Journal of Structural Engineering, ASCE, Vol. 129, No. 10, 2003, pp. 1312-1321

Knowledge-Based System on Optimum Design of Liquid Retaining Structures with Genetic Algorithms

K.W. Chau[1] & F. Albermani[2]

[1]Department of Civil & Structural Engineering, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong
[2]Department of Civil Engineering, University of Queensland, QLD 4072, Australia

Abstract
This paper delineates the development of a prototype hybrid knowledge-based system for the optimum design of liquid retaining structures, OPTLIQ, by coupling the blackboard architecture, an expert system shell VISUAL RULE STUDIO and genetic algorithm (GA). Through custom-built interactive graphical user interfaces under a user-friendly environment, the user is directed throughout the design process, which includes preliminary design, load specification, model generation, finite element analysis, code compliance checking and member sizing optimization. For structural optimization, GA is applied to the minimum cost design of structural systems with discrete reinforced concrete sections. The design of a typical example of the liquid retaining structure is illustrated. The results demonstrate extraordinarily converging speed as near-optimal solutions are acquired after merely exploration of a small portion of the search space. This system can act as a consultant to assist novice designers in the design of liquid retaining structures.

Keywords: genetic algorithm, knowledge-based system, liquid retaining structures, structural optimization

Introduction
The design of liquid retaining structures is specialized and requires assimilation of knowledge from heuristics, research findings and standard engineering methodology. Since these types of structures are exposed in corrosive environment, crack width control is, among others, very crucial in the design. As deviations often exist between the assumed properties of components at the preliminary design stage and their counterparts determined at the detailed design stage, re-analysis will be entailed and iterative steps such as configuration processing, numerical modeling, structural analysis, code conformance checking and sizing optimization are usually involved. It was not easy to code those empirical rules or expert knowledge in a conventional algorithmic or sequential framework. Previous computer-aided design was mainly through loose coupling of individual programs on the captioned sub-processes, which entail intensive knowledge of the structural designer and are prone to human errors during the data transferring processes. As such, an integrated as well as user-friendly system is valuable for storing and employing the expertise.

Recent advances in knowledge-based system (KBS) technologies have rendered it possible to incorporate the heuristic knowledge into the conventional algorithmic structural analysis models. A KBS can be defined as an interactive computer-based decision-making tool that emulates the intensive expert knowledge in a specific domain problem. During the last decade, KBSs have been widely adopted to solve problems in many different disciplines (Chau 1992, Chau et al. 2002, Chau and Ng 1996, Chau and Yang 1992, Chau and Zhang 1995). For application in structural design, the KBS framework is in addition required to couple symbolic processing and extensive numerical processing. Examples of such systems by employing various representation schemes are INDEX (Kumar 1995) and LADOME (Lin and Albermani 2001). However, none of these KBSs appear to have incorporated the recent structural optimization techniques.

In the past, a myriad of mathematical programming algorithms, which usually assume continuous design variables and simple constraints, have been developed for the optimum design of structural systems. Some popular methods, such as calculus-based gradient techniques, entail the construction or approximation of derivative information and yet may only attain local optima. In practical structural design problems, owing to the availability of standard practical sizes and their restrictions for construction and manufacturing purposes, the design variables are always discrete. In our case for instance, in order to optimize reinforced concrete structural systems, design variables, which are concrete cross-sectional area, reinforcement diameter and reinforcement spacing, should be considered as discrete quantities in practice. However, only very few algorithms have dealt with the optimization of structures under the actual design constraints of code specifications. In fact, it is more rational to use discrete variables during the evaluation process of optimization since every candidate design is a practically feasible solution. This may not be the case when the design variables are continuous, since some of the designs evaluated in the optimization procedures are merely mathematically feasible. This issue can have great significance in solving practical problems of design optimization.

In recent years, genetic algorithms (GAs), which are applications of biological principles into computational algorithm, have been employed to attain the optimum design solutions (Goldberg and Kuo 1987). By applying the principle of survival of the fittest into the optimization of structures, they are particularly amenable to deal with discrete optimum design problems. Besides, they entail only minimum subsidiary information, i.e. on objective function value, to direct the search, yet they are able to search through large spaces quickly. Owing to the processing leverage associated with GAs, the method has a much more global perspective than many commonly employed optimization techniques. However, as far as structural optimization is concerned, literature review shows that steel structures are usually dealt with (Hayalioglu 2001).

In this study, a genetic algorithm (GA) is incorporated into the KBS on the optimum design of reinforced concrete structures subjected to the actual constraints of the British Standard on design of concrete structures for retaining aqueous liquid, BS 8007, employing actual reinforced concrete sections as discrete design variables. When compared with optimum design of steel structures, it should be noted that reinforced concrete structures involve more design variables as it deals with more than a single type of material, namely, concrete and steel reinforcement. Besides, the objective function in the optimization of steel structures can be the minimum weight or cost which are representing the same situation whilst their counterparts of reinforced concrete structures have to be minimum material cost. It is because more than one material are involved and the unit cost of concrete is normally different from that of reinforcement.

In civil engineering, applications of machine learning are still rare. GAs are considered a means of machine learning and it is worthwhile to study its capability to implement knowledge acquisition under a KBS paradigm. Whilst there are numerous applications of KBS or genetic algorithms, a hybrid KBS with GAs has not been found. As such, the objective of this study is to develop a microcomputer-based hybrid KBS that can bring all the design stages, comprising structural optimization with GAs in particular, together into a single and user-friendly environment. It can offer assistance and advice to the design engineer in making decisions. Increase in efficiency, improvement, standardization and optimization of design output and automated record keeping are among the benefits of this hybrid KBS.

Programming Development Environment

The KBS development environment for OPTLIQ is VISUAL RULE STUDIO, which acts as an ActiveX Designer under the Microsoft Visual Basic programming environment. VISUAL RULE STUDIO is an application development environment that combines expert system technologies with object-oriented programming, relational database, graphics capabilities and debugging tools. It furnishes a variety of knowledge representation schemes, different inference mechanisms and capabilities to interface with external programs in windows environment. VISUAL RULE STUDIO provides an interactive windows-based user interface that runs under the conventions of Microsoft Windows.

Architecture of Hybrid System

On the basis of the nature of structural design, the selected knowledge base shell as well as its capability in coupling all design stages for liquid retaining structures, blackboard architecture is adopted here (Engelmore and Morgan 1988). The well-organized architecture can facilitate the communication between diversified knowledge modules involved in the structural design process as well as future extension. Figure 1 shows the architecture of the KBS and the relation between various components of OPTLIQ. The knowledge base is mainly composed of knowledge modules and the blackboard. Knowledge modules correspond to procedural expertise knowledge in solving design problem and are divided into two groups, namely, Design Process and Process Control. The design process can be subdivided into subtasks, each of which can be performed corresponding to a particular expertise within the knowledge base. Message communications between Design Process knowledge modules and Process Control knowledge modules have to effect via the blackboard.

The blackboard contains only declarative knowledge and is divided into two groups, namely, Design Entities and Design Stage. The Design Entities can be considered the breakdown of design concepts of liquid retaining structures. A typical example of objects in the blackboard is Wind Load. The properties of Wind Load are expressed by using several attributes, which is detailed in Figure 2. In this example, the attribute type of terrainCategory is compound, which means that it can take only one of the specified values “general terrain” or “builtup terrain”. Its search order is session context only, meaning that, in determining its value, the system context is searched. The attribute type of basicWindPressure is numeric, which means that it must have a numerical value. Its search order list is rules only. The class in Design Stage includes a number of attributes that represent indicators tracking the current stage of every design context. The knowledge represented in this level will handle the order of execution via the Process Control knowledge modules.

