SLAB FORMWORK DESIGN USING GENETIC ALGORITHM

Formwork design using GA

H AL-TABTABAI, A.P. ALEX and R JAMES
Department of Civil Engineering, College of Engineering and Petroleum, University of Kuwait, Safat, Kuwait

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier*. Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2407-2418.
© National Research Council Canada 1999

Abstract

A method to design cost-optimum slab formwork components is proposed in this paper. Genetic Algorithms (GAs), a technique based on the principles of natural selection and evolution, is applied to solve the optimisation problem. GAs search from a population of possible solutions limited by a set of constraints. The cost of form components and labor involved, were considered for the formulation of the objective function of the optimisation problem. The bending moment, shear, maximum deflection, imposed ACI code provisions, etc., were used as constraints for the optimisation problem. Application of GA to the formwork design problem provides optimum design parameters such as the optimum cross section for form members, optimum spacing of form members, etc., while minimising the total cost. Formwork made either from wood, wood-metal composite or metal alone can be designed using the proposed technique. The paper presents the case of general formwork design, however, the method as a whole readily applies to the design of formwok for elevated slabs and high rise concrete elements.

Keywords: Formwork design, genetic algorithm, optimisation, construction cost

1 Introduction

Slab formwork holds and provides support for freshly placed concrete through a framework of sheathing, joists, stringers and shores (Fig. 1). Sheathing is in the form of sheets of plywood, while joists, stringers and shores act like beams and columns. Sheathing retains both the concrete and applied loads while supporting the members comprising of joists, stringers and shores, which holds the sheathing in place. Joists and stringers act horizontally while shores act vertically. Stringers are supported on shores while joists are supported on stringers.

The essential requirements for good formwork are quality, safety and economy (Speigel & Limbrummer, 1992). Formwork should be designed accurately, erected and kept rigid and tightly jointed with proper finish. Economy requires the form to be simple, easy to handle, standardised and reusable. Safety requires the formwork to be strong and sound, as formwork failure is a major cause for accidents during construction.

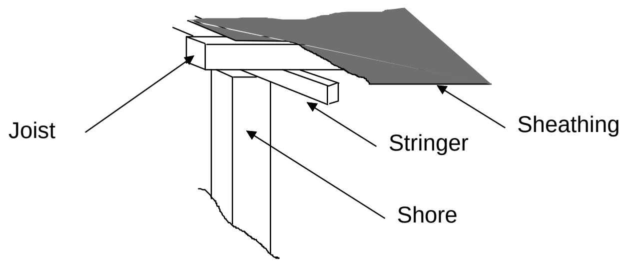

Fig. 1: Parts of a typical slab formwork

The formwork labor and material cost is estimated to be approximately 30% of the concrete slab cost (Senousi & Ansari, 1996) as shown in Fig. 2. The formwork material cost accounts for about 12% of the total formcost (Hanna & Senousi, 1997). Minimising the formwork cost helps to reduce the overall construction cost. In this research, a genetic-algorithm-based optimisation technique is proposed to design the optimum section of joist, stringer and shore along with their spacing.

