Minimum Cost Design of Concrete Slabs using Particle Swarm Optimization with time Varying Acceleration Coefficients

Behrouz Ahmadi-Nedushan and Hesam Varaee

Department of Civil Engineering, Faculty of Engineering, Yazd University,
Pejoohesh Street Safa-ieh, P.O. Box: 89195-741, Yazd, Iran

Abstract: This article deals with cost optimization of one-way concrete slabs according to the most recent American Concrete Institute code of practice (ACI 318-M08). The objective is to minimize the total cost of the slab including costs of concrete and reinforcement bars while satisfying all the design requirements. Particle Swarm Optimization (PSO) is used for solving the constrained optimization problem. As PSO is designed for unconstrained optimization problems, a multi-stage dynamic penalty was also implemented to solve the constrained optimization problem. Cost optimization of four different slabs with different support conditions are illustrated and the results of the optimum design results are compared with existing methods in the literature. A sensitivity analysis of optimal designs was also performed by optimizing the four examples for different span lengths between 2 to 5 meters to investigate the effect of span length on optimal costs and optimal reinforcement ratios. The results demonstrate that PSO is a promising method in design optimization of structural elements.

Key words: Concrete slab · optimization · penalty function · particle swarm optimization · sensitivity analysis

INTRODUCTION

Design is an iterative process. The designer's experience, intuition and ingenuity are required in the design of systems in most fields of engineering (aerospace, automotive, civil, chemical, industrial, electrical, mechanical, hydraulic and transportation). Iterative implies that several trial designs are analyzed one after another until an acceptable design is obtained. In the design process, the designer estimates a trial design of the system based on experience, intuition, or some mathematical analysis. The trial design is analyzed to determine if it is satisfies all the requirements. If it is, the design process is terminated and the design is accepted as final design [1].

In conventional design of structural systems and elements, most procedures first the adopt the cross-section dimensions and material grades based on common practice. Once the structure is defined, the structure is analyzed to determine the stress resultants. For concrete structure, the procedure is followed by the computation of reinforcement that satisfy the limit states prescribed by concrete codes. Should the dimensions or material grades be insufficient, the structure is redefined on a trial and error basis [2].

Engineers are faced with challenge of designing efficient and cost-effective systems without compromising the integrity of the system. The conventional design process depends on the designer's intuition, experience and skill. This presence of a human element can sometimes lead to erroneous results in the synthesis of complex systems. Furthermore, the conventional design process can lead to uneconomical designs and can involve a lot of calendar time. The conventional design leads to safe designs, but the economy of the design is very much linked to experience of the structural designer. Scarcity and the need for efficiency in today's competitive world have forced engineers to evince greater interest in economical and optimized designs.

Another design method, which is more systematic, is the optimization design process, in which the trial design is analyzed to determine if it is the best. Depending on the specifications, "best" can have different connotations for different systems. In general, it implies cost-effective, efficient, reliable and durable systems. The optimum design process forces the designer to use rigorous formulation of the design problem by defining explicitly a set of design variables, an objective function to be optimized and the imposed constraint functions [1].

Corresponding Author: Dr. Behrouz Ahmadi-Nedushan, Department of Civil Engineering, Faculty of Engineering, Yazd University, Pejoohesh Street Safa-ieh, P.O.BOX 89195-741, Yazd, Iran

A large number of articles have been published on optimization of structures. In the great majority, the objective is to minimize the weight of the structure [3, 4]. While weight of a structure constitutes a significant part of the cost, a minimum weight design is not necessarily the minimum cost design.

Especially, in optimal design reinforce concrete elements and structures, the minimum weight design is not always the cheapest design. Minimizing the cost is more realistic and a more useful goal for saving natural resources.

For a more widespread use of structural optimization algorithms in design office practice, they must be formulated as cost optimization and applied to realistic structures subjected to the actual constraints of commonly used design codes such as the American Concrete Institute Code (ACI 318-M08) [5].

State-of-the-art reviews of articles on cost optimization of concrete structures are presented in [6]. Recent examples of cost optimization of real life structures subjected to design code constraints can be found in [7-13].

The early work on optimization of Reinforced Concrete (RC) slabs was based on many simplifying assumptions. Brown (1975) formulates the optimum cost design of one-way concrete slabs as a single-variable optimization problem to find the optimum thickness of one-way slabs for uniformly loaded simply supported slabs considering flexural deformations only and making other simplifying assumptions [14].

Brondum-Nielsen (1985) presents a method for minimizing the cost of reinforcement in reinforced concrete shells, folded plates, walls and slabs by minimizing the summation of the forces in the steel reinforcement in two perpendicular directions. An academic example is presented without considering any code of practice in defining the constraints [15].

Hanna and Senouci (1995) present an iterative procedure for minimum cost design of all-wood concrete slab forms. Four design variables: type and thickness of sheathing, size and spacing of joists, type and size of stringers and the type of the wood shore were considered in the formulation. No formal mathematical optimization is used. They report cost savings as high as about 10% compared with traditional methods based on the use of charts and tables [16].

Tabatabai and Mosalam (2001) integrated three computer programs: a commercial program for nonlinear analysis of structures, a program for finite element discretization of reinforced concrete structures and minimization of its reinforcement content and a program for data communication between the first two programs for determination of the optimum reinforcement in beams and one-way slabs. The results

for steel dimensioning of a one-way slab, simply supported on one side and fixed on the other side are reported [17].