Knowledge Acquisition and Representation

Knowledge acquisition is mainly from literature, including handbooks, standards and codes, and interviews with experienced designers on liquid retaining structures. The domain knowledge of OPTLIQ, such as the minimum percentage of reinforcement, maximum percentage of reinforcement, minimum grade of concrete, etc. is obtained, which is then translated into rules or methods. For instance, the following rule group is expressed using the Production Rule Language, representing knowledge on the determination of bending moments and shear forces in preliminary design.

!RULE GROUP: moment & shear OF BBPreliminaryParameters

RULE to find momentMh : 1 of 4
IF shape OF BBConfigurationRequirement IS rectangular
THEN momentMh OF BBPreliminaryParameters := momentCoefficientMh OF BBPreliminaryParameters*specificGravityOfLiquid OF BBConfigurationRequirement*height OF BBConfigurationRequirement^3/1000

RULE to find momentMv : 2 of 4
IF shape OF BBConfigurationRequirement IS rectangular
THEN momentMv OF BBPreliminaryParameters := momentCoefficientMv OF BBPreliminaryParameters*specificGravityOfLiquid OF BBConfigurationRequirement*height OF BBConfigurationRequirement^3/1000

RULE to find shearVh : 3 of 4
IF shape OF BBConfigurationRequirement IS rectangular
THEN shearVh OF BBPreliminaryParameters := shearCoefficientVh OF BBPreliminaryParameters*specificGravityOfLiquid OF BBConfigurationRequirement*height OF BBConfigurationRequirement^2/100

RULE to find shearVv : 4 of 4
IF shape OF BBConfigurationRequirement IS rectangular
THEN shearVv OF BBPreliminaryParameters := shearCoefficientVv OF BBPreliminaryParameters*specificGravityOfLiquid OF BBConfigurationRequirement*height OF BBConfigurationRequirement^2/100

Interfacing Facilities

Microsoft Visual Basic offers the interfacing facilities where the execution of external program can be achieved by using a command "SHELL *external program name*". All data communications are effected in a fully integrated fashion. Once the execution of the external algorithmic program is done, the KBS resumes its design session environment. Following preparation of the input file in model generation, the nonlinear finite element analysis package ABAQUS is executed. The output data files are saved in the knowledge base for retrieval and manipulation. The KBS is not only aimed to act as a front-end to this finite element package, but also to encapsulate knowledge on the entire design process. In order to achieve the optimum design, structural re-analysis is often inevitable.

Whilst existing algorithmic models generally deal with numerical data input only, the novice user usually finds it more convenient to express information in a natural language. The numerical model generator functions to convert these linguistic variables entered by the user to numerical format conforming to stipulations of the analysis package. The code conformance checking module is used to check the code requirements of BS 8007 (British Standards Institution 1987). The communication between the programs and the knowledge base is performed mainly through the objects and attributes. When a consultation pertinent to extensive data is made to the system, external Access database files are accessed. The database is composed of structural properties of reinforced concrete sections, moment and shear coefficients for various configurations in preliminary design, structural properties of proposed alternatives and final member details in detailed design.

Explanation Facilities
Explanations of liquid retaining structure types, procedures to design various design loading, various code provisions and expert comments regarding design of liquid retaining structures are included in the explanation facilities via the Help button. The explanations consist of built-in specific texts together with associated values of design parameters generated by the knowledge base during system run time.

User Interface Facilities
The user interface facilities allow the user to specify all design requirements and acquire output result from interactive design consultation. Graphical user interfaces, consisting of layers of display screens and pop-up windows are used for messages transfer and hence input, handling and interpretation of data and information have been greatly simplified. During the design process, subjected to conformance with Process Control knowledge modules, the user has the control over the sequence of the actions. Communications with the system are directed mostly through selection of appropriate values or parameters from menu and through replying answers to the queries asked by the system.

Major Design Tasks
Major design tasks performed by the KBS are preliminary synthesis, detailed specification, numerical model generation, nonlinear structural analysis, code conformance checking, and member sizing optimization. The system applies some engineering heuristics such as approximate analysis using span depth ratio, crack width computation and moment and shear coefficients for different length and width ratios to evaluate each alternative. Through the GA optimization module, the alternative with the minimum cost will be recommended by the system as the selected proposed alternative.

During the detailed specification stage, the system generates default loading and support conditions, which can be modified by the user. If the liquid retaining structure is chosen to be underground, the user is required to enter level of ground surface, level of water table, specific weight of soil and active soil pressure coefficient. If the structure is above the ground, the wind loading is calculated according to the Code of Practice for Wind Effects in Hong Kong (Hong Kong Building Development Department 1983). Various load combinations according to BS 8110 (British Standards Institution 1985) are considered.

An iterative process of numerical model generation, finite element analysis, code conformance checking and optimization of structure are then involved. Upon receiving the requisite messages from the knowledge base, a structural model tailor-made for ABAQUS is automatically prepared by the model generator, thus relieving the user of the cumbersome task to manipulate a large amount of data manually. The system evaluates the structural stability from the analysis results and furnishes post-processing. The system then proceeds to check the structural members according to BS 8007. The computed crack width has to be less than the prescribed crack width.

Structural optimization of the structure involves sizing the components under the constraints of the structural adequacy as well as the crack width requirements, which is effected by GAs as depicted in the following sections. The optimum section is then compared with the original section used in the preceding structural analysis. This phase of design typically involves several iterations until convergence is accomplished. Figure 3 is the flowchart showing the overall design algorithm of OPTLIQ.

Optimization by GAs

Holland (1975) has put forward the GAs as an optimization method, which apply the concept on the artificial survival of the fittest coupled with a structured information exchange using randomized genetic operators taken from the nature. GAs differ from traditional optimization algorithms in aspects such as working on coded design variables, population processing, probabilistic operators, and separation of domain knowledge from search. GAs exploit efficiently useful information subsumed in a population of solutions with better performance, by employing operations to generate a new and improved population of strings from an old population. This iterative process to generate and test a population of strings mimics a natural population of biological creatures. GAs search from a population of strings and climb many peaks in parallel simultaneously, hence lowering the probability of locating local optima. GAs require that alternative solutions be coded as strings, which may comprise concatenation of some substrings so that each substring represents a design variable. Individuals and the characters are termed chromosomes and artificial genes, respectively.

The reproduction operator applies the principle of survival of the fittest in the population. Strings with better objective function values, representing more highly fit, receive more offspring in the mating pool. The crossover operator leads to the recombination of individual genetic information from the mating pool and the generation of new solutions to the problem. In the present work, a two-point crossover is utilized. A mutation operator is applied so as to avoid being trapped in local optima and to preserve the diversification among the population in the search. This operator is applied to each offspring in the population with a predetermined probability, termed the mutation probability.

Formulation of Optimum Design Problem

The set of design variables, including the thickness of concrete slab, the diameter of reinforcement bar and the bar spacing, is determined so that the total material cost of the structure comprising n groups of member,

$$\min C(x) = \sum_{i}^{n} U_i * V_i + R_i * W_i \tag{1}$$

is minimized subject to crack width and stress constraints. In eq. (1), U_i and V_i represent the unit cost and the concrete volume of member i respectively. R_i and W_i are the unit cost and the weight of steel reinforcement of member i respectively.

The serviceability limit state or crack width constraint is

$$W_a - W_{\max} \le 0 \tag{2}$$

where W_a is the actual crack width and W_{max} is the prescribed maximum crack width depending on the exposure environment. W_a, is determined using the following formula:-

$$W_a = \frac{3a_{cr}\varepsilon_m}{1 + 2(\frac{a_{cr} - c}{h - x})} \tag{3}$$

where a_{cr} is the distance from the point considered to the surface of the nearest longitudinal bar, ε_m is the average strain for calculation of crack width allowing for concrete stiffening effect, c is the minimum cover to the tension reinforcement, h is the overall depth of the member and x is the depth of the neutral axis (British Standards Institution 1987).