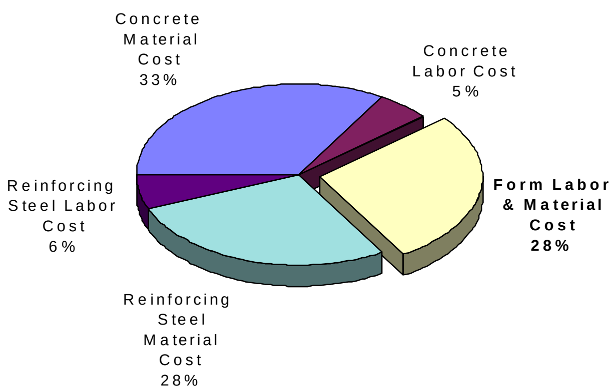

Fig. 2: Typical cost breakdown for concrete slabs

Traditionally slab forms are designed based on developed design tables, taking into consideration the strength of forms to resist the failure and stiffness to avoid deflection. Christian (1987) proposed an integrated microcomputer package for formwork design based on published data obtained from empirical guidelines and recommendations given in Hover (1981) and Peurifoy (1976). Ringwald (1985) proposed a set of design curves for specific wood type, for which safe formwork design values namely size and spacing of members can be inferred while Senousi and Al-Ansari (1996) proposed a computer program for the design of slab formwork components. In these works, the design process does not evaluate all possible solutions to find an optimum formwork design while some of them are even based on empirical grounds. Selecting the sizes and spacing of form members empirically can be very unsatisfactory, uneconomic, and even dangerous. Any under-designing can result in ultimate failure of the forms while over-designing can result in excessive cost. Further, selecting one of the near optimum solutions might not give the complete cost advantage required, as there can be a precise optimum design combination (size and spacing) available somewhere in the solution space.

An Artificial Intelligence (AI) based optimisation procedure to design the formwork components is introduced in this paper. The cost for each of the form components is minimised while maintaining the safety of the formwork. Using this formulation, an optimal solution is evolved by the Genetic Algorithm from a set of points limited by the given constraints.

2 Formwork

Design of construction formwork involves a step-by-step analysis of sheathing and framing members. The sequence of these design steps depends in large measure on the overall plan of the job. Formulas for shear, bending moment and deflection are traditionally used in designing and limiting the spacing of form members. Design should be such that the formwork should support all the loads applied on it without excessive deflection or collapse leading to accidents. A typical structural formwork system that supports the freshly placed concrete for an elevated slab construction is as shown in Figure 1. The procedure used in this paper for the formwork design is based on the standard equations available for structural design contained in the ACI formwork standard, explained further by Hurd (1989).

2.1 Load on the form members

Loads applied on the forms are of two categories: (a) Vertical loads such as the weight of the concrete, the reinforcing steel, the forms themselves as well as the construction live loads, and (b) lateral loads including the externally applied loads and the internally applied loads. The current formwork design specifications are based on allowable stresses developed due to these loads, as specified by Hurd (1989).

Form members are designed to bear the effects of imposed loads. Major slab formwork loads include dead load, live load, wind load and equipment impact load. Normally, weight of ordinary concrete including the reinforcement is taken as 150 lb/ft^3. Weight of formwork usually comes to about 3 - 15 psf which is usually neglected. Live load comprises of the weight of workmen, construction equipment and storage materials along with impact load of 50 psf. Recently, Karshenas and Heinrich (1994) performed an investigation on the impact loads associated with the crane-and-bucket method of concrete placement. They show that, for bigger slabs (30 cm or thicker), designing the formwork for the weight of slab plus 50 psf of impact load is not very safe where the effective impact load is higher than the design loads specified by ACI 347R-88. Pouring 1.5m^3 (2 cu yd) of concrete from 1.5 m height is an extreme situation, but according to a number of field superintendents, it is a probable event. For slab thickness less than 30 cm, (e.g., 20 cm, 12.5 cm, etc.) 50 psf is found to be within the range of design loads specified by ACI 347R-88. For the current problem, 50 psf live load based on a fairly conservative estimate of likely field situations is adopted as an example. A potential user can modify these design limits according to their standards without affecting the optimisation process. Loads in this study are defined as follows:

Design load = Dead load + Live load **(1)**

Dead load of concrete and steel = $\frac{t}{12} \times 150$ psf **(2)**

where "t" is the thickness of the slab.

Neglecting the weight of forms, and taking minimum recommended live load as 50 psf, the total load = $\frac{t}{12} \times 150 + 50$ psf **(3)**

2.2 Stress on form members

Bending members like sheathing, joists and stringers, are considered as uniformly loaded and supported on three or more spans. They are analysed for bending moment, shear and deflection. Shoring is analysed for compressive loads and also for the bearing stresses developed at supports. The traditional stress equations as specified by Hurd (1989) are used for the formulation of constraints.

