Session 2: Reasoning and learning

Objekttyp: **Group**

Zeitschrift: **IABSE reports = Rapports AIPC = IVBH Berichte**

Band (Jahr): **68 (1993)**

PDF erstellt am: **27.09.2024**

SESSION 2

REASONING AND LEARNING

Spatial Analysis and Reasoning for Design Railway Location

Analyse et raisonnement spatial pour les études ferroviaires

Räumliche Analyse und Folgerungen bei Eisenbahnprojektstudien

Wei WEN
Ph.D. Candidate
Northern Joatong Univ.
Beijing, China

Lian-Zi WANG
Assoc. Professor
Northern Jaotong Univ.
Beijing, China

Bao-Zong YUAN
Professor
Northern Jiaotong Univ.
Beijing, China

SUMMARY

The paper describes a new system for designing railway location applying the theories and technics of information and knowledge processing. It has some object - oriented characteristics which are data abstraction, behaviour sharing, evolution and correctness, wherein the objects are the basic processing units. Every object is divided in two parts: physical and logical. The Materialization Operator and Dematerialization Operator can realize the transformations between the physical objects and logical objects. The concepts and operators used form an algebra system of objects. All of these make the system have the capability of spatial analysis and spatial reasoning. Finally the paper gives the construction graph of the system and an example analysis and its processing.

RÉSUMÉ

Cet article décrit un nouveau système appliquant des théories et des techniques relatives au traitement de l'information ainsi que de la connaissance dans le tracé des voies de chemin de fer. Le système en question comporte des caractéristiques à orientation objet, tels qu'abstraction de données, partage du comportement, évolution et exactitude; chacun des objets, tout comme les unités de traitement de base, sont scindés selon leurs deux parties constituantes physiques et logiques. Un opérateur de matérialisation et un opérateur de dématérialisation accomplissent les transformations entre les objets physiques et les objets logiques. La forme d'un système algébrique d'objets qui en résulte est apte à permettre aussi bien l'analyse spatiale que le raisonnement spatial. Pour terminer, les auteurs exposent la construction graphique du système et le traitement d'un exemple d'analyse.

ZUSAMMENFASSUNG

Der Beitrag beschreibt den Einsatz von Theorien und Techniken der Informations- und Wissensverarbeitung in der Trassierung von Eisenbahnlinien. Im vorgestellten System, das bestimmte objektorientierte Merkmale wie Datenabstraktion, Verhaltensteilung, usw. enthält, sind die Objekte als die zu verarbeitenden Grundbausteine in ihren physischen und logischen Bestandteil aufgetrennt. Die Transformation zwischen dem physischen und dem logischen Objekt wird durch einen Materialisierungs- bzw. einen Dematerialisierungsoperator bewerkstelligt. Es entsteht die Form eines algebraischen Systems von Objekten, das die räumliche Analyse und räumliches Schliessen zulässt. Der Graph des Systemaufbaus und die Verarbeitung eines Beispiels runden den Beitrag ab.

1. INTRODUCTION

The design of railway location is an important part in the whole railway civil engineering. There are many facts to influence it, such as economics, hydrography, geology, geomorphology, topography, and so on. Planning basic railway direction, determining the spatial position of railway and distributing some railway buildings, for example, stations, bridges and channels are its main tasks according to the requirements combining with the natural resource and the economic development of regions through which the railway will pass.

In the past, the design of railway location mainly relied on a lot of data from point to point measured by human beings. Railway engineers repeatedly discussed the merits and demerits among different frames and designed the railway locations according to certain technical standards and some requirements. It was apperently that this design method cost much time and labour and its ability to adapt to new environment was not strong. When a new railway engineering started to be designed, all of designs had to begin from the start again. Much repetition work had to be done to adapt to new requirements and new data, so the design time of the old method was very long and the cost of design was very high.

In the recent years computers have been developed rapidly, especially, the theories and technics of information and knowledge processing have been applied in many areas successfully, so it is necessary and possible to use artificial intelligence in the design of railway location and improve design effectiveness more reasonably.

This paper gives a full discription of a new system for designing railway location. The system combines all sorts of data, information, rules and knowledge applying some theories and technics of information and knowledge processing, processes many complex facts with computers and has the capability of spatial analysis and spatial reasoning substituting engineers. At first, we segment regions according to the pictures taken from airplanes combining with the other data and information and determine rivers, mountains, cities, towns, villages and roads. Every region has many attributes and their values which can represent its typical features, such as mines, oil fields, forests, enterprises, agricutures and populations, etc. All of regions are classified and abstracted to form the basic processing units——objects. Each object includes two parts: physical and logical. The physical objects mainly not only come from the primitive images transformed through the pictures, but also are visualized after processing. The logical objects are logical representation of the objects and are obtained with abstracting operations at diverse levels and take part in all kinds of logical operation and reasoning. Thus the relations among the objects can be formed a net of semantic description, and every object is a node of semantic net. Certainly the semantic net is dynamic changeable at different abstraction levels, so the objects and their properties and attribute values can be queried and indexed. The attributes at higher level are the abstraction of data and information of the objects at lower level, and the objects at lower level can inherit the useful data and information of the objects at higher level.

In the system, there are two kinds of design strategies of railway location. One is from bottom to top, which firstly begins at the start point and the terminal point and goes into deeper levels step by step. The other is from top to bottom, which starts at two basic object nodes and extends outer levels and ends at the start point and the terminal point. Of course the distinction between two strategies is not extrict.

In a word, the system for designing railway location can combine with some theories and technics of information and knowledge processing and improve intelligent level of railway civil engineering. During designing this system, some new concepts, such as pure object and algebra system of object are put forward so that the complete design theory can be formed.

2. SOME BASIC CONCEPTS

In the introduction, we know that the system for designing railway location is based on objects. All sorts of computation and reasoning in the system are carried out through the attributes of objects. As follow some basic concepts are defined.

2.1 Objects

An object is an encapsulation of a set of operations or methods which can be invoked externally and of a state which remembers the effect of the methods.

The system for designing railway location supporting objects is characterised by the following features:

1.modularisation——all details of an object are brought together in one place.

2.information hiding——access to an object is controlled through a well–defined interface; all other details of the objects are hidden from the user of that object.

3.behavioural——the behaviour of an object is captured by the full operational interface presented by that object.

4.object interaction——a mechanism is provided to allow an object to invoke methods on another object.

5.self reference——local operations are accessed in the same way as remote operations by invoking a method on self.

2.2 Physical Objects

Physical objects are real physical meaning of objects. They not only refer to the primitive images transformed through the pictures taken, but also can be synthesized and visualized if necessary. The three dimensional objects such as geography and geomorphology can be shown on sreens of computers.

2.3 Logical objects

Logical objects are abstraction forms of objects and semantic descriptions of physical objects. They represent the logical relations of objects and show the properties and attribute values of physical objects. The attribute values take part in all kinds of computation and analysis in the system.

2.4 Classes

After the concepts of objects, physical objects and logical objects have been defined, they should be classified based on certain rules. A class is a template from which objects may be created. It contains a definition of the state desciptors and methods for the object. The class template therefore provides a complete description of a class in terms of its external interface and internal algorithems and data structures.

2.5 Inheritance

The classifications of objects are made at different levels. The new class is said to be a subclass of the old class. Similarly, the old class is the super class of the new class. The new class therefore shares the behaviour of the old class but has modified or additional behaviour. This sharing of behaviour is the essential feature of inheritance. Inheritance is the incorporation of the behaviour of one class into another. A class which inherits from another class inherits all the methods and attributes of that class.

3.THE ALGEBRA SYSTEM OF OBJECTS

In the system for designing railway location, the concepts, the transformations between physical objects and logical objects and a series of operations on objects can be formed an algebra system of objects which has the capability of spatial analysis and spatial reasoning. Its representation formalized is:

$G\ (V_L, V_P, S, X_0, R)$

wherein:

V_L is the set of logical objects.

V_P is the set of physical objects.

S is the limited non–null set of object names.

X_0 is an element of S and represents the set of object names which is head.

R is the mapping from S to $2^{V_L \cup S} \times V_P$ and is the object rules transformed from all kinds of technical standards of railway location design.