The stress constraints, representing the ultimate limit states of flexure and shear resistance, are expressed in terms of the following equations (British Standards Institution 1985) for members subject to bending and shear force:

$$M_{au} - M_{ult} \leq 0 \tag{4}$$

$$V_a - V_{ult} \leq 0 \tag{5}$$

where M_{au} is the actual ultimate bending moment, M_{ult} is the nominal ultimate moment capacity of the reinforced concrete section, V_a is the actual ultimate shear force and V_{ult} is the nominal ultimate shear capacity of the section. The ultimate moment capacity is determined by the following equations, depending on whether concrete or steel stresses is more critical.

$$M_{ult} = \frac{F_y}{1.15} A_s Z \text{ or } M_{ult} = 0.157 F_{cu} b d^2 \text{ whichever is the lesser} \tag{6}$$

where F_y is the yield strength of reinforcement, A_s is area of tension steel, Z is the lever arm, F_{cu} is the characteristic concrete strength, b is the width of section and d is the effective depth of section.

Ultimate shear capacity of the section ($V_{ult} = v_c b_v d$) is represented by shear strengths v_c for sections without shear reinforcement, which depend upon the percentage of longitudinal tension reinforcement [$100A_s/(b_v d)$] and the concrete grade:-

$$v_c = 0.79[100 A_s / (b_v d)]^{1/3} (400/d)^{1/4} / \gamma_m \tag{7}$$

where b_v is breadth of section, γ_m is a safety factor equal to 1.25, with limitations that [$100A_s/(b_v d)$] should not be greater than three and that (400/d) should not be less than one. For characteristic concrete strengths greater than 25 N/mm^2, the values given by the above expression is multiplied by $(F_{cu}/25)^{1/3}$.

The above is then converted into an unconstrained problem by employing a transformation based on the violations of normalized constraints (Hayalioglu 2001). The normalized form of constraints can be expressed as follows:

$$\frac{W_a}{W_{max}} - 1 \leq 0 \qquad i=1,\ldots..n \tag{8a}$$

$$\frac{M_{au}}{M_{ult}} - 1 \leq 0 \qquad i=1,\ldots..n \tag{8b}$$

$$\frac{V_a}{V_{ult}} - 1 \leq 0 \qquad i=1,\ldots..n \tag{8c}$$

The unconstrained objective function φ(x) is written as

$$\varphi(x) = C(x)[1 + K \sum_i^n \{(\frac{W_a}{W_{max}} - 1)^+ + (\frac{M_{au}}{M_{ult}} - 1)^+ + (\frac{V_a}{V_{ult}} - 1)^+\}] \tag{9}$$

where K is a penalty constant to be selected depending on the problem and

$$(\frac{W_a}{W_{max}}-1)^+ = \max(\frac{W_a}{W_{max}}-1,0) \tag{10a}$$

$$(\frac{M_{au}}{M_{ult}}-1)^+ = \max(\frac{M_{au}}{M_{ult}}-1,0) \tag{10b}$$

$$(\frac{V_a}{V_{ult}}-1)^+ = \max(\frac{V_a}{V_{ult}}-1,0) \tag{10c}$$

The penalty parameter largely depends upon the degree of constraint violation. In this case, values of 10 and 100 have been attempted and it is found that the results are not sensitive to these two values.

The minimum of the unconstrained function φ(x) is searched by the GA, with the best individual having the maximum fitness. In this study, the fitness value is acquired by subtracting from the summation of the maximum and minimum values of the objective function. This ensures that all the fitness values are non-negative and individuals acquire fitness values in accordance with their actual merit. The expression becomes

$$F_j = [\varphi(x)_{\max} + \varphi(x)_{\min}] - \varphi_j(x) \tag{11}$$

where F_j is the fitness of the j-th individual in the population, $\varphi(x)_{max}$ and $\varphi(x)_{min}$ are the maximum and minimum values of φ(x) among the current population respectively and $\varphi_j(x)$ is the objective function value computed for the j-th individual. Computation of the fitness of an individual requires the values of crack width and stresses in the structural system from the results of the finite element analysis.

Optimum Design Algorithm

Figure 4 shows the flowchart of GA for optimum design of reinforced concrete sections in liquid retaining structures. It is programmed under Microsoft Visual Basic programming environment. The starting population is first randomly constructed. The binary codes for the design variables of each individual are decoded and their sequence numbers in the available slab thickness, bar diameter and bar spacing are found. Based on the responses of the structure, the value of unconstrained function φ(x) for each individual is computed. The maximum and minimum values of this function in the population are determined and hence the fitness value for each individual is determined.

By applying the reproduction operator, the individuals are copied into the mating pool according to their fitness. A proportionately higher probability of reproduction selection, s_j, is given to those strings with higher fitness values F_j according to the following distribution

$$s_j = \frac{F_j}{\sum_{j}^{P_{size}} F_j} \tag{12}$$

where P_{size} is the population size. As the number of individuals in the next generation is also the same, the individuals with small fitness die off.

After the mating pool is created, individuals are coupled to generate offspring using a two-site crossover. A set of crossover parameters, consisting of a match and two cross sites, are generated randomly. The genetic operation of crossover is performed on each mated pair with a certain probability, referred to as crossover probability. Suppose that two strings X and Y of length 11 are the mating pair with the following genes

$$X = x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11} \tag{13a}$$

$$Y = y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11} \tag{13b}$$

Two cross sites cs_1 and cs_2 are randomly selected and two new strings are created by swapping all characters between positions cs_1 and cs_2 inclusively from one individual in the pair to the other. For instance, if the cross sites generated are 2 and 7, the resulting crossover yields two new strings X' and Y' following the partial exchange.

$$X' = x_1, x_2, | y_3, y_4, y_5, y_6, y_7, | x_8, x_9, x_{10}, x_{11} \tag{14a}$$

$$Y' = y_1, y_2, | x_3, x_4, x_5, x_6, x_7, | y_8, y_9, y_{10}, y_{11} \tag{14b}$$

Mutation is then applied to each offspring in the new population, which flips the gene of an offspring from 1 to 0 or vice versa at random position. The initial population is then replaced by the new population.

The above steps are then iterated until the distance between the maximum and the average fitness values of the current population is less than a certain threshold. The optimum values obtained from interactive optimization can then be added to the knowledge base, which is effectively extended through machine experimentation. Machine learning can be effected since the final optimum structural section under the specified loading and geometrical configuration from the finite element structural analysis is added to the database containing the heuristics during the preliminary design. The system can also be used as a means for testing the available empirical knowledge from the model run.

Application Case

A typical example of liquid retaining structure is used to illustrate the application of OPTLIQ. A rectangular shape liquid retaining structure with two compartments located above the ground, having a volume of 100 m^3, a depth of 5 m and breadth/width ratio of 1.2, is designed under a severe exposure environment, i.e. the maximum design crack width is 0.2mm.

Preliminary Design

Upon execution of the system, the main menu screen is displayed as shown in Figure 5. During the preliminary design stage, heuristics are used to evaluate different alternatives. The system searches the databases on moment and shear coefficients and on sectional properties and suggests an initial member thickness of 225 mm with reinforcement diameter 10 mm at 100 mm spacing as the most suitable alternative. The user can choose between system's selection and user's selection and, in this case, the system's selection is opted.

Detailed Specifications and Analysis

Detailed design specifications, as shown in Figure 6 are input for the selected alternative. The iterative process of numerical model generation, structural analysis, code conformance checking and optimized member sizing is commenced next.

Structural Optimization

The practically available values of the design variables are given in the lists L_1, L_2 and L_3, representing slab thickness, bar diameter and bar spacing respectively.