Wood has the property of being able to support excessive loads for short periods of time. So normal allowable stresses can be increased by 25% as formworks are temporary structures and the loads are of short duration. Further the loads reach a peak during pouring activity and rapidly fall off as concrete hardens. Partially seasoned wood is normally used for the sheathing. Partially seasoned wood has a moisture content of more than 19% and hence allowable stresses must be decreased by a factor of 0.86 for bending, 0.97 for horizontal shear and modulus of elasticity.

3 Design methodology

In a beam subjected to a uniformly distributed load (w) (lb/ft), expressions can be derived for the maximum allowable span length by equating the allowable unit stresses to the maximum unit stresses developed. The moment expressions are for maximum positive or negative moment. The limiting deflection is normally taken as $1/360^{th}$ of the Allowable Span (l) (Hurd, 1989).

For a beam uniformly loaded and continuous over three or more spans, the following expressions are used for the design:

Max bending moment (M) = $\frac{wl^2}{120}$ **(4)**

Resisting bending moment (M) = $f \times S$ **(5)**

where, f = calculated unit stress in bending (psi), w = uniform load (lb/ft),
S = section modulus (in^3),
M = maximum bending moment (in.-lb.), and l = spacing
From (1) through (5),

Maximum spacing (l) = 10.95 $\sqrt{\dfrac{fS}{w}}$ **(6)**

Calculated unit Shear stress (H) = $\dfrac{1.5V}{A}$ **(7)**

Max horizontal shear (V) = 0.6wl **(8)**

From (7) and (8),

Maximum spacing (*l*) = $\dfrac{Hbh}{0.9w}$ **(9)**

Neglecting loads within a distance (h) from supports,

Calculated unit Shear stress (H) = $\dfrac{0.9w}{bh}\left(1-\dfrac{2h}{12}\right)$ **(10)**

Max horizontal shear (V) = 0.6w $\left(1-\dfrac{2h}{12}\right)$ **(11)**

From (10) and (11),

Maximum spacing (*l*) = $12\left(\dfrac{Hbh}{0.9w}+\dfrac{2h}{12}\right)$ **(12)**

Where, V = maximum shear (lb), H = calculated unit stress in shear (psi), b = width of form member, h = breadth of form member

Forms must be designed so that the members will not deflect beyond certain maximum values to avoid the formation of bulges and cracks. Deflection is a governing factor where the depth of the member is relatively small compared to its span. Hence for sheathing, deflection will be the governing factor while for joists and stingers the bending and shear are often the governing factors.

Deflection = $\dfrac{w}{12}\times\dfrac{l^4}{145EI}$ **(13)**

Allowable deflection = l/360 **(14)**

from Eq. (13) and Eq. (14),

Maximum allowable span (*l*) = 1.69 $\sqrt[3]{\dfrac{EI}{w}}$ **(15)**

Where, E = modulus of elasticity (psi), I = moment of inertia (in^4)

Compression (both parallel and perpendicular to the grain) = $f_{comp} = \dfrac{P}{A}$; where, f_{comp} = calculated unit stress in compression (parallel or perpendicular to the grain) (psi), P = concentrated load (lb.), and A = cross-sectional area (in^2).

Consider a strip of sheathing of the specified thickness and 12 inch in width for this illustration. Based on Eq. (1) through Eq. (15), the allowable joist spacing based on bending stress, shear, and deflection consideration can be computed. The lowest of the computed values will finally determine the maximum spacing of the joists. Once the spacing is computed, the required quantity of each form member can be obtained. It should be noted that for the joist design, the values of f, S, E, and I in Eq (3 to 15) are the values that correspond to the sheathing material. Extreme fibre bending values were scaled by a factor 0.86, to account for the moisture content and by 1.25 for load duration consideration for this illustration. Douglas Fir Leech Lumber is assumed to be used for the form members and the corresponding extreme fibre bending value was used for the design computation.