Recently, few articles based on concrete codes of practice (i.e ACI) have been published. Sahab *et al.* (2005a, 2005b) presented a hybrid method based on genetic algorithm for cost optimization of reinforced concrete flat slab buildings according to the British Code of Practice (BS8110) [11, 12]. Ahmadkhanlou and Adeli (2005) used a neural dynamics model for optimal cost design of reinforced concrete slabs according to ACI, 1999 code privisions. They formulated the optimization problem as a mixed integer-discrete variable optimization problem with three design variables: thickness of slab, steel bar diameter and bar spacing [10].

In this article, optimal cost design of one way slabs based on the most recent ACI code (ACI 318-M08) in investigated. Particle swarm optimization, one of the most recent evolutionary algorithms is used for this purpose. The constrained optimization problem is tackled through the minimization of a non-stationary multi-stage assignment penalty function. Cost optimization of four different slabs with different support conditions are illustrated and the results of the optimum design results are compared with existing methods in the literature. A sensitivity analysis of optimized designs was also performed by optimizing the four examples for different span lengths.

The reminder of the article is organized as follows: The problem formulation is given in Section 2. Section 3 describes the PSO algorithm and its discrete version. Section 4 contains four illustrative examples of one-way slabs with different support conditions and in Section 5 the summary and conclusions are presented.

FORMULATION OF THE MINIMUM COST DESIGN

Cost function: In concrete structures, at least three different cost items should be considered in optimization: costs of concrete, steel and the formwork. The total cost function can be defined as

$$C_t = C_c + C_f + C_f \qquad (1)$$

where C_C, C_f and C_f are the costs of concrete, reinforcement bars and formwork and finishing materials, respectively. The formwork cost does not vary significantly for any given locality and consequently can be dropped from the formulation [10]. The concrete cost is defined by:

$$C_c = LbhC_c^1 \qquad (2)$$

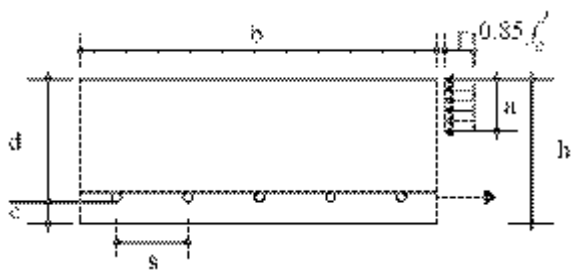Fig. 1: Typical cross-section of RC slab

Table 1: Moment coefficient of continuous slabs

Exterior span			Interior span		
Support	Middle	Support	Support	Middle	Support
-1/24	+1/14	-1/10	-1/11	+1/16	-1/11

Table 2: Maximum moment coefficient, k, used for design of RC slabs

Simply supported	One end continuous	Both ends continuous	Cantilever
1/8	1/10	1/11	1/2

where L, b, h and C_c^1 are the span length, span width, thickness of slab (Fig. 1) and cost of concrete per unit volume, respectively. The reinforcement cost is computed as:

$$C_r = w_s L A_s C_r^1 \qquad (3)$$

where w_s, A_s and C_r^1 are the unit weight of steel, cross sectional area of reinforcement bars and cost of reinforcement bars per unit weight. A_s is calculated by:

$$A_s = \frac{\pi d_b^2}{4} \left(\frac{b}{s} \right) \qquad (4)$$

where d_b and s are diameter of reinforcement bars and their spacing, respectively.

Constraints: All the design constraints imposed by the most recent ACI code (ACI 318-M08) are considered. The constraints include flexural constraints, shear constraints, serviceability constraints and deflection constraints.

Flexural constraint: Flexural resistance of the slab must be greater the implied moments by the loads. This constraint is presented in the following form:

$$M_u \leq \phi M_n \qquad (5)$$

where M_u and M_n are ultimate design moment and nominal bending moment, respectively. The strength reduction factor f is calculated based on net tensile strain in in the extreme tension steel at nominal strength and is varied in the range of 0.65 to 0.9 for compression controlled or tension controlled state
The ultimate design moment is calculated as follows:

$$M_u = kwl_n^2 \qquad (6)$$

where l_n is the clear span length and k is a moment coefficient that depends on the type of slab supports for continuous slabs as presented in Table 1. In Eq. (6), the

maximum value of the moment coefficient for any given span is used, for four different support conditions: simply-supported, continuous in one end and simply-supported at the other end, continuous at both ends and cantilever (Table 2). In Eq. (6), w the factored uniformly distributed load including the dead load and the self-weight of slab is calculated as:

$$w = 1.2 \times (DL \times b + DL_s) + 1.6 \times LL \times b \qquad (7)$$

where DL and LL are dead load of floor excluding the self-weight of slab and the live load, respectively. DL_s represents the self-weight of slab and is equal to:

$$DL_s = (bh - A_s) w_c + A_s w_s \qquad (8)$$

where w_c is the weight of concrete per unit volume.