L_1 = (200, 225, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000) (mm). (15a)
L_2 = (10, 12, 16, 20, 25, 32, 40) (mm). (15b)
L_3 = (100, 125, 150, 175, 200, 225, 250, 275, 300) (mm). (15c)

There are 13, 7 and 9 different values for the three design variables respectively. A binary coding is adopted for the design variables because it is easy to handle. The total length of the string becomes eleven, with two substrings of length four, representing the slab thickness and the bar spacing, and one substring of length three, representing the bar diameter. The population size, the crossover probability and the mutation probability are selected as 10, 0.95 and 0.01, respectively. These values are consistent with other empirical studies with high crossover probability, low mutation probability, and moderate population size, although it is found that GAs are not highly sensitive to these parameters.

The individual numbers and the strings generated randomly are shown in the first and second columns of Table 1 respectively. The third, fourth and fifth columns display the corresponding values of the three design variables for each individual in the population, which can be acquired by decoding the first substring of length four, second substring of length three and last substring of length four, respectively. Column six gives the costs of the structure for the design represented by the individual strings. Values of the objective function, which account for the possible violation of constraints, are given in column seven. It can be seen that constraints are violated for individuals 1, 4, and 10. Table 1 shows that the individual 3 is the best fit since it has the least cost ($45296) among the ten individuals and has not violated any constraints. Fitness values are computed for all the ten individuals as shown in column eight. By applying the proportionate probability of reproduction selection, the actual count of individuals in the mating pool is shown in column nine. As shown in column ten, a mating pool is created where individuals 1,2,5,6,8, and 9 get one copy each, individuals 3 and 7 get two copies, and 4 and 10 die off. The crossover operator is then applied and the crossover parameters generated, including the mating pair and the cross sites, are shown in columns eleven, twelve and thirteen. Since there are ten individuals in the population, there are five matching pairs selected randomly. Individual 1 gets 6, 2 gets 8, 3 gets 7, 4 gets 5, and 9 gets 10. The population after the crossover becomes the initial population of generation 2, which is processed as shown in Table 2.

Figure 7 shows the relationship between the minimum cost versus the number of generations for the population size of 10. The minimum cost of $38687, representing a reinforced concrete section of member thickness 300 mm with reinforcement diameter 25 mm at spacing 225 mm, is found after 5 generations. It is interesting that near-optimal results are obtained after only 5 generations (approximately 48 new function evaluations) even though the size of the search space is huge (2^{11} = 2048). The number of points explored is small and represents only 2.3% or so of the total search space.

Since the population size may play an important role in the value of the minimum cost and in the number of generations produced, another population size of 8 members has also been performed. In that case, as shown also in Figure 7 the same minimum cost is found after 6 generations. Again, in order to acquire the near-optimal results, the number of points explored is small and represents only 2.2% or so of the total search space.

Design Report
Figure 8 shows the final design report that provides the type of structure, location, volume, selected reinforced concrete section, total number of node, total number of element, calculated crack width, maximum bending moment, shear force and the corresponding member number at both ultimate and serviceability limit states. The values acquired from the final design, together with detailed specifications, can be added to the knowledge base to improve the existing heuristic knowledge.

Conclusions
A coupled microcomputer KBS on optimum design of liquid retaining structure (OPTLIQ) was implemented to combine expert knowledge with GA optimization, object-oriented programming, graphics capabilities, KBS technologies, conventional algorithmic models and relational databases under a windowing environment. The prototype system undertakes all major design stages including preliminary synthesis, detailed specification, numerical model generation, finite element analysis, code conformance checking, and member sizing optimization. The incorporated GA encapsulating reproduction, crossover, and mutation operators locates the optimal solution quickly after examining a minute portion of the discrete design alternatives in the design of liquid retaining structures. The system will be an ideal research tool to validate and enhance our empirical knowledge, which in turn may lead to more efficient and optimized structural design. It can act as a repository of empirical knowledge provided by experienced specialists.

References
British Standards Institution (1985). *BS 8110: Structural use of concrete*, British Standards Institution, London.
British Standards Institution (1987). *BS 8007: Design of concrete structures for retaining aqueous liquids*, British Standards Institution, London.
Chau, K.W. (1992). “An expert system for the design of gravity-type vertical seawalls.” *Engineering Applications of Artificial Intelligence*, 5(4), 363-367.
Chau, K.W., Cheng, C.T., and Li, C.W. (2002). “Knowledge management system on flow and water quality modeling.” *Expert Systems with Applications*, 22(4), 321-330.
Chau, K.W., and Ng, V. (1996). “A knowledge-based expert system for design of thrust blocks for water pipelines in Hong Kong.” *Water Supply Research and Technology - Aqua*, 45(2), 96-99.
Chau, K.W., and Yang, W.W. (1992). “A knowledge-based expert system for unsteady open channel flow.” *Engineering Applications of Artificial Intelligence*, 5(5), 425-430.
Chau, K.W., and Zhang, X.N. (1995). “An expert system for flow routing in a river network.” *Advances in Engineering Software*, 22(3), 139-146.
Engelmore, R. and Morgan, T. (1988). *Blackboard systems*, Addison_Wesley, Wokingham.
Goldberg, D.E., and Kuo, C.H. (1987). “Genetic algorithms in pipeline optimization.” *Journal of Computing in Civil Engineering*, *ASCE*, 1(2), 128-141.
Hayalioglu, M.S. (2001). “Optimum load and resistance factor design of steel space frames using genetic algorithm.” *Structural and Multidisciplinary Optimization*, 21(4), 292-299.
Holland, J.H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, Mich.
Hong Kong Building Development Department (1983). *Code of Practice for Wind Effects in Hong Kong*, Hong Kong Building Development Department, Hong Kong.
Kumar, B. (1995). *Knowledge Processing for Structural Design*, Topics in Engineering Vol. 25, Computational Mechanics, Southampton.

Lin, S., and Albermani, F. (2001). “Lattice-dome design using a knowledge-based system approach.” *Computer-aided Civil and Infrastructure Engineering*, 16(4), 268-286.

Figure captions

Figure 1. Architecture of the hybrid knowledge-based system

Figure 2. Characteristics of class Wind Load

Figure 3. Flowchart showing overall design algorithm of OPTLIQ

Figure 4. Flowchart of genetic algorithm for optimum design of discrete reinforced concrete sections in liquid retaining structures

Figure 5. Screen showing the main menu

Figure 6. Screen showing summary of design specifications

Figure 7. Minimum cost versus number of generation

Figure 8. Screen showing the final design report

No	*Population*	*Thickness*	*Bar size*	*Bar spacing*	*C(x)*	*φ(x)*	*F*	*count*	*Mating pool*	*Mate*	cs_1	cs_2
1	11100101110	1000	12	100	83116	452151	264973	1	11100101110	6	1	9
2	01101101011	400	32	100	86951	86951	630173	1	01101101011	8	9	10
3	01001100111	300	32	250	45296	45296	671828	2	01001100111	7	9	10
4	11011010110	1000	25	225	90550	597195	119929	0	01001100111	5	1	9
5	10010000011	600	40	150	104345	104345	612779	1	10010000011	4	1	9
6	00001001010	1000	20	100	97342	97342	619782	1	00001001010	1	1	9
7	01001010011	300	25	150	45689	45689	671435	2	01001010011	3	9	10
8	10011110110	600	40	225	84586	84586	632538	1	10011110110	2	9	10
9	11100000010	1000	40	125	146246	146246	570878	1	11100000010	10	4	6
10	00000011011	1000	10	100	80671	671828	45296	0	01001010011	9	4	6

Table 1. Details of computations in generation 1

No	*Population*	*Thick ness*	*Bar size*	*Bar spacing*	*C(x)*	*φ(x)*	*F*	*count*	*Mating pool*	*Mate*	cs_1	cs_2
1	10001001010	500	20	100	59786	59786	395541	1	10001001010	8	7	9
2	01101101011	400	32	100	86951	86951	368376	1	01101101011	4	3	5
3	01001100111	300	32	250	45296	45296	410031	2	01001100111	9	8	11
4	00010000011	200	40	150	74300	74300	381027	1	00010000011	2	3	5
5	11001100111	900	32	250	90365	410031	45296	0	01001100111	6	6	9
6	01100101110	400	12	100	38048	123360	331967	1	01100101110	5	6	9
7	01001010011	300	25	150	45689	45689	409638	2	01001010011	10	0	9
8	10011110110	600	40	225	84586	84586	370741	1	10011110110	1	7	9
9	11101000010	1000	20	125	92897	170001	285326	1	11101000010	3	8	11
10	01000010011	300	10	150	26239	370620	84707	0	01001010011	7	0	9