Based on the computed joist spacing, the joist itself is analysed to determine its maximum allowable span. Joists are assumed to be continuous over three or more spans. Each joist must support the load from the sheathing halfway over to the adjacent joist on either side. Therefore the width of the load carried by the joist is equal to the spacing of the joists. The selected joist span becomes the spacing of the stringers. Based on the obtained stringer spacing, the process is repeated to determine the maximum stringer span which is the distance between the vertical supports (shores). Joist loads will be acting on the stringer as a series of concentrated loads but for the sake of simplicity the load is treated as a uniformly distributed load. Once the distance between the vertical supports (shores) is obtained, the load to be carried by each shore should be less than the safe working load available for the shores which can be calculated as the product of shore spacing, stringer spacing, and design load. A steel shore was assumed for this illustration.

3.1 Check for bearing stresses

The bearing stresses produced when one member rests upon another are critical in the design of formwork. The members must have sufficient area of bearing on their supports to prevent crushing of the grain. Otherwise the formwork will settle out of position or have undesirable cracks and openings. These stresses need to be analysed wherever joists rest on stringers, and where stringers rest on shores. The bearing forces applied to the horizontal timber member cause compression perpendicular to grain. The allowable stresses for compression perpendicular to the grain are lower than those for compression parallel to grain. The calculated stringer span is checked against the capacity of the shores used to support the stringers. The load on each shore is computed as the product of shore spacing and the load per foot of the stringer. Thus the maximum shore spacing is limited to the lower span length as governed by the stringer strength or shore strength. Stringer bearing on shore also should be kept below the short term compression perpendicular to grain. Stringer bearing on shore is computed as the ratio between the total load on shore to the bearing area.

It is also necessary to check the bearing at the point where each joist rests on the stringer. This is done by comparing the resulting stress with the allowable unit stress in compression perpendicular to grain. Joist bearing on stringer should be kept below the short-term compression perpendicular to grain. Joist bearing on stringer is computed as the ratio between the total stringer load to the corresponding bearing area.

4 Genetic algorithm

Genetic algorithms (GAs) is a computerised search method based on the ideas of genetics and natural selection proposed by Holland (1975). They use random techniques but exploit information from the past experience to evolve solutions to real world problems, once they are appropriately encoded. GAs have been demonstrated to be robust heuristic search techniques that are capable of rapid identification of optimal design options whilst avoiding convergence on local optima. Even though GAs are not guaranteed to find the global optimum solution to a problem within finite time, they are generally proved valuable at finding near optimal solutions to problems that were previously considered too large or complicated to solve within reasonable amount of time. This adaptive search technique, which has powerful non-linear processing capabilities, can be used for solving multi-dimensional optimisation problems with discrete variables and discontinuous functions. However GAs are computationally expensive and will usually be outperformed when specialised algorithms for a problem exist.

GAs work with a population of individuals each representing a possible solution to a given problem. Each candidate solution is represented as a string of bits (a set of binary/character strings) analogous to chromosomes and genes in evolution theory. GAs assign a fitness score to each individual based on the quality of the solution it represents and highly fit individuals are reproduced by breeding with other individuals. New populations are continuously evolved over generations. During the evolution process the quality of the population increases, hopefully leading to an optimal solution. Finally the population will converge to an optimal solution to the coded problem.

A fitness function is necessary for each problem to be solved by GAs. Fitness is determined by an objective function or subjective judgement. Objective function provides a measure of performance with respect to a particular set of parameters. Fitness function transforms that measure of performance into an allocation of reproductive opportunities. Fitness function is generally used to map individual bit strings into a positive number which is called the individual's fitness. Probability that an individual in the current population is propagated to the future generation is proportional to its fitness. Survival-of-the fittest method adopted by GA selects new strings from the old population randomly, but biased by their fitness.

Reproduction operators of GAs provides a means to weed out the bad and to generate new and better set of solutions during every iteration. Desirable characteristics of the parents are inherited by off springs during the reproduction phase by combining the best characteristics of both parents. The main reproductive operators are: (a) selection, (b) crossover, and (c) mutation. In the selection phase, individuals are selected from the population and recombined to generate new offspring. The objective of selection procedure is to give exponentially increasing trials to the fittest individuals; thereby, less fit individuals are eradicated and individuals of higher quality are selected and rewarded by letting them reproduce more often. Crossover is a reproduction process by which the bit-strings of two fit parent individuals combine to produce two child individuals. Single point crossover is usually performed by swapping the fragments between two parents at a random point along the bit string.