For design, the ACI code allows the use of an equivalent rectangular compressive stress distribution (stress block) to replace the more exact concrete stress distribution. In the equivalent rectangular stress block, an average stress of 0.85f'$_c$ is used with a rectangle of depth a = β_1c where β_1 is defined as follows:

$$\text{for } f_c^1 \leq 28 \text{ MPa}, \beta_1 = 0.85$$

$$\text{for } f_c^1 > 28 \text{ MPa}, \beta_1 = 0.85 - 0.05 \left(\frac{f_c^1 - 28}{7} \right) \geq 0.65 \qquad (9)$$

The nominal bending moment, M_n, is then calculated as follows:

$$M_n = A_s f_y \left(d - \frac{a}{2} \right) \qquad (10)$$

where f_y is the specified yield strength of reinforcement bars and a is the equivalent depth of the concrete compression stress block calculated from (Fig. 1)

$$a = \frac{A_s f_y}{0.85 f_c^1 b} = \frac{c}{\beta_1} \qquad (11)$$

Table 3: Minimum thickness of solid one-way slabs according to ACI

	Minimum thickness, h			
	Simply supported	One end continuous	Both end continuous	Cantilever
Member	Members not supporting or attached to partitions or other construction likely to be damaged by large deflections			
Solid one way slabs	L/20	L/24	L/28	L/10

Notes: Values given shall be used directly for members with normal weight concrete and grade 420 reinforcement. for other conditions, the values shall be modified as follows:

For lightweight Concrete having equilibrium density, w_c, In the range of 1440 to 1840 kg/m^3, the values shall be modified by Eq. (20) but not less than 1.09

For f_y other than 420 MPa, the values shall be multiplied by Eq. (21)

where f'_c is the specified compressive strength of concrete.

Shear constraint: The shear constraint is represnted in the following form:

$$V_u \leq \varnothing V_c, \varnothing = 0.75 \tag{12}$$

where V_u and V_c are the ultimate factored shear force and the nominal shear strength of concrete. The shear is carried entirely by concrete since no stirrup is used. The ultimate factored shear force and the nominal shear strength of concrete are given by:

$$V_u = k_v \frac{w \, l_n}{2} \tag{13}$$

$$V_c = 0.17 \sqrt{f'_c} \, bd \tag{14}$$

k_v is a number specified by ACI with values of 0.5, 0.575, 0.575 and 1.0 for examples 1 to 4 respectively.

Serviceability constraints: The serviceability constraints are presented in terms of limits on the steel reinforcement ratio and the bar spacing.

ACI 318-M08 requires that the net tensile strain e_t at nominal strength of non-prestressed flexural members and non-prestressed members with factored axial compressive load less than 0.10fcAg, shall not be less than 0.004.

$$\varepsilon_t = 0.003 \frac{\frac{a}{0.85}}{d - \frac{a}{0.85}} \geq 0.004, \quad a = A_s f_y / 0.85 f'_c b \tag{15}$$

ACI also specifies a minimum amount of reinforcement and requires that for structural slabs, minimum As in the direction of the span shall provide at least the following ratios of reinforcement area to gross concrete area, but not less than 0.0014:

(a) Slabs where Grade 280 or 350 deformed bars are used 0.0020
(b) Slabs where Grade 420 deformed bars or welded wire reinforcement are used..0.0018
(c) Slabs where reinforcement with yield stress exceeding 420 MPa measured at a yield strain of 0.35 percent is used 0.0018 × 420/ fy

In slabs, primary flexural reinforcement shall not be spaced farther apart than three times the slab thickness, nor farther apart than 450 mm.

$$s < \min(3h, 450 \text{ mm}) \tag{16}$$

ACI 318-M08 also specifies that the minimum clear spacing between parallel bars in a layer shall be d_b, but not less than 25 mm.

$$s \geq \max(d_b, 25 \text{ mm}) + d_b \tag{17}$$

Deflection constraints: The ACI 318-M08 specifies a minimum slab thickness (h_{min}) of L/20, L/24, L/28, or L/10 for different support conditions (Table 3), with an absolute minimum thickness of 1.5 inch (38.1 mm). In order to take into account the effect of the weight of concrete and the reinforcement yield strength, the numbers in Table 3 must be multiplied by the following modification factors:

$$\alpha_1 = \max(1.65 - 0.0003 w_c, 1.09) \tag{18}$$

$$\alpha_2 = \left(0.4 + \frac{f_y}{700}\right) \tag{19}$$

Constraint normalization: In numerical calculations, it is desirable to normalize all the constraint functions [1]. This normalization speeds up the convergence and prevents undue dominance of any particular constraint since different constraints involve different orders of magnitude. The normalized constraints are introduced by the following equations:

$$g_1 = \frac{kwl_n^2}{\varnothing\left(A_s f_y\left(d - \frac{(A_s f_y / 0.85 f_c' b)}{2}\right)\right)} - 1 \leq 0 \qquad (20)$$

$$g_2 = \frac{V_U}{0.75 \times \left(0.17 \sqrt{f_c'} bd\right)} - 1 \leq 0 \qquad (21)$$