Two–element representation of objects is (A_m, A_i) , wherein A_m is logical objects and A_i is physical objects. Every operator of the system is divided two parts also. OP_m is logical operator and OP_i is physical operator, so the operator can be written as:

$$OP = (OP_m, OP_i)$$

When an operator acts on two objects $X\ (X_m, X_i)$ and $Y\ (Y_m, Y_i)$, it is written as:

$$OP\ (X,Y) = (OP_m\ (X,Y)\ , OP_i\ (X,Y)\ )$$

If OP_m is independent of X_i and Y_i and OP_i is independent of X_m and Y_m , then

$$OP\ (X,Y) = (OP_m\ (X_m, Y_m)\ , OP_i\ (X_i, Y_i)\ )$$

Among the operators MOP(Materialization Operators) and DMOP(Dematerialization Operator) are extremely important, because they can realige the transformations between logical objects A_m and physical objects A_i . If there is an object $X = (X_m, X_i)$ in which logical part X_m and physical part X_i are fully transformed each other, i.e. $\{X_m\} = DMOP\ \{X_i\}$ and $\{X_i\} = MOP\ \{X_m\}$, this object is called the pure object being useful in the system.

Besides above two kinds of basic operators MOPand DMOP, following other operators are necessary to be introduced.

3.1 Combining Operator COM

$$\text{COM} \ ((A_m, A_i) , (B_m, B_i))$$
$$= (\text{CONCEPT}-\text{MERGE} \ (A_m, B_m)$$
$$, \text{SUPERPOSE} \ (A_i, B_i))$$

Explanation: physical objects A_i and B_i add to form a new physical object whose corresponding logical meaning combines two concepts of logical objects A_m and B_m, for example, COM (City A, Station B) = (Station B in City A)

3.2 Subtracting Operator SUB

$$\text{SUB} \ ((A_m, A_i) , (B_m, B_i))$$
$$= (\text{CONCEPT}-\text{DIEF} \ (A_m, B_m) , \text{REMOVE} \ (A_i, B_i))$$

Explanation: when physical object B_i is removed from A_i, the logical meaning of A_m will change and form a new concept, for example, SUB (Town A having bridge B, Bridge B) = (Town A without bridge B)

3.3 Inverting Operator INV

$$\text{INV} \ (A_m, A_i)$$
$$= (\text{CONCEPT}-\text{INV} \ (A_m) , \text{INVERT} \ (A_i))$$

Explanation: this operator is used to transform objects which are under ground and above surface.

3.4 Marking Operator MAR

$$\text{MAR} \ ((A_m, A_i) , (B_m, B_i))$$
$$= (\text{CONCEPT}-\text{MARKING} \ (A_m, B_m) , \text{MARK} \ (A_i, B_i))$$

Explanation: marking refers to the important feature of objects. Object B is a feature of object A, for example,B represents mountains, A is a city, MAR (A,B) = (City A with mountains)

3.5 Enhancing ENH

$$\text{ENH} \ ((A_m, A_i) , (B_m, B_i))$$
$$= (\text{CONCEPT}-\text{ENH} \ (A_m, B_m) , A_i)$$

Explanation: this operator add some attributes of object B to object A, so the meaning of A will expand.

3.6 Indexing Operator IDX

$$\begin{aligned} & IDX\ (A_m,A_i) \\ = & (CONCEPT\text{-}REDUCE\ (A_m)\ ,\ IMAGE\text{-}REDUCE\ (A_i)) \end{aligned}$$

Explanation: the meanings and images of objects acted by IDXare reduced so that the simple features of objects are represented.

3.7 Clustering Operator CLU

$$\begin{aligned} & CLU\ ((c_1,e),\ldots,(c_m,e),(\{\},p_1),\ldots,(\{\},p_n)) \\ = & \{(c_i,p_j):1 \leqslant i \leqslant m, 1 \leqslant j \leqslant n\} \end{aligned}$$

Explanation: (c_i,e) represents physical object without any concept, and $(\{\ \},p_j)$ represents logical object which has pure meaning or concept. Clustering Operator makes physical objects have adequate logical meaning and concept.

3.8 Similar Operator SIM

$$SIM\ (X,Y)$$

$$= (SIM_m\ (X_m,Y_m)\ ,SIM_i\ (X_i,Y_i)\ )$$

Explanation: this operator can compare the similarity between two objects X and Y which include the similarity between two physical objects X_i and Y_i and the similarity between two logical objectsX_m and Y_m .

3.9 Existing Operator EXI

$$\begin{aligned} & EXI\ (X,Y) \\ = & (EXI_m\ (X_m,Y_m)\ ,EXI_i\ (X_i,Y_i)\ ) \end{aligned}$$

Explanation: this operator can tell us if there exists X in Y.

4. THE SYSTEM CONSTRUCTION

The old design method of railway location required that the engineers be imaginative and familiar with all sorts of technical standard. Now computers can substitute the engineers to design railway location based on above concepts and operators. When the decision to build a railway between city A and B has been made, first of all, the pictures of geography and geomorphology taken from airplanes are inputed into computer with an image scanner. The equal height map of the entensive area between A and B is established combining with other measure data, as shown in Fig 1.

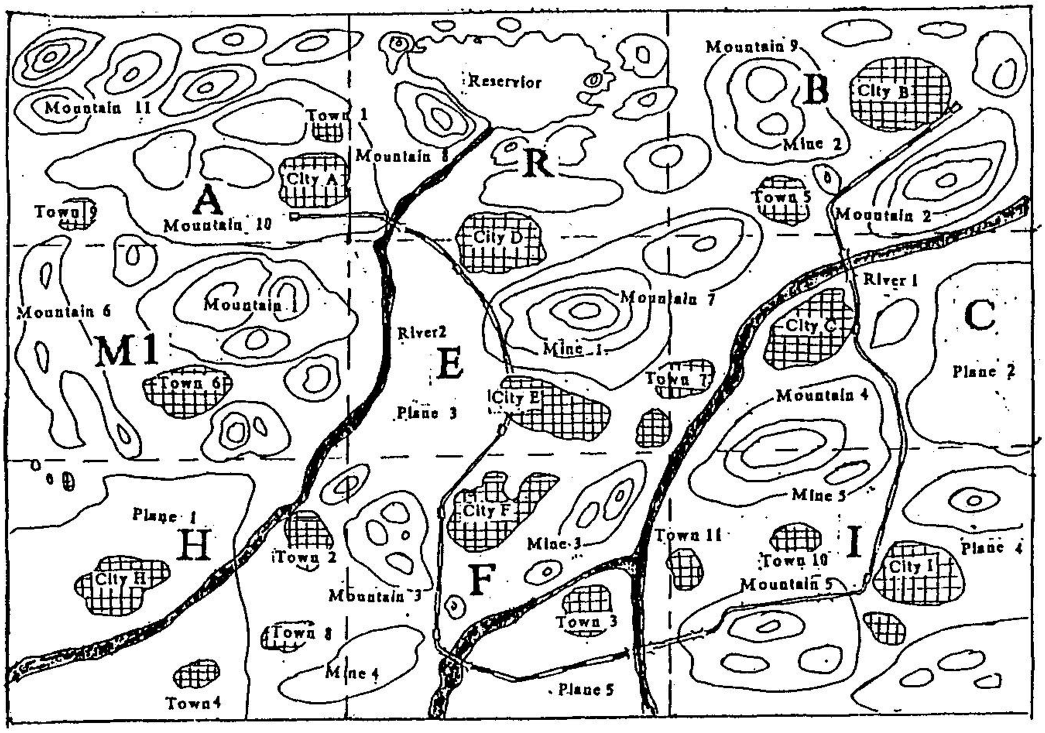


Fig.1 The map of an example processing.

Besides cities A and B, there are other seven cities (C, D, E, F, G, H, I), eleven towns, five planes, eleven mountains, two rivers and reservoir N, etc. According to typical features of regions, the area shown in Fig.1 is classified and segmented to form several regions. Each region is nominated with the most typical charateristics, and other features, such as geology coordinates, areas, natural resources, population, geology constructions and the product values of industry and agriculture, and so on, are its attributes. Thus we can use one point to represent one region segmented. The relations of points and regions are one–to–one mapping. The points are the corresponding abstraction form of the regions. If we think the regions as physical objects, the points are logical objects and the relations of the points represent the semantic descriptions of the regions. Following an example is given.

Region F is nominated as the name of city F after segmented, because city F is the important feature of region F. In the region F there are two mines (Mine 3 and Mine 4) , Mountain 3 , Plane 5 and River 2. They are the attributes of point F, shown in Fig. 3 and Fig.4.

