conference on Soft Computing Technology* (Civil-comp Press, Stirbngshire, 2011)
23. M.J. Esfandiary, S. Sheikholarefin, H.A.R. Bondarabadi, A combination of particle swarm optimization and multi-criterion decision-making for optimum design of reinforced concrete frames. Int. J. Optim. Civ. Eng. **6**(2), 245–268 (2016)
24. IS: 456:2000, *Code of Practice for Plain and Reinforced Concrete* (fourth revision) (Bureau of Indian Standards, New Delhi)
25. P.C. Vergese, *Limit State Design of Reinforced Concrete* (PHI Learning Private Limited, Delhi, 2013)