$$g_3 = \frac{0.004}{\varepsilon_t} - 1 \leq 0 \qquad (22)$$

$$g_4 = 1 - \frac{A_s}{A_{s\,min}} \leq 0 \qquad (23)$$

$$g_5 = \frac{s}{min(3h, 450\ mm)} - 1 \leq 0 \qquad (24)$$

$$g_6 = 1 - \frac{s}{max(d_b, 25\ mm) + d_b} \leq 0 \qquad (25)$$

$$g_7 = 1 - \frac{h}{h_{min}} \leq 0 \qquad (26)$$

$$g_8 = 1 - \frac{d_b}{d_{b\,min}} \leq 0 \qquad (27)$$

OPTIMIZATION PROBLEM FORMULATION AND SOLUTION APPROACH

Discrete optimization formulation: First, a continuous variable optimization problem is defined as:

$$minimize\ f(x) = C_t$$

subjected to the following inequality constraints:

$$g_i(x) \leq 0, i=1,\ldots,m \qquad (28)$$

where x is the vector of continuous design variables, f(x) is the cost function defined by Eq. (1) and m is the number of inequality constraints (equal to 8 in this problem, as defined by Eqs. 20-27). Three design variable are considered in the problem definition: the thickness of slab (h), the diameter of reinforcement bars (d_b) and the spacing of reinforcement bars (s). The thickness of the concrete slab and spacing of the reinforcement bars can be considered as discrete variables, considering the common practice of using multiple integers of centimeters in the SI system, or a multiple of 1/8 in or 1/4 in inch the US customary system. The bar diameters are treated as discrete variables as their values must be assigned from a limited number of commercially available bar sizes.

ACI provides eleven different bar sizes starting from bar size #3 with diameter of 0.375 inch (0.953 cm) to bar size #18 with diameter of 2.257 inch (5.733 cm).

Therefore, the problem is formulated as a discrete nonlinear programming problem expressed as:

$$minimize\ f(x)$$

subjected to the following constraints:

$$g_i(x) \leq 0, \quad i=1,\ldots,m$$

$$x_i \in D_i = \{d_{i1},\ldots,d_{in}\}, \quad i=1,\ldots,n_d \qquad (29)$$

where n_d is the total number of discrete design variables (equal to three in this article) and D_i is the set of discrete values for the ith variable.

Particle swarm optimization: The PSO algorithm was first proposed in 1995 by Kennedy and Eberhart. It is based on the premise that social sharing of information among members of a species offers an evolutionary advantage [18]. Recently, the PSO has been applied and proven useful on a few structural engineering applications such as optimal truss design [19, 20], structural damage detection [21] among others. A number of advantages with respect to other Evolutionary algorithms make PSO an ideal candidate for engineering optimization problems. The algorithm is robust and well suited to handle non-linear, non-convex design spaces with discontinuities. Furthermore, its easiness of implementation makes it more attractive as it does not require specific domain knowledge information, internal transformation of variables or other manipulations to handle constraints [22].

In PSO, a number of simple entities (the particles) are placed in the search space of some problem or function and each particle evaluates the objective function at its current location. Each particle then determines its movement through the search space by combining some aspect of the history of its own current and best (best-fitness) locations with those of one or more members of the swarm, with some random perturbations. The next iteration takes place after all particles have been re-located. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function.

Each individual in the particle swarm is composed of three D-dimensional vectors: the current position $\vec{x}_i$, the previous best position $\vec{p}_i$ and the velocity $\vec{v}_i$ where D is the dimensionality of the search space (i.e. number of design variables).

The current position $\vec{x}_i$ can be considered as a set of coordinates describing a point in space. On each iteration of the algorithm, the current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, $\vec{p}_i$. The value of the best function result so far is stored in a variable often called pbest$_i$ (for "previous best"), for comparison on later iterations. The objective is to keep finding better positions and updating $\vec{p}_i$ and pbest$_i$. New points are chosen by adding $\vec{v}_l$ coordinates to $\vec{x}_l$ and the algorithm operates by adjusting $\vec{v}_l$, which can effectively be seen as a step size [23]. The velocity of each particle is iteratively adjusted so that the particle performs a stochastically oscillation around $\vec{p}_l$ and $\vec{p}_g$ locations. The new velocity of each particle is calculated as follows:

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 r_{1i}^d(t)\left(p_i^d(t) - x_i^d(t)\right) + c_2 r_{2i}^d(t)\left(p_g^d(t) - x_i^d(t)\right)$$

$$(32)$$

where $v_i^d(t)$ represents the current velocity of a design variable. The superscript d stands for the dth particle, the subscript i indicates the ith design variable and t is the iteration number. c_1 and c_2 are two positive constants called acceleration coefficients, ω is the inertia factor and r_1 and r_2 are two independent random numbers uniformly distributed in the range of [0, 1]. After the velocity is updated, the new position of each particle for the next generation is calculated according to the following equation:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$$

$$(33)$$

The particle is then evaluated according to its new position and $\vec{p}_l$ and $\vec{p}_g$ are updated at each generation. This process is repeated until a user-defined stopping criterion is reached.

The original PSO algorithm is designed for optimization problems with continuous variables. As design variables of slab optimization problems are discrete, we used An approach for tackling discrete optimization problems by PSO which is based on the truncation of the real values to their nearest integer [24, 25].