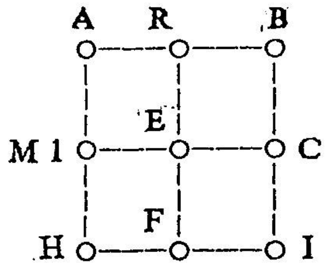


Fig.2 Semantic description net of Fig.1.

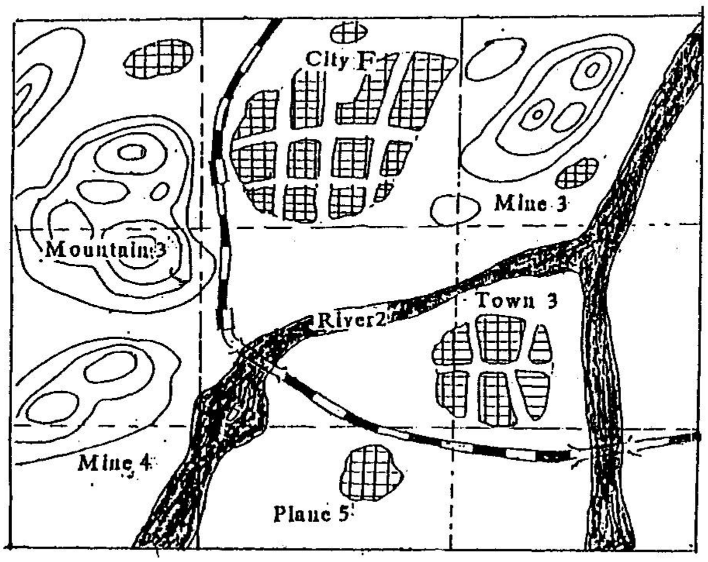


Fig.3 The detailed map of region F in Fig.1 at deeper level.

Region F

Name: F

Attributes:

Average Latitude: 40

Average Longitude:95

Average Height:400

(above see level)

Area:100 sq.km

Population:100,000

Mines: a coal and an iron

Industry: a steel factory

Mountain: a forest

Fig.4 The attributes representations of region F.

At certain abstraction level, the points form a semantic net in which every point is a node. The net can change dynamically based on the different levels. The attributes of subclasses at deeper level inherit the ones of superclasses at the fore level automatically and augment some if neccessary at the same time. The semantic nets take part in spatial analysis and sapatial reasoning and caluculate the meaning value to build a railway at one node and determine technical difficulty of building and engineering cost.

Fig.5 shows the construction of the system for designing railway location. The system consists of a large system knowledge-base which collects all kinds of information, such as geometry, graphy, hydrography and economy. Every kind of information has relative independence and completeness. In the system there are the capabilities of index and query. Under certain environment, we obtain information stored in the system knowledge-base, space constructions and logical relations and semantic descriptions of objects. The system realizes on SUN / 386 in C language. SUN CGI graphic interface system makes the system knowledge-base have the graphic information base.

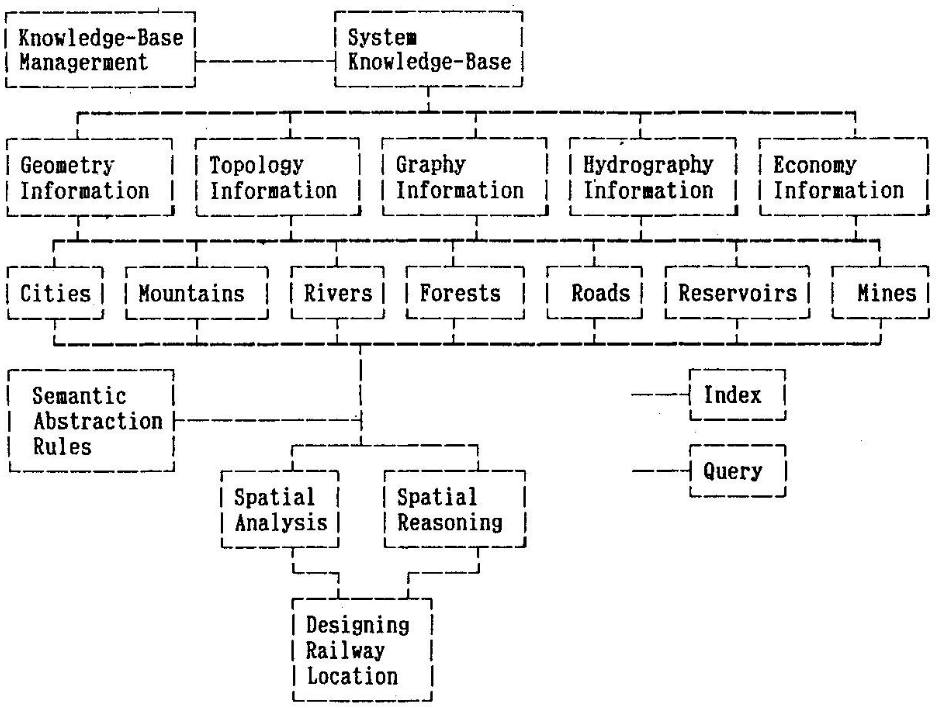


Fig.5 The construction graphy of the system.

5. CONCLUSION

We have described a new system for designing railway location. The system has some object–oriented characteristics which data abstrction, behaviour sharing, evolution and correctness. The semantic nets whose nodes represent objects are the system framework. The techniques introduced in the system are encapsulation, classification, flexible sharing and interpretation. The new system improves the old methods whose measuring, charting and reasoning mainly rely on human beings and decreases designing cost and labour power greatly. The theory and method of the system can be applied to other engineerings.

REFERENCES

1. CHANG S.K., Principles of Pictorial Information Systems Design, Prentice–Hall, Inc.,1989.
2. BLAIR G., GALLAGHER J., HUTCHISON D. and SHEPHERD D., Object–Oriented Languages, Systems and Applications, Longman Group UK limited, 1991.
3. MULLIN M., Object Oriented Program Design, Addison–Wesley Publishing Company, Inc., 1989.
4. CHANG S. K., Visual Reasoning for Information Retrieval from Very Large Database, Journal of Visual Languages and Computing, pp41–58, Vol.1, No.1, March 1990.
5. GOUTAS S., SOUPOS P., CHRISTODOULAKIS D., New Approach Towards an Object–Oriented Database System, Microprocessing and Microprogramming, Vol.27,No.1–5 August 1989.

Optimality Problem in Artificial Intelligence

Problème d'optimisation de l'intelligence artificielle

Optimierungsproblem in der künstlichen Intelligenz-Technologie

A. MIYAMURA
Associate Professor
Meijo University
Nagoya, Japan

Atsunori Miyamura, born 1937, received his doctoral degree in 1971 from Kyoto Univ. He has been working 20 years in structural eng. with a consulting firm.

Y. KOHAMA
Professor
Mie University
Tsu, Japan

Yoshiro Kohama, born 1945, received his doctoral degree in 1981 from Nagoya Univ., spent 10 years in research and education on structural engineering.

T. TAKADA
Research Associate
Mie University
Tsu, Japan

Toyofumi Takada, born 1967, received his Master degree in 1992 from Nagoya Univ. on discrete optimality on structural design. He dedicates further studies to this subject and reliability analysis.

SUMMARY

The combinatorial searching, an essence of Artificial Intelligence Technology, plays an important role in structural decision problems including the present failure load analysis of frame systems. Both generation of a combined failure mode and test of a failure load factor should be completed over all combinations between elementary failure modes with a result of explosive increase of burden on computer which can be, herein, improved effectively by means of a heuristic rule of the similarity index. Thus, the present method becomes a powerful tool when the mode approach is applied to reliability analysis or design.

RÉSUMÉ

La recherche combinatoire est une partie essentielle de la technologie de l'intelligence artificielle et joue un rôle important dans les problèmes impliquant des choix déterminants dans les ouvrages, par exemple dans la détermination de la charge de rupture des cadres. Il faudrait prendre en compte toutes les combinaisons possibles de mécanismes élémentaires, aussi bien dans la génération d'un mécanisme de rupture combinée que dans l'essai d'un facteur de charge ultime; ceci entraînerait un accroissement de type explosif du volume des calculs à l'ordinateur. Une règle heuristique de l'indice de similitude est plus efficace, grâce auquel la méthode des mécanismes devient plus performante dans l'analyse de fiabilité et dans le dimensionnement.