Table 2. Details of computations in generation 2

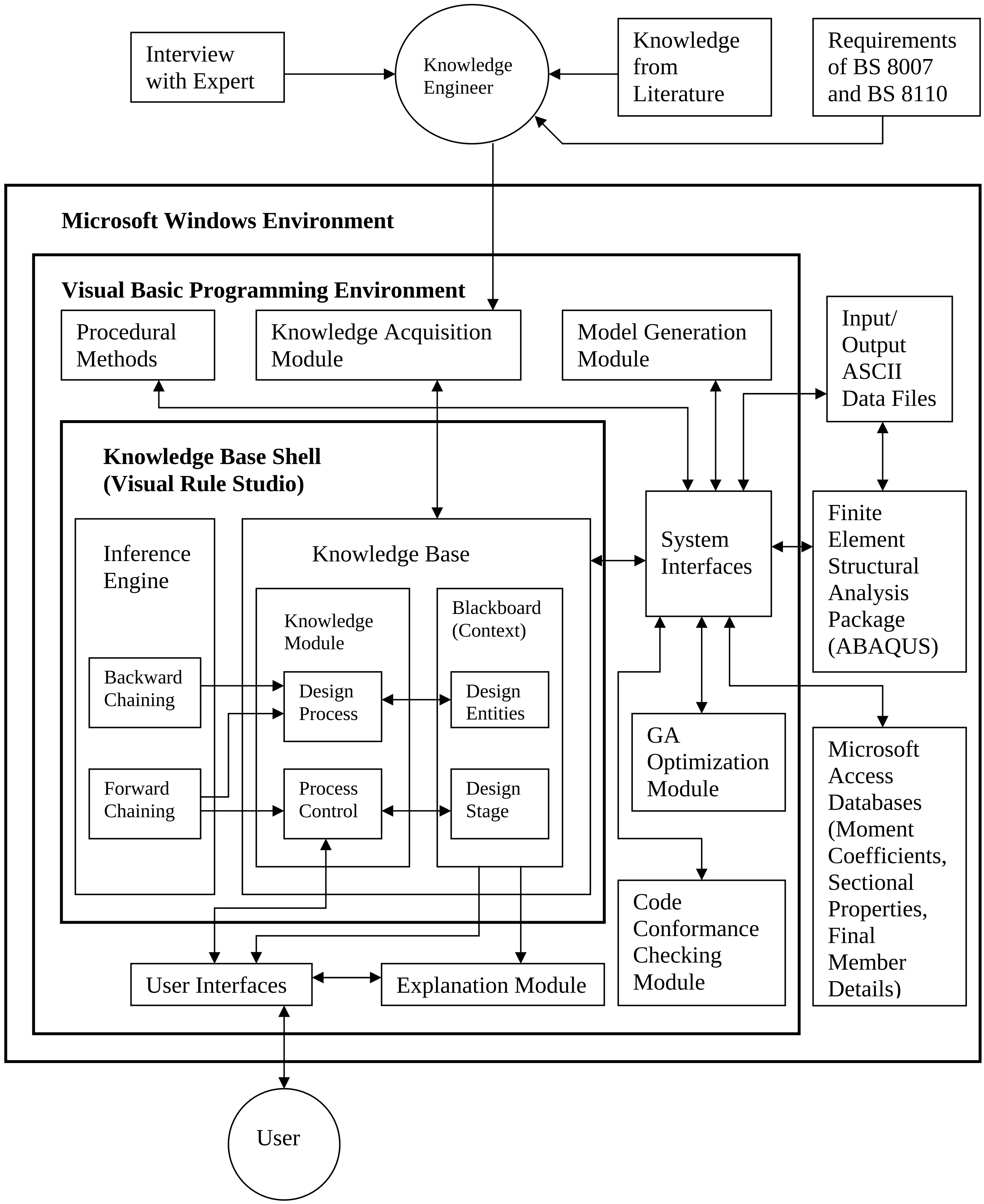
Interview with Expert
Knowledge Engineer
Knowledge from Literature
Requirements of BS 8007 and BS 8110
Microsoft Windows Environment
Visual Basic Programming Environment
Procedural Methods
Knowledge Acquisition Module
Model Generation Module
Input/ Output ASCII Data Files
Knowledge Base Shell (Visual Rule Studio)
Inference Engine
Knowledge Base
Knowledge Module
Blackboard (Context)
Backward Chaining
Design Process
Design Entities
Forward Chaining
Process Control
Design Stage
System Interfaces
Finite Element Structural Analysis Package (ABAQUS)
GA Optimization Module
Code Conformance Checking Module
Microsoft Access Databases (Moment Coefficients, Sectional Properties, Final Member Details)
User Interfaces
Explanation Module
User

Figure 1.

```
CLASS BBWindLoad
        WITH terrainCategory COMPOUND
                general terrain,
                builtup terrain
                WHEN CHANGED
                        BEGIN
                        FORGET BBWindLoad.basicWindPressure
                        END
                SEARCH ORDER CONTEXT
        WITH methodOfSelection COMPOUND
                found by system,
                user defined
                WHEN CHANGED
                        BEGIN
                        FORGET BBWindLoad.shapeFactorCs
                        FORGET BBWindLoad.shapeFactorCsx
                        FORGET BBWindLoad.shapeFactorCsy
                        FORGET BBWindLoad.heightAspectFactorChx
                        FORGET BBWindLoad.heightAspectFactorChy
                        FORGET BBWindLoad.forceCoefficientCfx
                        FORGET BBWindLoad.forceCoefficientCfy
                        FORGET BBWindLoad.basicWindPressure
                        END
                SEARCH ORDER CONTEXT
        WITH shapeFactorCs NUMERIC
                SEARCH ORDER RULES
        WITH shapeFactorCsx NUMERIC
                SEARCH ORDER RULES
        WITH shapeFactorCsy NUMERIC
                SEARCH ORDER RULES
        WITH heightAspectFactorChx NUMERIC
                SEARCH ORDER RULES
        WITH heightAspectFactorChy NUMERIC
                SEARCH ORDER RULES
        WITH forceCoefficientCfx NUMERIC
                SEARCH ORDER RULES
        WITH forceCoefficientCfy NUMERIC
                SEARCH ORDER RULES
        WITH topLevel NUMERIC
                WHEN CHANGED
                        BEGIN
                        FORGET BBWindLoad.basicWindPressure
                        PURSUE BBWindLoad.basicWindPressure
                        END
                SEARCH ORDER CONTEXT
        WITH basicWindPressure NUMERIC
                SEARCH ORDER RULES
        WITH designedWindPressureX NUMERIC
                SEARCH ORDER RULES CONTEXT
        WITH designedWindPressureY NUMERIC
                SEARCH ORDER RULES CONTEXT
```

Figure 2.