The original PSO algorithm lacks exploitation and is generally slow at late stages of optimization. An improved version of PSO algorithms is adapted in this article. The objective of this modification is to try to avoid premature convergence in the early stages of the search and to facilitate convergence to the global optima in the final stages of the search. An annealing scheme was used for the setting of the parameter ω, where ω decreases linearly from ω_i to ω_f over the whole run

$$\omega = (\omega_f - \omega_l)\frac{iter}{MAXITER} + \omega_l$$

$$(34)$$

where ω_i and ω_f are the values of parameter ω at the start and end of the search., iter is the current iteration number and MAXITER is the number of maximum allowable iterations.

The constraint handling approach: Different constraint-handling techniques have been used over the years to handle linear and nonlinear inequality constraints in evolutionary algorithms. An excellent survey on constraint handling techniques is written by Coello [26].

The search space in constrained optimization problems consists of feasible and infeasible points. In feasible points all the constraints are met. In contrast, in infeasible points at least one of constraints is violated. The most common constraint-handling approach is the use of a penalty function for penalizing infeasible points. In this approach, the constrained problem is transformed to an unconstrained one, by penalizing the infeasible points and building a single objective function, which in turn is minimized using an unconstrained optimization algorithm

Penalty functions can be categorized into two main divisions: stationary and non-stationary. Stationary or static penalty functions use fixed penalty values throughout the minimization, where in contrast, in non-stationary penalty functions, the penalty values are dynamically modified. In the literature, results obtained using non-stationary penalty functions are almost always superior to those obtained through stationary functions [27, 28].

A penalty function can be defined as:

$$F(x) = f(x) + h(k)H(x), \ x \in S \subset R^n$$

$$(35)$$

where $f(x)$ is the original objective function; $h(k)$ is a dynamically modified penalty value, k is the algorithm current iteration number; and $H(k)$ is a penalty factor, defined as:

$$H(x) = \sum_{i=1}^{m} \theta(q_i(x)) \, q_i(x)^{\gamma(q_i(x))}$$

$$(36)$$

where

$$q_i(x) = \{0, g_i(x)\}, \ i = 1, \ldots, m$$

The 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.

RESULTS

Illustrative design examples: Four examples of one-way reinforced concrete slabs with different support conditions are optimized in this section. Example 1 is a simply supported slab at both ends. Example 2 has a simple support at one end and a continuous support at the other end. Example 3 it is part of a multi-space reinforced concrete slabs and therefore can be considered continuous at both ends (). Example 4 is a one-way cantilever reinforced concrete slab.

The common data for the examples are presented in Table 4. These examples were previously optimized with neural dynamic model using an earlier version of ACI code (1999) edition as the design code [10]. The cost of reinforcement steel, C_r^1 is set to \$1.43/Kg. The cost of concrete is a function of concrete strength as noted in Table 5, per Means [29]. For variables h and s, practical values are assumed to be a multiple of 1/4 and 1/2 inch respectively.

Penalty parameters of equation 34 are the same values reported in [27]. Specifically, if $q(x)<1$, then $\gamma((q_i(x)) = 1$, otherwise $\gamma(q_i(x)) = 2$. Moreover, if $q_i(x))<0.001$ then $\theta(q_i(x)) = 10$, else, if $0.001<q_i(x)<0.1$ then $\theta(q_i(x)) = 20$, else, if $q(x)<1$ then $\theta(q_i(x)) = 100$, otherwise $\theta(q_i(x)) = 300$. h(k), the dynamically modified penalty value in equation 36, was set to $k\sqrt{k}$

Following parameters were used in the PSO: c1 = c2 = 2. An annealing scheme was used for the ω-setting of the PSO, where ω decreases linearly from $\omega_i = 0.9$ to $\omega_f = 0.4$ over the whole run. The size of the swarm was set equal to 30 The PSO algorithm ran for 1000 iterations for each example and 30 runs were performed for each example problem.

The optimum values for these examples and associated costs are shown in Table 6. Results of neural dynamic model on the same examples using an earlier version of ACI code are also presented. The direct comparison of results is difficult as the optimal designs are based of different ACI code versions; however, it can be noted that the values obtained by PSO algorithm are generally very similar to neural dynamics for all examples

It can be seen that example 4 (cantilever slab) has the maximum cost (59.31\$) among all examples. This is expected as the required minimum thickness for deflection control of cantilever slab is 1/10 of the span length which is much higher than minimum thickness requirements of other examples: 1/20, 1/24 and 1.28 of span lengths for examples 1 to 3 respectively (Table 3).

Table 4: Common data used in design examples

f_y	40 ksi (275.8 Mpa)	b	1 ft (0.3048 m)
w_s	490 Ib/ft^3 (77 KN/m^3)	L	13 ft (3.96 m)
f'_c	3 ksi (20.68 Mpa)	DL	10 Ib/ft^2 (0.48 KN/m^2)
w_c	150 Ib/ft^3 (23.6 KN/m^3)	LL	40 Ib/ft^2 (2.39 KN/m^2)
Cover	3/4 in (19.05 mm)	C_r^1	\$1300/Ton (short)(\$1.43/kg)

Table 5: Concrete cost, C_c^1, according to its specified compressive strength

f'_c (psi)	C_c^1 (\$/cyb)
2000 (14 Mpa)	71.5
2500 (17 Mpa)	64.0
3000 (21 Mpa)	76.0
3500 (24 Mpa)	78.0
4000 (28 Mpa)	81.5
4500 (31 Mpa)	83.0
5000 (34 Mpa)	84.5
6000 (41 Mpa)	96.5
8000 (55 Mpa)	158.0
10000 (17 Mpa)	224.0

1.0 \$/cyb = 0.122 \$/m^3

It should be noted that ultimate bending moments is also much higher in example 4 (Table 2).