ZUSAMMENFASSUNG

Wesentlicher Bestandteil in der Technologie der künstlichen Intelligenz ist das kombinatorische Suchen. Es spielt auch eine wesentliche Rolle bei Entscheidungen im konstruktiven Ingenieurbau, etwa bei der Bestimmung der Grenztragfähigkeit von Rahmensystemen. Sowohl bei der Generierung eines kombinierten Versagensmechanismus als auch beim Test eines Grenzlastfaktors sollten alle denkbaren Kombinationen der Elementarmechanismen einbezogen werden, wodurch der Rechenaufwand explosionsartig grösser würde. Als effizientes Mittel hat sich eine heuristische Regel des Aehnlichkeitsindex erwiesen, mit der die Mechanismusmethode in der Zuverlässigkeitsanalyse und Bemessung sehr leistungsstark wird.

1. INTRODUCTION

The development of AI technology was triggered by Dartmouth Summer Conference in 1956 where the role of computer in the future was emphatically discussed. Subsequently, the important idea was realized such as GPS, LISP language and the frame theory. Furthermore, expert systems were developed such as Dendral at Stanford, MYCIN and Prospector which could get their successful position because of dealing with specified subjects though under enormous consumption of man power. The fifth generation project in Japan has aimed at high performance of inference of knowledge base. Through the development of AI technology the effective searching technique in an enormous database theoretically and practically was obviously a main subject to approach. Particularly, when the database consists of combination of elements, its space becomes growing so exponentially that search for optimality becomes significantly laborious and frequently almost impossible because of its nondifferentiability(Polak[1987]). Such combinatorial searching subject as the traveling salesman problem(Kernighan[1973]) can be found not only in application field of AI technology but many structural engineering analyses including the present failure load analysis. Many searching techniques for the combinatorial optimality are developed(Padberg[1987], Lin[1973], Johnson[1989]), among which the branch-and-bound method that is closely related to the dynamic programming is a generalized technique(Ibaraki[1991]). A large class of structural engineering design problems is also transcribed into the form of a nondifferentiable optimization problem with inequality constraints involving maximum function. When dealt with such nondifferentiable optimal problems(discrete optimization), the exhaustive enumeration including the generate-and-test procedure should be inevitably required. This is partly due to a lack of information of extrapolation on a searching space which is explosively enormous practically. It is laborious to describe algorithm of exhaustive enumeration in procedural language such as FORTRAN. On the contrary, declarative languages including Prolog(Clocksin[1984]) can handle it directly. Some important properties of Prolog are backtracking and nondeterminism to search for prescribed goal. The transitivity and inheritance inferences extend the searching efficiency so to large extent that combinatorial problems are more practically approached(Corkill[1983], Fennell[1977]). Thus, regarding nondifferentiable combinatorial optimality problem to accomplish an effective searching for an appropriate goal is equivalent to establishment of the pruning-futile-alternative technique including the branch-and-bound method, the heuristic approach and the qualitative reasoning that can realize sub-dimensionalization of searching space with a result of its rapid shrinkage. Frequently they are applied interactively. Unfortunately such pruning technique depends heavily upon particularity of the problem. Thus, an attempt of its generalization tends to lose sharpness of their efficiency as a result. Herein, the heuristics implies in wide sense a pruning technique to reduce the amount of generate-and-test drastically. Furthermore, the fact is that the rigorous goal cannot be necessarily attained even by the laborious generate-and-test method unless certain problem-oriented pruning technique is applied or unless the problem is relaxed into searching feasible goals. In general, the branch-and-bound method that belongs to exhaustive enumeration methods is applied with the aid of effective algorithms such as depth-first, best-bound and heuristic algorithm. Since the problem-oriented technique or the heuristic algorithm that can prune futile alternatives depends largely upon particularity of the problem and hence incidental human flair, the systematic development of heuristic algorithm becomes almost impossible. Recent conspicuous approaches such as the simulated annealing and neural network technique are mooted with considerable success(Hopfield[1985]).

Failure load analysis of structural systems from kinematically admissible field belongs to a typical combinatorial searching problem. Watwood[1979] proposes the generation of elementary mechanisms and their linear combinations of frames by the linear programming technique. Gorman[1981] presents an automatic method to generate the failure mode equations for all possible failure modes. Systematic generation of failure modes is important, because to add further constraints such as minimum weight criteria and reliability threshold(Henley[5]) more realistic description of structural design can be attained(Ditlevsen[1984], Melchers[1985]). The present study deals with the failure load analysis of rigid-plastic frames by the upper bound theorem which shows a combinatorial problem. When a kinematically admissible mode is assumed, the virtual work equation provides the corresponding failure load factor, γ_j^i. After generation of elementary failure modes from kinematically admissible displacement fields by topology and geometry(Onodera[1967]) their linear combinations produce successively the remaining failure modes whose predominance should be tested. Thus, the present optimal problem can be described by minimization of the objective function or the virtual work equation of possible failure modes.

2. ESSENCE OF AI TECHNOLOGY - COMBINATORIAL OPTIMALITY

Practical implementation of discrete optimality requires any of enumeration approaches such as the dynamic programming technique(DP), the branch-and-bound method and the exhaustive enumeration. DP has limitation to combination problem to some extent(Miyamura[1992]). The branch-and-bound method is applicable to widely diversified problems. It is accepted as a method to transform the combinatorial problem, which is difficult to solve directly by recursive decomposing, into partial problems until a set of more simplified problems. These partial problems have less large number of parameters while the number of problems to solve demand large amount of computing time due to explosive increase of combination. The branch-and-bound method can be summarized as follows: First, initialization of both the tentative value of evaluation function equal to infinitive, $z = \infty$, and the active partial problem the original problem, P_o, to solve. Second, searching for a new active partial problem, P_i, or ending for no more active partial problem. Third, testing P_i and a new z is obtained for a upper bound of feasible solution, $\{x\}$. Lastly, branching of descendent partial problems to add the active space and searching.

Thus, the branch-and-bound method tells that when fail occurs by test for any solution generated from an active partial problem, then further branch operation is not required with a result of decrease of combinatorial generate-and-test. Regarding searching for a new partial problem, P_i, this can be attained by the following two criteria: First, when the optimal solution is obtained from a partial problem, P_i, it is not necessary to deploy further branch operation. Second, if a partial problem cannot provide optimal solution of the original problem, it is not necessary to extend further branch operation.

These two criteria to halt further branch operation is the bound operation that thus can terminate the partial problem, P_i. Practical implementation of the bound operation can be made by either the lower bound test based upon relationship between optimal solutions from relaxed problems and admissible solutions or the dominance test based upon binary relationship of the evaluation function and constraint between two partial problems, P_k and P_l. The conventional exhaustive enumeration or blind searching corresponds to the case that the evaluation fuction, $f(\{x\})$ can be calculated after completion of branch operations and a set of feasible solutions are obtained from parameter vector including the optimal solution.

The present failure load analysis relates closely to the combinatorial optimality problem in the sense that a minimum load factor should be searched between possible kinematically admissible fields or failure modes including linear combination modes of elementary modes. Conventional LP(linear programming method) requires combination of k elementary modes to determine mode weighting coefficient, C_i, to optimize an evaluation function, where $C_i \neq 0$ for N active modes and $C_i = 0$ for non-active modes. Thus, combinatorial searching is accomplished for any N modes from k modes, and a memory size of combination defined by numbers of both elementary mode and member become practically enormous. On the contrary, the branch-and-bound method does not necessarily require a large memory size for searching optimality, when effective rules, frequently from heuristic knowledge, can bound non-active searching trees or descent futile alternatives. Herein, two bounding rules or heuristics are applied: generation of complete failure mode with one degree-of-freedom by the recursive expression of combination and pruning by the similarity index, S_{ij}, that can estimate similarity of plastic hinge distribution between two failure modes. Between the present generate-and-test technique and the conventional branch-and-bound method there is a significant difference: the evaluation function from virtual work equation cannot guarantee monotony. This suggests necessity of exhaustive enumeration of a larger combination space. However, any even higher order combination requires at least to possess a common plastic hinge between combined modes. This becomes less possible for the higher order combinations, which is empirically recognized from numerical simulations.