Crossover is generally applied to randomly paired strings with a moderate to high probability denoted P_c(usually the value of P_c falls between 0.6 and 1). Mutation is another reproductive operator that provides a theoretical guarantee that no bit value is ever permanently fixed in all strings. Mutation introduces random modifications and hence induces a random walk through the search space. During mutation, a portion of the new individuals will be flipped to generate a new bit. This is a critical operator that prevents the GAs from being stuck at good but non-optimal solutions. Mutation is done with a low probability denoted by P_m (usually applied with less than 1% probability). Mutation helps to maintain diversity within population and restrain any premature convergence.

The first generation in the GA process is a population of randomly generated individuals. From there on, the genetic operations, in concert with the fitness measure, operate to improve the initial population. Fig. 3 shows the steps involved in a typical GA implementation. Introduction to GA can be found in Austin (1990), Beasley et. al (1993) and Whitley (1993). Tabtabai and Alex (1997, 1998a 1998b 1998c and 1999) present the utility of GA to various construction management optimisation problems.

5 Formwork optimisation procedure using genetic algorithm

The formulation and solution procedures adopted for the formwork optimisation problem in this paper, have the following major steps:

a) Identify the solution structure of the optimisation problem,
b) Code the solution in the form of strings,
c) Define the objective function to be optimised,
d) Define the GA operators and stopping criteria,
e) Evolve solutions until the stopping criteria are met, and
f) Decode the evolved solution strings into the optimum solution.

Figure 3 illustrates the complete GA application methodology, which is analogous to the evolution strategy found in nature.

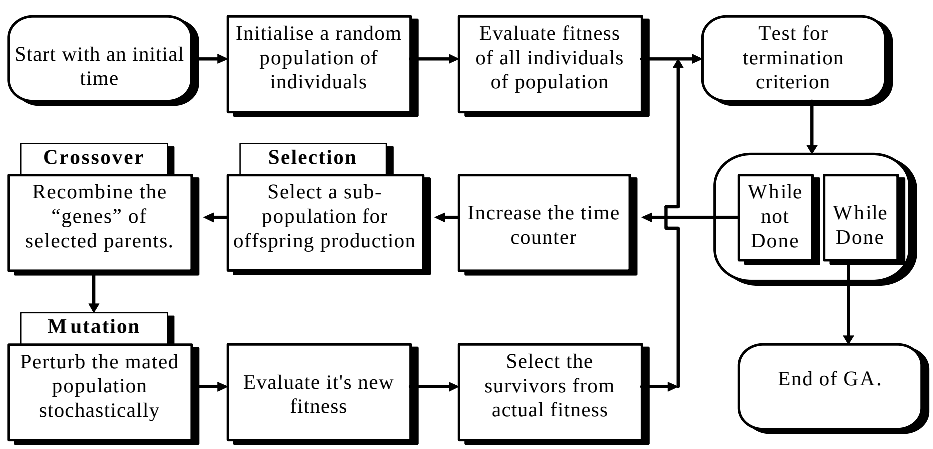

Fig. 3: A typical genetic algorithm optimization process

The objective function for the optimisation problem can be formulated as:

$C= N_1 \times C_1 + N_2 \times C_2 + N_3 \times C_3 + N_4 \times C_4 + A \times t \times C_5$ (16)

in which,

N1= No. of sheathing,	C1 = Unit cost of sheathing,
N2 = No. of joist,	C2 = Unit cost of joist,
N3 = No. of stringers,	C3 = Unit cost of stringers,
N4 = No. of shores,	C4 = Unit cost of shores,
A = Area of slab,	C5= Unit labor cost for unit volume of concrete
t = thickness of slab	

5.1 Implementation

The values for GA control parameters, namely probability of crossover (P_c), probability of mutation (P_m) and number of generations (N_{gen}), were selected based on values reported by other researchers in various optimisation problems. The elitist strategy, which always allows the best string in each generation to survive in the next generation, is employed. GA implementation was performed using GeneHunter™, a commercially available genetic algorithm add-on software from Ward System, Inc., for the Microsoft Excel™ spreadsheet. A microcomputer (IBM compatible Pentium-class, 16 MB Ram, 100 MHz machine), was chosen as the hardware platform to run the evolution process outlined in Fig. 3. This set-up was found to be simple, adequately efficient, and cost effective. Moreover, microcomputers of above minimum configuration are commonly available in most medium to large construction offices.