The values of normalized constraint functions for four examples are presented in Table 7. It can be noted that all constraint function values are negative and therefore all constraints are met at optimal points. In all examples, the constraints representing moment capacity (g_1) and minimum slab thickness (g_7) are active. For examples, 2 and 3, (g_8) is also active as a minimum bar size reinforcement is used in the optimal designs. It should be noted that small deviations of active constraints from zero are due to existence of discrete variables.

Parametric study: This section describes the parametric study of one-way slabs with different support conditions for various practical span lengths. The goal of the parametric study was to investigate the effect of slab span length on optimal design variables, the cost components and optimal reinforcement ratio ρ, ratio of A_s to bd. Four previously described examples are optimized with different span lengths, ranging from 7 to 16 ft (2 to 5 meters), using the same procedure previously mentioned.

Results of corresponding optimal values of design variables, concrete and reinforcement costs and total slab cost are presented at Table 8. Total cost obtained for four examples for different span lengths are presented in Fig. 2. Cantilever slab (example 4) has the

Table 6: Cost optimization results for examples 1-4 (length = 13 ft)

	Ahmadkhanlou & Adeli [10]				This paper			
	h (in)	d_b (in)	s (in)	Total cost ($)	h (in)	d_b (in)	s (in)	Total cost ($)
Example 1	6.75	3/8	6.5	26.45	6.25	1/2	9.00	26.57
Example 2	5.75	3/8	7.0	22.98	5.25	3/8	5.50	22.76
Example 3	4.75	3/8	7.0	19.93	4.50	3/8	5.50	20.64
Example 4	13.50	3/8	2.0	60.22	12.50	5/8	12.50	59.31

1 in = 25.4 mm

Table 7: Normalized constraints for four examples (length = 13 ft)

	Normalized constraints							
	g_1 (Eq. 22)	g_2 (Eq. 23)	g_3 (Eq. 24)	g_4 (Eq. 25)	g_5 (Eq. 26)	g_6 (Eq. 27)6	g_7 (Eq. 28)	g_8 (Eq. 29)
Example 1	-0.012	-0.771	-0.889	-0.745	-0.492	-5.064	-0.009	-0.333
Example 2	-0.035	-0.706	-0.875	-0.913	-0.651	-3.046	-0.017	0.000
Example 3	0.000	-0.667	-0.845	-1.231	-0.593	-3.046	-0.017	0.000
Example 4	-0.019	-0.683	-0.853	-1.462	-0.718	-2.105	-0.009	-0.669
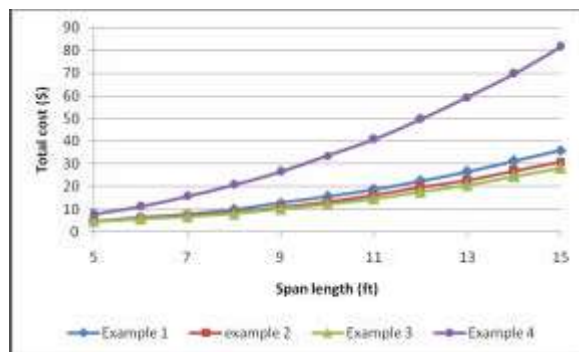
Fig. 2: Optimal total cost of four examples for different span lengths
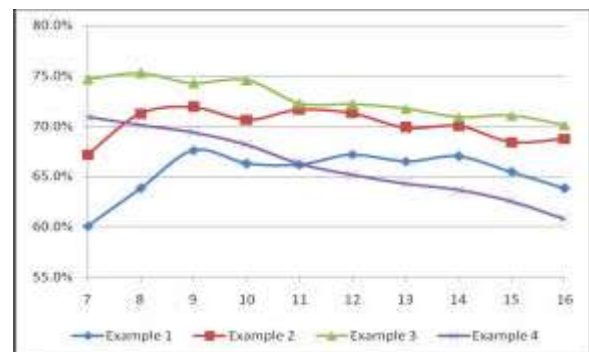
Fig. 3: Optimal reinforcement ratios of four examples for different span lengths

maximum total cost among all examples. Among the remaining examples which are also more common and practical slab configurations (examples 1-3), the simply supported slab has the maximum cost. It can be seen that as expected in all examples the cost increases at higher span lengths.

For different span lengths, optimum reinforcement ratios corresponding to minimum cost designs of four examples are shown in Table 8 and Fig. 3. Optimum ratios for example 1 are very close to 0.4%. verying between 0.38% and 0.44%. For example 2, the optimum ratios vary between 0.44% and 0.63%. It can be observed that for most span lengths (9 ft to 16 ft), the ratios are in the tight range of 0.44% to 0.48%. The optimum reinforcement ratios of example 3 vary between 0.55% and 0.94%. For most span lengths (9-16 ft), the ratios are in the very tight range of 0.55% to 0.60%. For the cantilever slab (example 4), the optimum reinforcement ratios vary between 0.44% and 0.63%. It should be noted

that cantilever slabs are rarely used in high span lengths and for most practical span lengths (7-10 ft), the optimum ratios are in the very tight range of 0.43% to 0.47%.