3. COMBINED MODE AND HEURISTICS

The present generate-and-test consists of both generation of failure mode and test of failure load factor. Thus, the generate process is classified into two categories: generation of elementary failure modes and their linear combinations. When a rigid-plastic plane frame with m-members, n-nodes and n_r-fixed supports collapses with plastic hinges subject to nodal loading, elementary failure modes, whose number is $(3n - m)$, can be expressed as follows:

$$\{\tau_p\} = [C]\{_I\tau_{pt}\} \tag{1}$$

where $\{\tau_p\}$ means the plastic hinge rotation vector at ends of member, $\{_I\tau_{pt}\}$, the corresponding independent hinge rotation vector of tree members, respectively. The displacements at nodal point can be expressed by means of the path matrix, $[H]$:

$$\begin{aligned} \{D_x\} &= [H][\mu][L][_cH_{mt}]\{\tau_{pt}\} \\ \{D_y\} &= -[H][\lambda][L][_cH_{mt}]\{\tau_{pt}\} \end{aligned} \tag{2}$$

where $\{D_x\}$ and $\{D_y\}$ mean nodal displacement vectors in x- and y-directions, respectively. After generation of elementary failure modes a combined mode is expressed by their linear combination. To implement the generate-and-test effectively it is preferable to describe the combination process in recursive form. When both the hinge rotation vector and nodal displacement vector of the i-th elementary mode is expressed by $\{Y_i\} = \{\{\tau_p\}_i^t, \{D\}_i^t\}$, the combination of any two elementary modes, $\{Y_i\}$ and $\{Y_j\}$, becomes:

$$C_2(\{Y_i\}, \{Y_j\}|\tau_{p,s}) = A_i\{Y_i\} + A_j\{Y_j\} \tag{3}$$

$$A_i\tau_{pi,s} + A_j\tau_{pj,s} = 0 \tag{4}$$

where $\tau_{pi,s}$ and $\tau_{pj,s}$ mean hinge rotations common to both the i- and j-th modes at the critical section, S. The lefthandside of Eq.(3) means the resulting mode combined $\{Y_i\}$ with $\{Y_j\}$, which are not plastic at the critical section. Extending Eq.(3) to combination of the other modes, the following recursive expression can be obtained:

$$\begin{aligned} C_k(\{Y_k\}, \{Y_{k-1}\}, \ldots, \{Y_1\}|\tau_{p,k-1}, \tau_{p,k-2}, \ldots, \tau_{p,1}) &= \\ A_I C_{k-1}^I(\{Y_{k-1}\}, \{Y_{k-2}\}, \ldots, \{Y_1\}|\tau_{p,k-2}, \tau_{p,k-3}, \ldots, \tau_{p,1}) &+ \\ A_{II} C_{k-1}^{II}(\{Y_k\}, \{Y_{k-2}\}, \ldots, \{Y_1\}|\tau_{p,k-2}, \tau_{p,k-3}, \ldots, \tau_{p,1}) & \end{aligned} \tag{5}$$

$$A_I\tau_{p,k-1}^I + A_{II}\tau_{p,k-1}^{II} = 0 \tag{6}$$

where $\tau_{p,k-1}^I$ and $\tau_{p,k-1}^{II}$ are the hinge rotations at the $(k-1)$-th critical section common to the $(k-1)$-th combined failure modes, $C_{k-1}^I(\cdot)$ and $C_{k-1}^{II}(\cdot)$. Eq.(5) shows that the k-th combined mode can be decomposed into two $(k-1)$-th modes each of which has the same tail of minus-one order but different head. The recursive expression thus generalized ensures both easy composition and decomposition of failure modes by a simple algorithm. Subsequently, the test procedure should be implemented by evaluation of a failure load factor, γ_j, which is given by the following virtual work equation for a generated failure mode:

$$\gamma_j = \sum_{k=1,2,\cdots,n} C_k\{\tau_{pk}\}^t\{M_p\} / \sum_{k=1,2,\cdots,n} C_k\{D_k\}^t\{P\} \rightarrow min \tag{7}$$

where C_k means a weighing coefficient of the k-th mode from n elementary failure modes $(0 \leq C_k \leq 1)$. Summation is implemented for any number of combinations less than that of n elementary modes. $\{\tau_{pk}\}$ means the hinge rotation vector, $\{M_p\}$, the member yielding resistance vector, $\{D_k\}$, the nodal displacement vector, and $\{P\}$, the external nodal loading ratio vector, respectively. Eq.(7) shows that the goal failure load factor, γ_{cr}, is the lowest factor derived from possible failure modes given by both elementary failure modes and their linear combinations. This implies that for a number of elementary failure modes their combinations become exponentially increasing, which is subjected to a combinatorial searching technique.

In order to implement effective generation-and-test of failure modes it is necessary to develop certain heuristic rules which can prune futile alternatives. In the following the similarity index is used as a heuristic rule. It is difficult to implement effective search of predominant modes either by the exhaustive enumeration method with the generate-and-test or the conventional approximate searching approaches with reduced search space by the depth-first or the breadth-first. This depends upon the fact that the more test of predominance is necessary for the larger number of generation of modes. Thus, if an approximate estimation of effective combination is implemented before actual combination procedure the amount of calculation decreases drastically even for a large sized structural systems. A combination of the smaller internal virtual work to the external one becomes predominant. Hence, when plastic hinge rotations decrease by the combination of appropriate modes the corresponding internal work decreases. This can be more accomplished for two candidate modes whose common hinges become larger in number, in other words, whose hinge distribution becomes more similar. Thus, pruning of the futile searching space is attained if it is possible to evaluate an extent of similarity of hinge distribution with less burden. Such evaluation is established by enumeration of both common and non-common plastic hinges between the i- and j-th modes, and satisfied to some extent by the following similarity index, S_{ij}:

$$S_{ij} = \sum_s min(h_{is},\ h_{js}) / \sum_s max(h_{is},\ h_{js}); \quad i,j = 1,\ldots,k \tag{8}$$

where h_{is} is a binary parameter given by:

$$h_{is} = \begin{cases} 0, & \text{if } \tau_{pis} = 0; \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

Eq.(8) provides the ratio of the numbers of common and non-common hinges between any two modes. Thus, $h_{is} = 1$ corresponds to a plastic hinge at the s section in the i-th failure mode. Furthermore, $0 \le S_{ij} \le 1$ is valid. Applying Eq.(8) to all of k elementary failure modes the plastic similarity index matrix, $k \times k$, can be obtained which is reflexive and symmetric but not transitive like of the fuzzy similarity. Consequently the present heuristics becomes: First, before implementation of the conventional generate-and-test for the exhaustive enumeration the similarity index, S_{ij}, by Eq.(8) should be firstly evaluated and pruned if not tolerable. Empirically, $0.3 \sim 0.4 \le S_{ij}$ is preferable. This is applicable to further higher order combination modes whose similarity indices are easily evaluated by Eq.(8) recursively with substitution of both $i = n$ and $j = n-1$. Second, whenever there exists no common plastic hinge for combination of more than three modes by Eqs.(5) and (6), further searching can bound even with $S_{ij} \neq 0$.

4. NUMERICAL SIMULATION

A 12 story, 3 bay rectangular frame with 120 members subjected to vertical and horizontal proportional loading(Fig.1) is analyzed by the present method that is described in Prolog language on PC9801 personal computer. Prolog predicate has non-determinism by its backtracking ability which easily generate combination modes and automatically implement branching operation subject to generation rule. As a side-effect due to non-determinism a number of futile alternatives(combinations) appear, and should be pruned. It is advantageous to avoid floating calculation as far as possible. Furthermore, Prolog predicates of recursive rule with non-determinism can play role of both the generate and the test, which is significantly effective. Fig. 1 shows a typical combinatorial searching from 132 elementary modes around the optimal combination with $\gamma_{opt} = 3.176$. However, generation of elementary modes is irrelevant to loading condition which can change order of γ corresponding to the elementary modes. This implies that the elementary mode that provides the lowest load factor between elementary modes is expected to participate combinations which include γ_{opt} or its vicinities. The present heuristic bounding by S_{ij} limitation can effectively prune futile alternatives(modes) although it does not guarantee optimality. Thus, this heuristics provides an upper bound, and is effective for lower order combinations such as two-mode combination. S_{ij} becomes smaller with higher order combination with an elementary mode. The heuristic bounding by Eq.(5) prune combination that has not at least a common hinge even with $S_{ij} \neq 0$ (Note that the combination of elementary modes, [1+49+55], becomes fail in Fig.1). This becomes more prominent when the