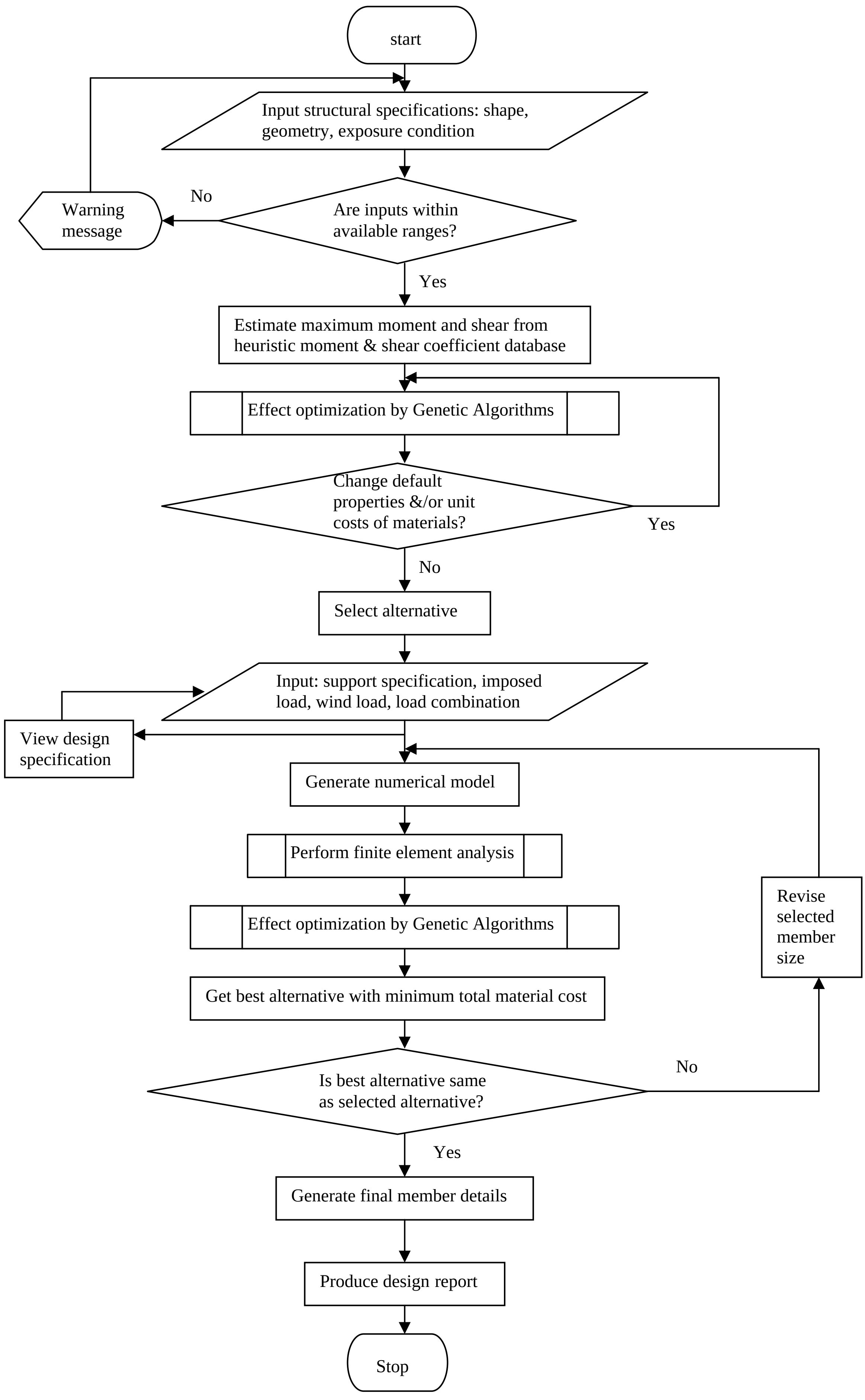
start
Input structural specifications: shape, geometry, exposure condition
Warning message
No
Are inputs within available ranges?
Yes
Estimate maximum moment and shear from heuristic moment & shear coefficient database
Effect optimization by Genetic Algorithms
Change default properties &/or unit costs of materials?
Yes
No
Select alternative
Input: support specification, imposed load, wind load, load combination
View design specification
Generate numerical model
Perform finite element analysis
Effect optimization by Genetic Algorithms
Revise selected member size
Get best alternative with minimum total material cost
No
Is best alternative same as selected alternative?
Yes
Generate final member details
Produce design report
Stop

Figure 3.

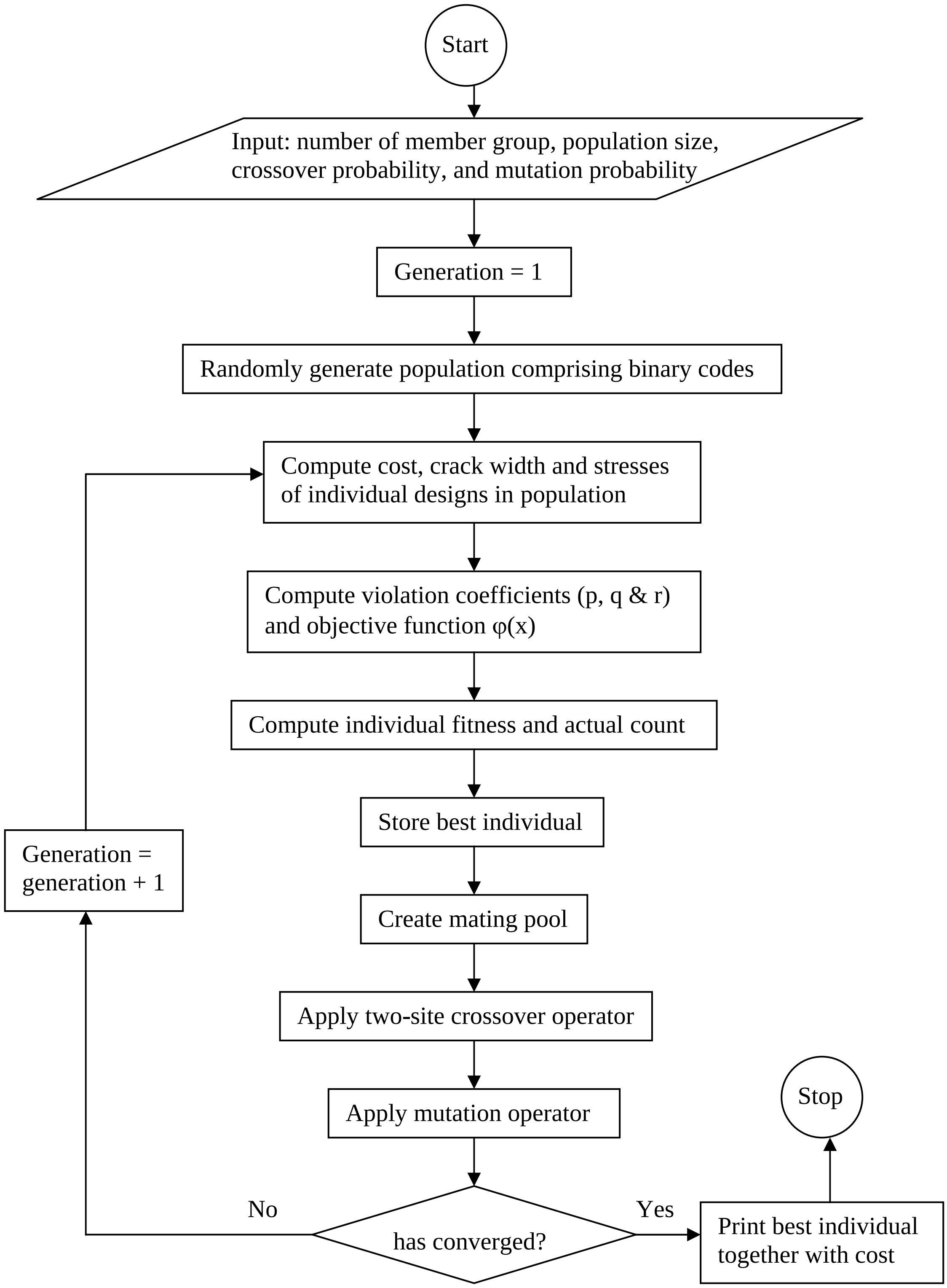
Start
Input: number of member group, population size, crossover probability, and mutation probability
Generation = 1
Randomly generate population comprising binary codes
Compute cost, crack width and stresses of individual designs in population
Compute violation coefficients (p, q & r) and objective function φ(x)
Compute individual fitness and actual count
Store best individual
Create mating pool
Apply two-site crossover operator
Apply mutation operator
has converged?
No
Yes
Generation = generation + 1
Print best individual together with cost
Stop

Figure 4.