Adeli (2001) used a neural dynamics model for presented examples. ℏ their work, they obtained the solution in two stages. In the first stage, the neural dynamics model was used to obtain an optimum solution assuming continuous variables. In order to find practical discrete values for the design variables, the second stage of optimization was performed by formulating the problem as a mixed integer-discrete optimization problem.

The main goal of this article has been to present a simple algorithm that can be efficiently used in optimal design of engineering problems. In the approach presented in this article, the optimal values of design variables are obtained directly and in a single step, which in turn simplify the design for practicing engineers.

Table 8: Cost optimization results for examples 1-4 for different lengths (Span width = 1 ft)

	Span length (ft)	h (in)	s (in)	d_b (in)	Cost of concrete (C_C)	Cost of reinforcement (C_r)	Total cost ($)	Reinforcement Ratio (%)
Ex.1	7	3.50	10.5	3/8	5.75	1.95	7.70	0.41%
	8	4.00	9.5	3/8	7.51	2.46	9.97	0.38%
	9	4.50	8.0	3/8	9.50	3.29	12.79	0.39%
	10	5.00	13.0	1/2	11.73	4.00	15.73	0.38%
	11	5.25	11.0	1/2	13.55	5.20	18.75	0.42%
	12	5.75	10.0	1/2	16.19	6.24	22.43	0.41%
	13	6.25	9.0	1/2	19.06	7.51	26.57	0.42%
	14	6.75	8.0	1/2	22.17	9.10	31.27	0.43%
	15	7.25	7.5	1/2	25.51	10.40	35.91	0.42%
	16	7.75	10.5	5/8	29.09	12.42	41.51	0.44%
Ex 2	7	3.00	8.5	3/8	4.93	2.41	7.34	0.63%
	8	3.25	9.5	3/8	6.10	2.46	8.56	0.50%
	9	3.75	8.5	3/8	7.92	3.10	11.01	0.46%
	10	4.00	7.5	3/8	9.38	3.90	13.28	0.48%
	11	4.50	7.0	3/8	11.61	4.60	16.21	0.44%
	12	5.00	11.0	1/2	14.07	5.67	19.75	0.45%
	13	5.25	5.5	3/8	16.01	6.91	22.92	0.47%
	14	5.75	9.0	1/2	18.88	8.09	26.97	0.46%
	15	6.00	4.5	3/8	21.11	9.75	30.86	0.48%
	16	6.50	7.5	1/2	24.40	11.09	35.49	0.48%
Ex 3	7	2.50	7.5	3/8	4.10	2.73	6.84	0.94%
	8	2.75	8.0	3/8	5.16	2.93	8.09	0.76%
	9	3.25	8.0	3/8	6.86	3.29	10.15	0.60%
	10	3.50	7.0	3/8	8.21	4.18	12.39	0.62%
	11	3.75	6.5	3/8	9.68	4.95	14.63	0.60%
	12	4.25	6.0	3/8	11.96	5.85	17.81	0.56%
	13	4.50	5.5	3/8	13.72	6.91	20.64	0.56%
	14	5.00	9.0	1/2	16.42	8.09	24.51	0.55%
	15	5.25	4.5	3/8	18.47	9.75	28.22	0.57%
	16	5.50	4.0	3/8	20.64	11.70	32.34	0.61%
Ex 4	7	6.75	8.0	1/2	11.08	4.55	15.63	0.43%
	8	7.75	10.5	5/8	14.54	6.21	20.75	0.44%
	9	8.75	9.0	5/8	18.47	8.15	26.62	0.44%
	10	9.75	11.0	3/4	22.87	10.69	33.56	0.47%
	11	10.50	6.5	5/8	27.09	13.79	40.89	0.50%
	12	11.50	11.0	7/8	32.37	17.34	49.71	0.53%
	13	12.50	5.0	5/8	38.12	21.19	59.31	0.54%
	14	13.50	6.5	3/4	44.33	25.34	69.67	0.55%
	15	14.50	4.0	5/8	51.02	30.57	81.59	0.57%
	16	15.25	11.5	9/8	57.23	36.95	94.19	0.63%

SUMMARY AND CONCLUSIONS

This article presents the cost optimization of one-way slabs with different support conditions using the PSO algorithm, one of the most recent evolutionary algorithms. The total cost of the slab was used as the objective function and the ACI-M08 design requirements are used to consider all imposed design constraints including criteria for strength, ductility and serviceability among others. a multi-stage dynamic penalty was also implemented to solve the constrained optimization problem

The optimization results of four different slab configurations demonstrate that PSO is a promising

method in design optimization of one-way reinforce concrete slabs. It should be noted that the presented results are obtained based on prices in U.S. Obviously the results depends of on the relative cost of concrete and reinforcement and therefore are location dependant. However, the presented algorithm can be applied in design offices in any location by using the relevant unit costs of concrete and reinforcement. This, in turn, generally reduces the cost of the construction and saves the natural resources.

REFERENCES

1. Arora, J.S., 2004. Introduction to optimum design. 2nd Edn. Elsevier.

2. González-Vidosa, F., V. Yepes, J. Alcalá, M. Carrera, C. Perea and I. Payá-Zaforteza, 2008. Optimization of Reinforced Concrete Structures by Simulated Annealing. In: Cher Ming Tan. Editor. Simulated Annealing, Vienna, Austria, pp: 307-320.