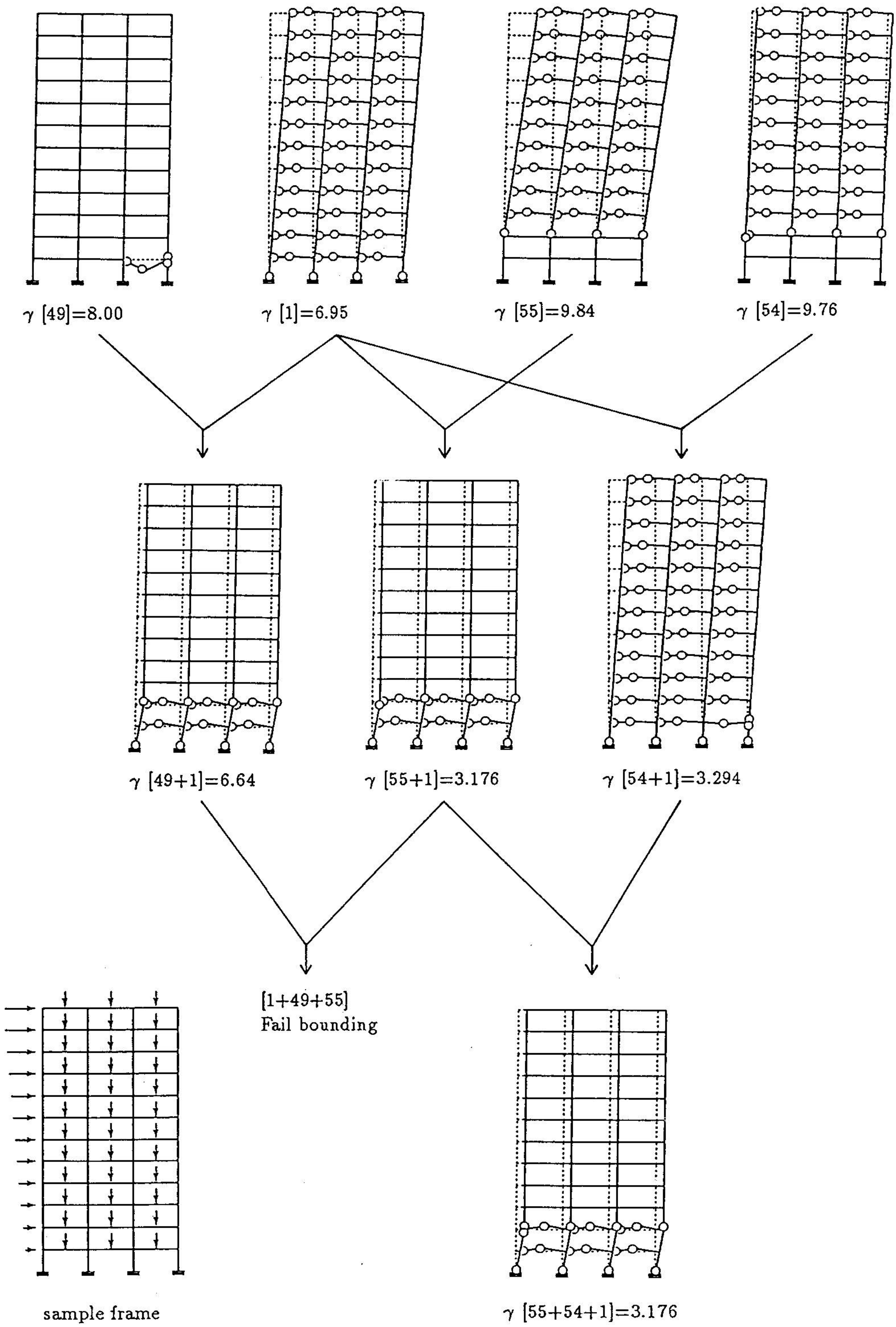


Fig.1 Example and failure modes

order of combination becomes higher. Numerical results suggest superiority of the depth-first combination from elementary modes in order of ascent of γ.

5. CONCLUDING REMARKS

The failure load analysis based on statically admissible field is a typical combinatorial optimality problem, which can be approached by the generate-and-test with heuristics in AI technology. Hence, the following concluding remarks are obtained:

a) The present searching is implemented on a multi-branch tree so that the corresponding generate-and-test can be accomplished by parallel procedure. Consequently, such declarative language as PARALOG is expected more drastic acceleration of searching for practical system.
b) Practically LP requires a larger memory size. While the present method can generate predominant modes (smaller γ_i) with a smaller memory size that are applicable to reliability analysis by the mode approach.
c) Although the present evaluation function by Eq.(7) does not guarantee monotony after recursive combination by Eq.(5), its pruning can realize significant decrease of searching space.
d) It is expected that topological measurement of frames can accelerate further pruning of futile alternatives.
e) $S_{ij} = 0$ corresponds to the exhaustive enumeration that can provide the optimal solution or the lowest load factor. Practically, to save computing time a tolerable value is taken with a result of upper bound.

REFERENCES

(1) R. Onodera and H. Orui[1967], Foundational Theory of Electrical Networks, Gakuken-sha, Tokyo
(2) S. Lin and B. Kernighan[1973], An efficient heuristic algorithm for the traveling salesman problem, Operations Research, Vol.21, pp.498-516
(3) R. D. Fennell and V. R. Lesser[1977], Parallelism in artificial intelligence problem solving: A case study of Hearsay II, IEEE Trans. on Computers, pp.98-111
(4) V. B. Watwood[1979], Mechanism Generation for Limit Analysis of Frames, J. Struct. Div., ASCE, 109, ST1, pp.1-15
(5) M. R. Gorman[1981], Automatic Generation of Collapse Mode Equations, Proc. ASCE, ST7, pp.1350-1354
(6) D. D. Corkill and V. R. Lesser[1983], The use of metalevel control for coordination in a distributed problem solving network, Proc. of the Eighth Intern. Joint Conf. on Artificial Intelligence, pp.748-756
(7) W. F. Clocksin and C. S. Mellish[1984], Programming in Prolog, Springer Verlag, New York
(8) O. Ditlevsen and P. Bjerager[1984], Reliability of Highly Redundant Plastic Structures, J. Eng. Mech., ASCE, 110, EM5, May, pp.671-693
(9) J. J. Hopfield and D. W. Tank[1985], Neural computation of decision in optimization problems, Biol. Cybern. 52 pp.141-152
(10) R. E. Melchers and L. K. Tang[1985], Failure Modes in Complex Stochastic System, Proc. 4th Int. Conf. of Structural Safety and Reliability, vol.1, pp.97-106
(11). E. Polak[1987], On the mathematical foundations of nondifferentiable optimization in engineering design, SIAM Review, 29-1, Vol 29, No.1, pp.21-89
(12) M. Padberg and G. Rinaldi[1987], Optimization of a 532-city symmetric traveling salesman problem by branch and cut, Operations Research Letters; Vol.6, pp.1-7
(13) D. S. Johnson, C. R. Aragon, L. A. McGeoch and C. S. Schevon[1989], Optimization by simulated annealing: An experimental evaluation; Part I, Graph partitioning, Operations Research, 37 pp.865-892
(14) T. Ibaraki[1991], Efficient Searches and Their Limitation, J. Japanese Soc. Artificial Intelligence, 6-1 pp.15-23
(15) A. Miyamura, Y. Kohama and T. Takada[1992], Application of heuristics to structural design problem, Proc. of Structural Engineering, AIJ, Vol. 38B, pp.1-7

APPENDIX

Eqs.(1) and (2) can be derived by topology and geometry of a frame as follows: Any member in a frame corresponds to an oriented edge and by introduction of an imaginary member at supports connecting to the fixed point, O, resulting in an oriented graph. The compatibility condition of rigid body displacements of each member at a failure state becomes the closing condition that the sum of rigid body rotations at a circuit in an oriented graph should be zero:

$$\begin{aligned}\sum_{circuit} du_{ij} &= \sum_{circuit} \mu_{ij}\psi_{ij}l_{ij} = 0 \\ \sum_{circuit} dv_{ij} &= \sum_{circuit} \lambda_{ij}\psi_{ij}l_{ij} = 0\end{aligned} \qquad (a)$$

where du_{ij} and dv_{ij} mean displacements of a member, i-j, in x- and y-directions due to rigid rotation, ψ_{ij}. λ_{ij} and μ_{ij} are the x- and y-direction cosines and l_{ij}, the member length, respectively. For all independent circuits the following relations are obtained:

$$\begin{aligned}[R][\mu][L]\{\psi\} &= 0 \\ [R][\lambda][L]\{\psi\} &= 0\end{aligned} \qquad (b)$$

where $\{\psi\}$ means the rigid body rotation vector and $[\mu]$, $[\lambda]$ and $[L]$, the diagonal matrices with elements, μ_{ij}, λ_{ij} and l_{ij}, respectively. A fundamental circuit matrix, $[R]$, has the following elements:

$$R(c,e) = \begin{cases} 1, & \text{when a fundamental circuit, c, includes an edge, e, positively,} \\ -1, & \text{when a fundamental circuit, c, includes an edge, e, negatively,} \\ 0, & \text{otherwise.} \end{cases} \qquad (c)$$