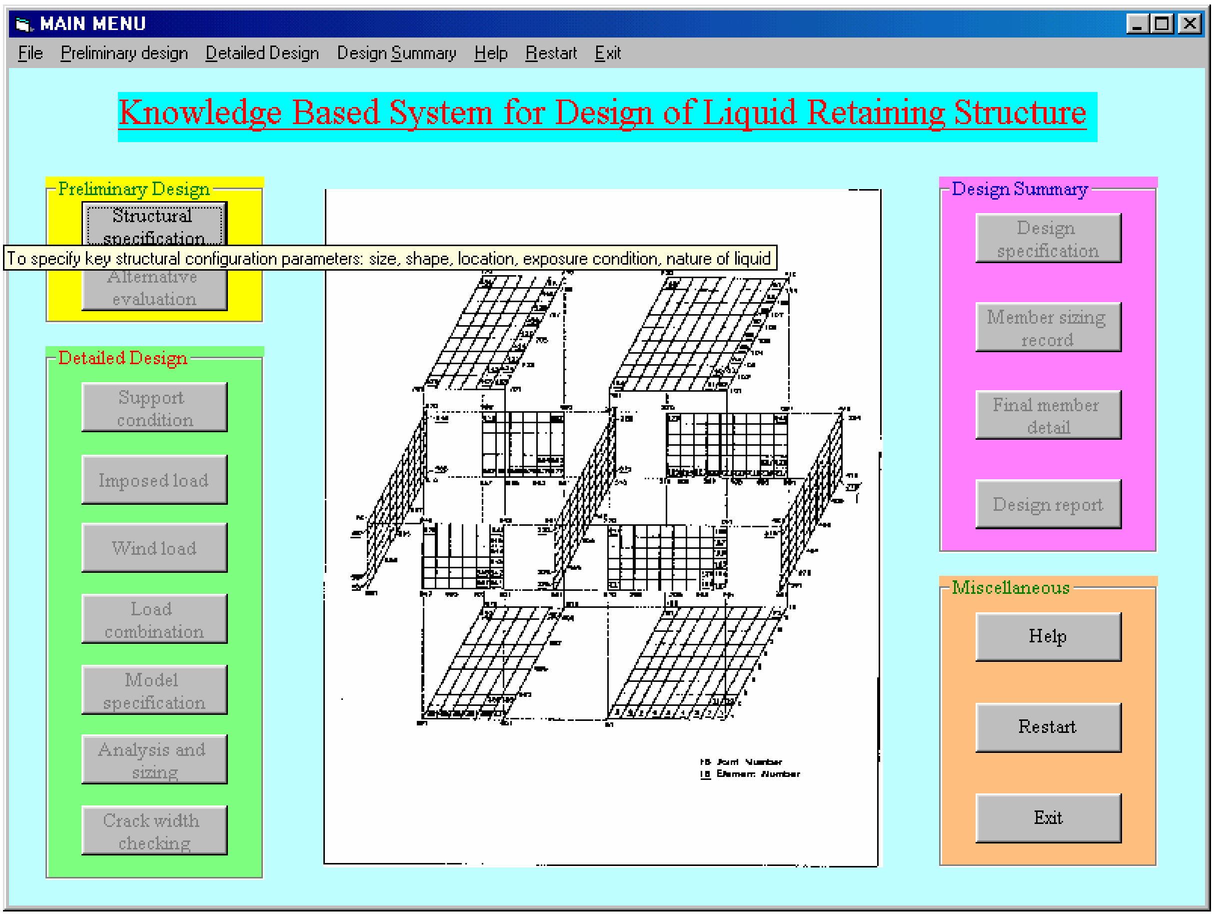
MAIN MENU
File
Preliminary design
Detailed Design
Design Summary
Help
Restart
Exit
Knowledge Based System for Design of Liquid Retaining Structure
Preliminary Design
Structural specification
To specify key structural configuration parameters: size, shape, location, exposure condition, nature of liquid
Alternative evaluation
Detailed Design
Support condition
Imposed load
Wind load
Load combination
Model specification
Analysis and sizing
Crack width checking
Design Summary
Design specification
Member sizing record
Final member detail
Design report
Miscellaneous
Help
Restart
Exit

Figure 5.

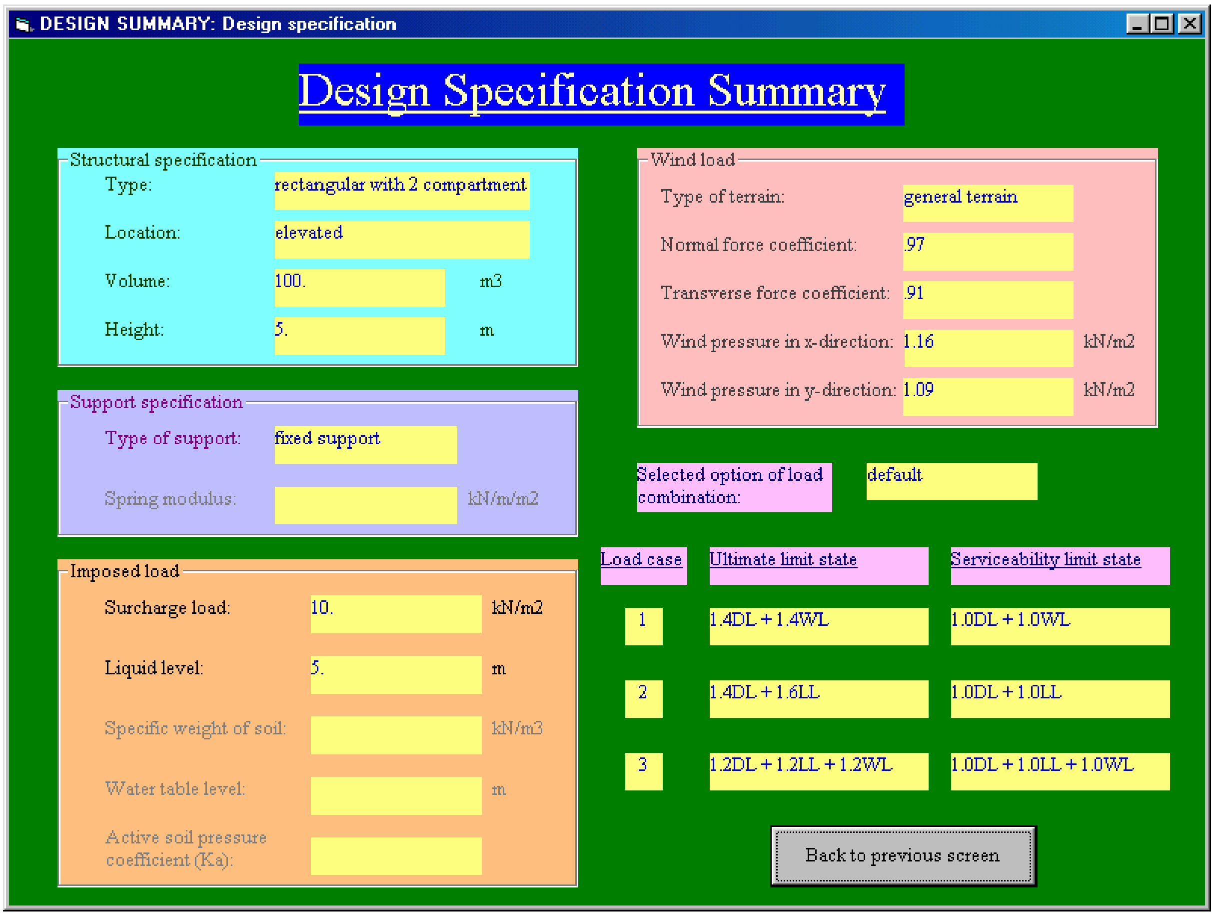
DESIGN SUMMARY: Design specification
Design Specification Summary
Structural specification
Type: rectangular with 2 compartment
Location: elevated
Volume: 100. m3
Height: 5. m
Support specification
Type of support: fixed support
Spring modulus: kN/m/m2
Imposed load
Surcharge load: 10. kN/m2
Liquid level: 5. m
Specific weight of soil: kN/m3
Water table level: m
Active soil pressure coefficient (Ka):
Wind load
Type of terrain: general terrain
Normal force coefficient: .97
Transverse force coefficient: .91
Wind pressure in x-direction: 1.16 kN/m2
Wind pressure in y-direction: 1.09 kN/m2
Selected option of load combination: default
Load case
Ultimate limit state
Serviceability limit state
1 1.4DL + 1.4WL 1.0DL + 1.0WL
2 1.4DL + 1.6LL 1.0DL + 1.0LL
3 1.2DL + 1.2LL + 1.2WL 1.0DL + 1.0LL + 1.0WL
Back to previous screen