The design problem is first represented in the spreadsheet. The inputs required from the designer are first represented under various sections namely;

- structural inputs (length of slab (m), thickness of slab (cm), breadth of slab (m)),
- sheathing inputs (sheathing thickness (mm), sheathing width (breadth) (m), sheathing length (m), bending stress (MPa), young's modulus (GPa), shear stress (MPa)),
- joist & stringer inputs (bending stress (MPa), young's modulus (GPa), shear stress (MPa), compression perpendicular to the grain (MPa)), and
- shore inputs (bending stress (MPa), young's modulus (GPa), shear stress (MPa), end plate dimensions (cm), shore capacity (KN)).

Individual cells are designated for the entry of these input variables. Once the input variables are specified in the spreadsheet, a new set of cells are designated for the output variables, namely the breadth of joist, height of joist, breadth of stringer, and height of stringer measured in appropriate units. During the optimisation process, GA considers a set of joist and stringer sizes for each iteration. The evolved shore and stringer sizes are used to compute the allowable spacing using Eq (6), Eq (12), and Eq (15). The minimum of this value is then selected as the spacing for members. The spacing of shore, stringer, and joist are limited by considering the stringer bearing on shore, joist bearing on stringer, and load carrying capacity of each shore. The load carrying capacity of steel shore is assumed as 22.2KN. The expected reuse was taken as 3 for sheathing, joist and stringer and 25 for the steel shores as practised in the local industry. The number of reuses can be judged based on the care exerted by the form handlers during the dismantling, erecting and storage of formwork. The type of form components, unit material costs, unit labor costs, and the potential reuses of the slab form components were assumed in this problem.

During successive evolution, GA attempts to evolve the optimum dimensions for these members that will eventually result in minimum total cost. For this example, a population of 100 individual solutions has been considered during each evolution. From the initial population the worst solutions were discarded while the best solutions were combined with each other by crossover, thus creating a new population. The probability of crossover used for the problem was 0.9 (P_c = 0.9). A 1-point crossover was used for the current problem. Occasionally a gene will be altered to produce a mutation. The probability of mutation adopted for the current problem was 0.01 (P_m = 0.01). Fitness of the current population was determined by the GA and the above steps were repeated on the current population until the subsequent adaptation created a more fit solution. The entire genetic algorithm process continues through many generations until the best solution is good enough to become the required solution decided based on the iteration time and the rate of change of objective function.

The total cost of the formwork was derived from the number and spacing of each individual formwork component used for the construction of the facility. Optimisation process using GA was carried out for various slab sizes. The material properties of each component, the unit cost of each component, deflection limits, and the slab dimensions were required to be input by the user. During the evolution process, the GA evaluates different dimensions of each formwork component, which will satisfy the design constraints, till the combination that gives the minimum cost is reached. Different sizes of joists and stringers have been evaluated by the program to find out the optimum dimensions resulting in overall minimum formwork cost.

For a standard slab carrying standard load with thickness ranging from 10 cm to 15 cm, based on the constrains set, GA yields an optimum section of 5 cm by 5 cm joist and 5 cm by 20 cm stringer. For a slab thickness of 10 cm, 45 cm spacing is found to be adequate for the joist, while stringer and the shore need to be spaced at 58 cm and 220 cm respectively. With the increase in the thickness of slab to 15 cm, the dead load of slab increases and the spacing becomes 43 cm., 55 cm., and 210 cm, for the joist, stringer and shore respectively. When the slab size increases, the joist size remains the same but the stringer size varied to 5 cm by 20 cm. The spacing for the 10 cm thick slabs are 45 cm for joist, 60 cm for stringer, and 200 cm for shores. For the slab of thickness of 15 cm the joist and stringer spacing remains the same but the shore spacing decreases to 190 cm. The computed formwork design parameters were found to be safe and economical. The above problem was presented as an illustration about the use of GA for formwork design. However, a comparison between the design options based on traditional methods and use of practical design constraints will provide more confidence in these results.