In Eq.(b) the rigid body rotations of imaginary members are assumed zero. The number of fundamental circuits becomes $(m-n)$, and the size of $[R]$, $(m-n)\times(m+n_r)$. The plastic hinge rotations at the ends of a member, i-j, correspond to $\tau_{pij} = \theta_i - \psi_{ij}$ and $\tau_{pji} = \theta_j - \psi_{ij}$. By applying the connection matrix, $[D_m]$, of the expanded graph with introduction of a new node at the middle of an edge this rotation vector becomes:

$$\{\tau_p\} = [D_m]^t\{\{\theta\}^t, \{\psi\}^t\} \qquad (d)$$

where $\{\theta\}$ means the nodal rotation vector. The connection matrix, $[D_m]$ has the following elements:

$$D_m(v,e) = \begin{cases} 1, & \text{when an edge, } e\text{, leaves a node, } v, \\ -1, & \text{when an edge, } e\text{, enters a node, } v, \\ 0, & \text{otherwise.} \end{cases} \qquad (e)$$

Since the orthogonality, $[R_m][D_m]^t = [0]$, is valid for the circuit and connection matrices, by premultiplying the fundamental circuit matrix to Eq.(d) the following expression is derived:

$$[R_m]\{\tau_p\} = \{0\} \qquad (f)$$

Eq.(f) implies thus the compatibility of the vector, $\{\tau_p\}$, the closing condition that the sum of rotation at plastic hinges in a fundamental circuit becomes zero. When a set of edges of the expanded graph are separated into those of trees and cotrees, the path matrix, $[H_{mt}]$, can be derived. Since a fundamental circuit consists of the tree and cotrees, the relation, $[H_{mt}][D_m]^t = [E]$, where $[E]$ means a unit matrix, is defined. Thus, by premultiplying $[H_{mt}]$ to Eq.(d), the following equation is obtained:

$$[H_{mt}]\{\tau_{pt}\} = \begin{bmatrix} [_jH_{mt}] \\ [_cH_{mt}] \end{bmatrix} \{\tau_{pt}\} = \begin{Bmatrix} \{\theta\} \\ \{\psi\} \end{Bmatrix} \qquad (g)$$

where $[_jH_{mt}]$ and $[_cH_{mt}]$ mean the path matrices from the fixed point to a node and to a middle point of an edge on the tree of the expanded graph, respectively. $\{\tau_p\} = \{\{\tau_{pt}\}, \{\tau_{p\bar{t}}\}\}$, $\{\tau_{pt}\}$ and

$\{\tau_{p\bar{t}}\}$ mean plastic hinge rotation vectors corresponding to the tree and the cotree, respectively. Substitution of Eq.(g) into Eq.(b) describes that Eq.(b) has $(3n-m)$ independent solutions. Gaussian elimination provides the following relation:

$$\{{}_D\tau_{pt}\} = [C_I]\{{}_I\tau_{pt}\} \tag{h}$$

where $\{\tau_{pt}\} = \{\{{}_D\tau_{pt}\}^t, \{{}_I\tau_{pt}\}^t\}^t$. Furthermore, when partitioned such $[R_m] = [[R_{mt}], [R_{m\bar{t}}]] = [[R_{mt}], [E]]$ and $[R_{mt}] = [[{}_DR_{mt}], [{}_IR_{mt}]]$, Eqs.(f) and (h) give:

$$\{\tau_p\} = \begin{Bmatrix} \{\tau_{p\bar{t}}\} \\ \{{}_D\tau_{pt}\} \\ \{{}_I\tau_{pt}\} \end{Bmatrix} = \begin{bmatrix} -[{}_DR_{mt}][C_I] - [{}_IR_{mt}] \\ [C_I] \\ [E] \end{bmatrix} \{{}_I\tau_{pt}\} = [C]\{{}_I\tau_{pt}\} \tag{1}$$

The size of $[C]$ is $2m \times (3n-m)$, whose column vector, $\{C_i\}$, means the corresponding hinge rotations to ${}_I\tau_{pti} = 1$. Eq.(1) implies that the number of hinges with non-zero rotation cannot exceed $3(m-n)+1$, which is equal to the degree of redundancy plus one. The column elements of $[C]$ are independent of each other, and deformation elements thus expressed contribute directly to an elementary failure modes. The displacements at nodal points can be expressed by means of the path matrix, $[H]$:

$$\begin{aligned} \{D_x\} &= [H][\mu][L][{}_cH_{mt}]\{\tau_{pt}\} \\ \{D_y\} &= -[H][\lambda][L][{}_cH_{mt}]\{\tau_{pt}\} \end{aligned} \tag{2}$$

where $\{D_x\}$ and $\{D_y\}$ mean nodal displacement vectors in x- and y-directions, respectively.

Machine Learning in Blackboard System for Steel Structures

Apprentissage automatique dans la construction métallique

Maschinen-Lernen nach dem Schultafelsystem im Stahlbau

K.B. AJI
Research Scholar
IIT Madras
Madras, India

V. KALYANARAMAN
Professor
IIT Madras
Madras, India

K.B. Aji, born 1962, graduated from REC Calicut, Kerala. Currently he is a doctoral student and a Senior Project officer in a project on expert Systems for steel struct. in the Civil Eng. Dep. at IIT Madras.

V. Kalyanaraman, born 1947, got his Bachelor in Civil Eng. at IIT Madras and PhD at Cornell Univ. Currently he is Prof. at IIT Madras specialising in stability, steel struct. and CAD.

SUMMARY

Machine learning paradigms in the recent decade have made considerable strides in the area of Artificia Intelligence. Eventhough structural engineering domain is a fertile ground for using these paradigms to improve engineering process, the literature in this area are only a few. This paper describes appropriate machine learning strageties for implementation in an integrated engineering system for knowledge based engineering of steel structures.

RÉSUMÉ

Au cours de la dernière décennie, une évolution paradigmatique considérable a eu lieu dans la conception de l'apprentissage automatique relatif au domaine de l'intelligence artificielle. Bien que la technique de la construction soit un milieu fertile pour utiliser ces paradigmes en vue de perfectionner les processus d'études, la littérature sur ce sujet reste limitée. Cet article décrit des stratégies adéquates d'apprentissage automatique pour être appliquées dans un système d'études intégrées destiné à la construction métallique.

ZUSAMMENFASSUNG

In Gestalt des Maschinen-Lernens hat sich im letzten Jahrzehnt mit grossen Schritten ein Paradigmenwechsel auf dem Gebiet der künstlichen Intelligenz vollzogen. Obwohl der konstruktive Ingenieurbau sich für Verbesserungen im Entwurfsprozess durch derlei Konzepte anbietet, gibt es nur wenig Literatur darüber. Der Beitrag beschreibt geeignete Strategien des Maschinen-Lernens zur Implementierung in ein integriertes Entwurfssystem für Stahlbauten.

1. INTRODUCTION

Machine learning enables a system to perform the same task or a task drawn from the same population more efficiently and effectively the next time [1]. Objectives of research on machine learning may be one of the following: i) simulate and thereby understand and improve human learning process, ii) develop natural language processing capabilities to serve as interface between man and machine, iii) improve problem solving skills of computing and iv) enhance learning from discovery.

The motivation for research which forms the basis for this paper, is to improve the problem solving capability of computer aided engineering systems by machine learning. Understanding the role and the application of machine learning strategies would facilitate acquisition of new knowledge, efficient reorganization of the existing knowledge, faster and better solution, expansion of the problem solving capabilities, learning of control knowledge, simulation of creative problem solving, and efficient solution even under uncertain and incomplete problem specification.