Figure 6.

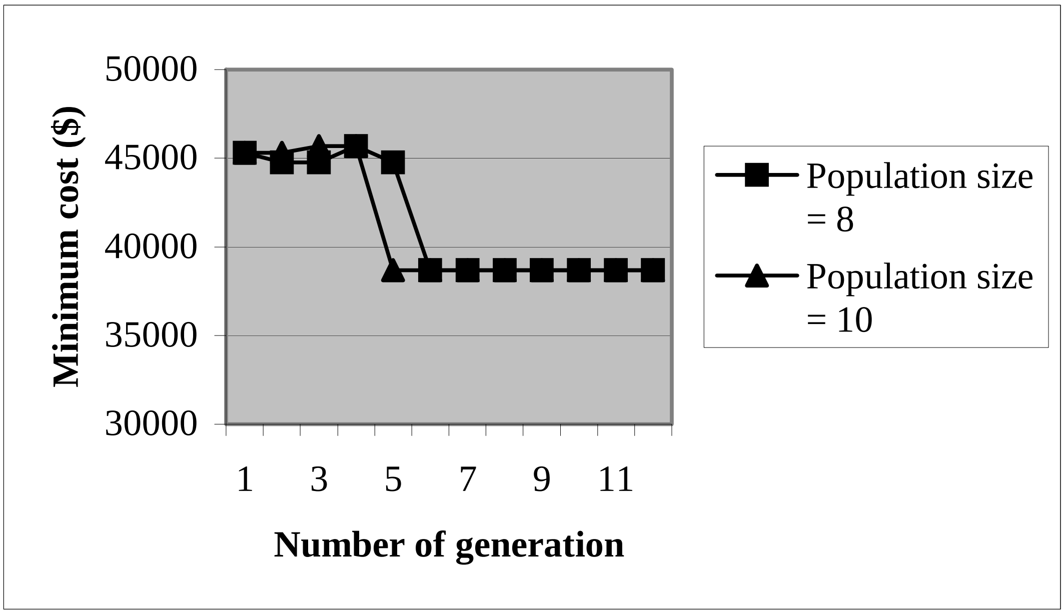
50000
45000
40000
35000
30000
Minimum cost ($)
1
3
5
7
9
11
Number of generation
Population size = 8
Population size = 10

Figure 7.

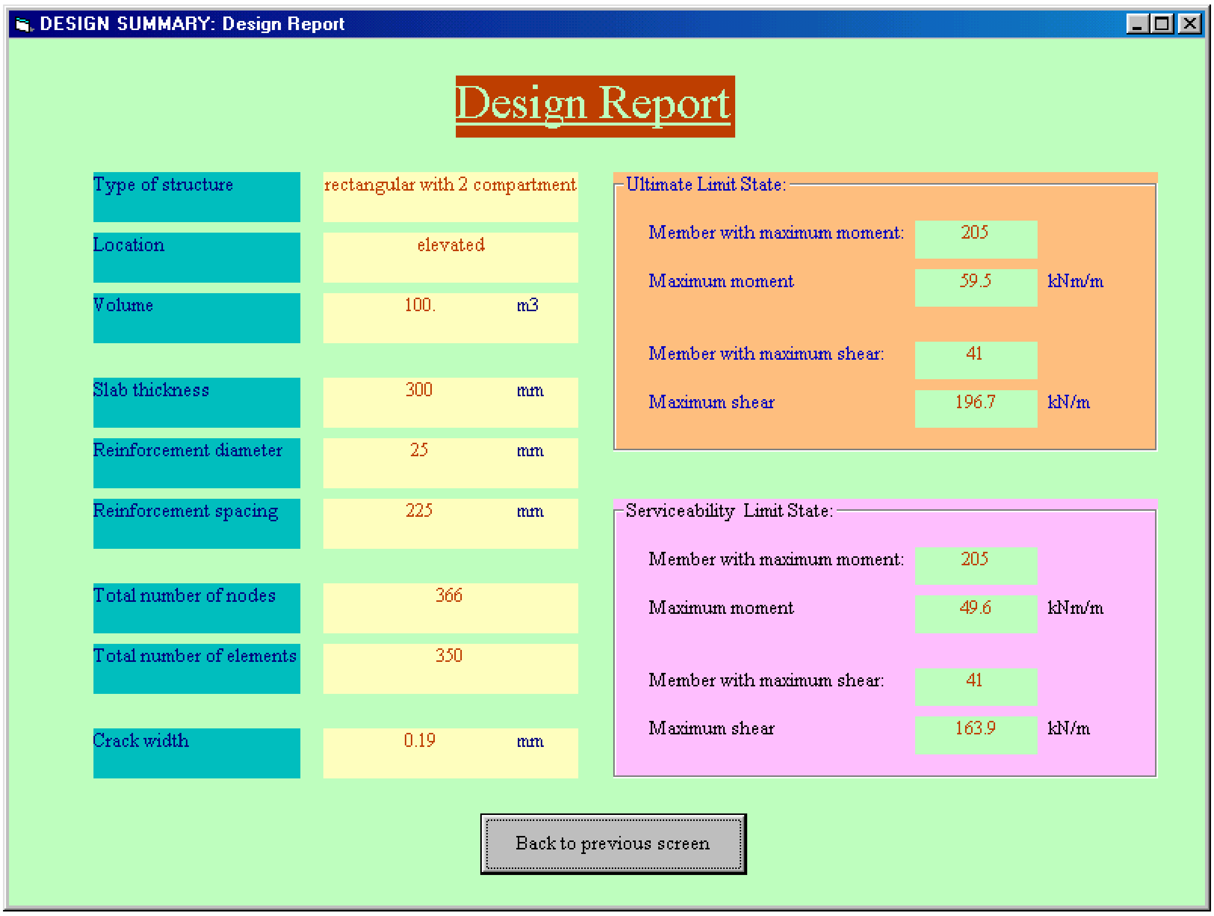
DESIGN SUMMARY: Design Report
Design Report
Type of structure
rectangular with 2 compartment
Location
elevated
Volume
100.
m3
Slab thickness
300
mm
Reinforcement diameter
25
mm
Reinforcement spacing
225
mm
Total number of nodes
366
Total number of elements
350
Crack width
0.19
mm
Ultimate Limit State:
Member with maximum moment:
205
Maximum moment
59.5
kNm/m
Member with maximum shear:
41
Maximum shear
196.7
kN/m
Serviceability Limit State:
Member with maximum moment:
205
Maximum moment
49.6
kNm/m
Member with maximum shear:
41
Maximum shear
163.9
kN/m
Back to previous screen

Figure 8.