6 Summary

The formwork should be designed such that the system is both safe and economical. Improper formwork design is one of the major causes of concrete construction failure. In the traditional formwork design procedure, the section is kept constant by the designer and the spacing is calculated based on bending, shear and deflection limits without specific consideration to the involved cost. The spacing that satisfy the constraints are considered acceptable, which might not necessarily give the optimum minimum cost. A new approach to design the concrete slab formwork using Genetic Algorithm is proposed in this paper. The objective is to design the formwork in a most economical way with maximum functionality.

7 Acknowledgement

The work presented here has been supported by the Kuwait University. This support is gratefully acknowledged.

8 References

Al-Tabtabai, H. and Alex, A. P. (1997). Manpower Schedule Optimisation using Genetic Algorithms, *4th ASCE Congress on Computing in Civil Engineering*, Proc., pp 702-709, June 16-18, Philadelphia, Pennsylvania, USA.

Al-Tabtabai, H. and Alex, A. P. (1998). An evolutionary approach to capital budgeting, *Cost Engineering*, October.

Al-Tabtabai, H. and Alex, A. P.(1998). Resource allocation in construction projects using genetic algorithms, *Kuwait Journal of Science and Engineering*, Vol. 25, No. 1.

Al-Tabtabai, H. and Alex, A. P. (1998) Network Compression using Genetic Algorithm, *Proc., 5th ASCE Congress on Computing in Civil Engineering*, Oct 18-21, Boston, Massachusetts, USA.

Al-Tabtabai, H. and Alex, A. P. (1999) Using genetic algorithms to solve optimisation problems in construction, *Engineering Construction and Architectural Management*, Blackwell Science Ltd. (Accepted for publication).

Austin, S. (1990). An introduction to genetic algorithm. AI Expert, March 48-53.

Beasley, D., Bull, D. R., and Martin, R. R. 1993. An overview of genetic algorithms: Part 1 & II, University Computing, University of Cardiff, UK, 2.

Christian, J. (1987). Use of integrated microcomputer package for formwork design. Journal of Construction Engineering and Management, ASCE, 4: 603-610.

Hanna A.S., and Senousi, A.B. (1997). Material cost minimization of concrete wall forms. Building and Environment 1: 57-67.

Holland, J. (1975). Adaption in natural and artificial systems, The University of Michigan Press, Ann Arbor.

Hover, K. C. (1981). Design and selection consideration for concrete falsework. Journal of the Construction Division, ASCE 3: Sept., 527-541.

Hurd, M.K. (1989). Formwork for concrete, Spec. Publ. No. 4, American Concrete Institute, Detroit, Mich,.

Karshenas, S. and Heinrich, S. M. 1994. Dynamic modeling of slab formwork during concrete placement. Journal of Structural Engineering, 7: 2199-2217.

Peurifoy, R. L. (1976). Formwork for concrete structures, 2nd Ed., McGraw-Hill Book Co., Inc., New York, N.Y.

Ringwald, R.C. (1985). Formwork design. Journal of Construction Engineering and Management, ASCE 4: 391-403.

Senousi, A.B., and Al-Ansari, M.S. (1996). Optimum design of concrete slab forms. Engineering Journal of University of Qatar 9: 79-93.

Speigel, L. and Limbrummer, G.F. (1992). Reinforced Concrete Design, Third Edition, Prentice Hall, Englewood Cliffs, New Jersey.

Whitley, D., (1993). A genetic algorithm tutorial, Tech. Rep. Colorado state university, Fort Collins, Colorado CS-93-103.