3. Pezeshk, S., 1998. Design of framed structures: An integrated non-linear analysis and optimal minimum weight design. International Journal of Numerical Methods in Engineering, 41: 459-471.

4. Chung, T.T. and T.C. Sun, 1994. Optimization for flexural reinforced concrete beams with static nonlinear response. Structural Optimization, 8: 174-180.

5. Adeli, H. and K.C. Sarma, 2006. Cost optimization of structures: Fuzzy logic, genetic algorithms and parallel computing. John Wiley and Sons. Ltd.

6. Sarma, K. and H. Adeli, 1998. Cost optimization of concrete structures. Journal of Structural Engineering, ASCE, 124 (5): 570-578.

7. Sarma, K. and H. Adeli, 2000. Cost optimization of steel structures. Engineering Optimization, 32 (6): 777-802.

8. Sarma, K. and H. Adeli, 2001. Bi-level parallel genetic algorithms for optimization of large steel structures. Compututer-Aided Civil and Infrastructure Engineering, 16 (5): 295-304.

9. Adeli, H. and H. Kim, 2001. Cost optimization of composite floors using the neural dynamics model. Communications in Numerical Methods in Engineering, 17: 771-787.

10. Ahmadkhanlou, F. and H. Adeli, 2005. Optimum cost design of reinforced concrete slabs using neural dynamic model. Engineering Applications of Artificial Intelligence, 18: 65-72.

11. Sahab, M.G., A.F. Ashour and V.V. Toropov, 2005. Cost optimization of reinforced concrete flat slab buildings. Engineering Structures, 27: 313-322.

12. Sahab, M.G., A.F. Ashour and V.V. Toropov, 2005. A hybrid genetic algorithm for reinforced concrete flat slab buildings. Computers and Structures, 83: 551-559.

13. Atabay, S., 2009. Cost optimization of three-dimensional beamless reinforced concrete shear-wall systems via genetic algorithm. Expert Systems and Applications, 36: 3555-3561.

14. Brown, R.H., 1975. Minimum cost selection of one-way slab thickness. Journal of the Structural Division, ASCE, 101: 2585-2590.

15. Brondum-Nielsen, T., 1985. Optimization of reinforcement in shells, folded plates, walls and slabs. Proceedings of American Concrete Institute, 82 (3): 304-309.

16. Hanna, A.S. and A.B. Senouci, 1995. Design optimization of concrete slab forms. Journal of Construction Engineering and Management, ASCE, 121(2): 215-221.

17. Tabatabai, S.M.R. and K.M. Mosalam, 2001. Computational platform for non-linear analysis/optimal design of reinforced concrete structures. Engineering Computations, 18 (5/6): 726-743.

18. Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. In: IEEE international conference on neural networks, Piscataway, NJ, 4: 1942-1948.

19. Perez, R.E. and K. Behdinan, 2007. Particle swarm approach for structural design optimization. Computers and Structures, 85: 1579-1588.

20. Li, L.J., Z.B. Huang and F. Liu, 2009. A heuristic particle swarm optimization method for truss structures with discrete variables. Computers and Structures, 87: 435-443.

21. Begambre, O. and J.E. Laier, 2009. A hybrid Particle Swarm Optimization-Simplex algorithm (PSOS) for structural damage identification. Advances in Engineering Software, 40: 883-891.

22. Perez, R.E. and K. Behdinan, 2007. Particle swarm optimization in structural design. In F.T.S. Chan and M.K. Tiwari Eds. Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Itech Education and Publishing, Vienna, Austria, pp: 373-394

23. Poli, R., J. Kennedy and T. Blackwell, 2007. Particle swarm optimization: An overview. Swarm Intelligence, 1: 33-57.

24. Husseinzadeh Kashan, A. and B. Karimi, 2009. A discrete particle swarm optimization algorithm for scheduling parallel machines. Computers and Industrial Engineering, 56: 216-223.

25. Laskari, E.C., K.E. Parsopoulos and M.N. Vrahatis, 2002. Particle swarm optimization for integer programming. Proceedings of the IEEE Congress on Evolutionary Computation, Honolulu, pp: 1582-1587.

26. Coello, C.A.C., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Computer Methods in Applied Mechanics and Engineering, 191: 1245-1287.

27. Parsopoulos, K.E. and M.N. Verahatis, 2002. Particle swarm optimization method for constrained optimization problems. In Sincak, P., J. Vascak, V. Kvasnicka and J. Pospichal, Eds. Intelligent Technologies-Theory and Application: New Trends in Intelligent Technologies. Frontiers in Artificial Intelligence and Applications, IOS Press, 76: 214-20.

28. Joines, J.A. and C.R. Houck, 1994. On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GA's. Proceeding of IEEE International Conference on Evolutionary Computations, pp: 579-585.

29. Means, Co, 2002. Means Building Construction Cost Data. Kingston, MA.

30. Shi, Y. and R.C. Eberhart, 2001. Fuzzy Adaptive Particle Swarm Optimization. Proceedings of the Congress on Evolutionary Computation, Seoul, Korea, pp: 101-106.