This paper deals with the machine learning strategies for computer aided engineering of steel structures in an extended blackboard system developed in a project on knowledge based expert system for integrated engineering of steel structures. Initially, generic paradigms in AI for machine learning are briefly reviewed in order to introduce the state of the art. Subsequently literature on the application of machine learning techniques in the civil engineering domain are discussed. It is shown that both the Machine learning techniques and applications are yet to deal with the needs of a large engineering domain. Finally the opportunities for and issues in machine learning in the engineering of steel structures are discussed and appropriate learning strategies are evolved for such a system. The discussion is illustrated with a few examples.

2. MACHINE LEARNING

AI research on machine learning over the past few decades has led to four well accepted machine learning paradigms, namely, inductive learning, analytic learning, genetic algorithms (classifier systems) and connectionist learning methods [2].

2.1 Inductive Learning

Formulation of plausible general assertion that explain given facts and prediction of new facts based on these general assertions is induction. Induction is an essential component of human learning. We induce a concept from a series of observations of a process or a phenomenon. Thus inductive learning involves the formation of a concept from examples and counter examples. In general induction can be either a single-shot process based on initial training examples or an incremental one. Induction is by far the most widely studied paradigm [3,4,5]. Gennari et al. [6] have identified the common features in induction learning such as unsupervised learning, incremental learning, integrated with performance, top down classification and incremental hill climbing.

The programs based on induction can handle inputs represented in a specific manner, such as attribute value pairs. This requires large scale structuring of the knowledge and hence limits the scope of the learning task. Moreover the learning is highly empirical, which constrains the extent of knowledge that can be learnt.

2.2 Analytic Learning

Analytic learning methods are deductive in nature and use the past experience in problem solving to arrive at the solution. These methods are superior to inductive methods as they can provide explanation for the classification of instances. The major contributions to this paradigm are in the areas of analogical reasoning [7,8,9], case based reasoning [10,11] and explanation based learning [12,13].

Analogical reasoning [14] consists of transferring knowledge from past problem solving episodes to new problems that share significant aspects with corresponding old experience and using it to construct solutions to the new problems. Case based reasoning also involves drawing conclusions from problems solved in the past to use in new problems. This kind of reminding of old experience [10] in the form of explanation can be processed by an EBL mechanism to generate new solutions. Explanation based learning involves generalizing the explanation obtained from an instance. Thus EBL produces a description of a concept based on the domain theory, which explains a particular instance of that concept.

2.3 Classifier Learning

Classifier systems are massively parallel, message passing, rule based systems that learn through credit assignment and rule discovery [15,16]. The algorithm used for rule discovery is analogous to the biological mutation process and hence the name genetic algorithm is also used for these systems. The learning process is closely similar to the inductive mechanisms and the connectionist methods. Although the nature of learning is highly empirical, under complex environments characterized by noisy and incomplete data this methods offers a viable alternative for learning.

2.4 Connectionist Learning

Connectionist methods, (also known as neural networks) emulate the function of mammalian brain. Typically a neural network [17, 18] consists of three different layers namely, the input layer, the hidden layer and the output layer. Each layer consist of a group of processing elements characterised by their weights. These processing elements enable the network to map the internal representation of a problem by suitably modifying their weights to match the input-output patterns. A concept can be represented over the entire network (distributed representation) or represented at a local level (localised representation). Once the network is trained with sufficiently large number of examples, it can generate solution to new problems. This method is highly suitable for parallel processing and is promising for future computing requirements. However, the requirement of large number of examples for training and the slow rate of convergence [17] for complex problems makes it unsuitable for many real world applications at this time.

In addition to these four major paradigms there are other sub paradigms such as learning by discovery [20] learning by experimentation [21], and learning by instruction [22], which are not studied extensively to derive useful applications. A more detailed treatment of the various machine learning paradigms is presented by Carbonell [2] and in the other papers [6, 9, 11, 15, 17] in the particular special issue of the journal. Carbonell concludes that connectionist paradigms are appropriate for learning in unstructured continuous domains with many training examples. Analytical paradigms at the other end are best suited in domains with rich structured knowledge even if only a few examples are available. Inductive and classifier systems bridge the gap between these two extremes.

3. CIVIL ENGINEERING APPLICATIONS

Literature on the application of machine learning in the civil engineering domain problem are very few. Rooney and Smith [22] discussed a feed back mechanism based model to two case studies covering the design of single span simply supported wide flange beams. Similarly many researchers have resorted to storing non-synthesized data from past experiences in a database for future reference. Such techniques are practically useless when the past experience is not much and becomes computationally inefficient when the number of stored examples increase.

Maher and Li [23] have demonstrated learning of default values, ranges for variables, relationship among numerical valued variables and patterns among nominal valued variables, using dependency network of earlier problem solving experience. However, the problem of when and how the decisions are made to perform the learning is not addressed in the paper.

Navinchandra et al. [24] have illustrated the role of analogues, heuristic rules, observed effects and engineering principles in problem solving through an example of a lever problem. The learning algorithm illustrated in the paper is conceptual and can not be extended to serious engineering application readily. Zhao et al. [25] used transformational analogy and similarity metric to retrieve solution to new problems from closely matching building examples in database. Murlidharan et al. [26] have used learning algorithms based on induction and analogical reasoning. These strategies create only a database that reduces the subsequent search space used to generate alternate configuration.

Arciszewski and Ziarko [27] have presented rough sets approach to inductive learning in civil engineering. The system extracts decision rules which can be used to acquire knowledge for problem solving to develop shallow model, to identify governing rules in a domain and to develop learning expert systems. Yeh et al. [28] have used the ID3 inductive learning algorithm to acquire diagnostic knowledge about the damage to PC piles while driving.

Adeli and Yeh [29] have demonstrated perceptron learning model for simple engineering design. This algorithm works for very simple tasks, which are trivial in engineering design, whereas this algorithm can not learn complex tasks since there are no hidden layers. Kamarthi et al. [30] have demonstrated a neural network learning system for vertical formwork selection. The paper discusses the merits and demerits of the neural network system when compared to rule based system and demonstrates that the difficulty of eliciting knowledge for rule based system can be overcome by the neural network learning system. Moselhi et al. [31] have illustrated neural network applications in the field of bidding for construction projects.

The examples of application of machine learning in the civil engineering domain clearly illustrate growing capability and complexity. Inductive learning methods are the most frequently used. Applications using neural network methods are being explored more recently. Applications in analytical methods and classifier systems are least explored, probably due to their computational complexity and application interface problems. It is also clear that applications so far discussed deal with only narrow domains of engineering problems.

4. MACHINE LEARNING IN ENGINEERING OF STEEL STRUCTURES

According to Simon [1] large knowledge based AI systems, particularly systems that can be expected to continue to grow and accumulate over a period of years of use, are fertile areas of application of machine learning. Engineering of steel structures is a large problem domain involving conceptual design, structural system planning, preliminary sizing, detailed analysis, design, document preparation and construction planning. The attributes in the domain represent the solution at various levels of a abstraction. Furthermore, the development of CAD system in the domain is incremental involving group effort. The system should model and accommodate the cooperative problem solving behaviour of domain experts working together. The machine learning in such an environment should be able to handle the varied requirements of the large domain. An integrated engineering system (IES) for the knowledge based engineering of steel structures has been already developed [32] under an ongoing project. The development and implementation of machine learning strategy in this system is currently under progress. The details of machine learning in this system are discussed in the following sections.

4.1 IES: Integrated Engineering System

IES uses an extended blackboard shell. Before discussing the machine learning implementation on this system, basic features of the systems are briefly reviewed [32]. Fig.1 shows the architecture of IES. The knowledge represent various functional activities of the engineering process, are compiled as production rules in independent knowledge sources. The knowledge sources generally do not interact directly but only through the global data referred to as blackboard. The blackboard has two panels namely solution blackboard and Control blackboard. The solution blackboard contains hierarchy of objects of the solution space with named links for inheritance. Objects and their attributes are represented as frames. Instances of the objects are stored in a relational database with links to blackboard objects. The control blackboard contains the status of the abstracted events of the solution process. Since the engineering process involves a large number of computations which is more efficiently carried out using algorithmic programs, C functions are used for such procedural programs. These function can be called from production rules in the knowledge sources. Generation of dependency network which is used in knowledge based backtracking, domain specific knowledge based control, opportunistic scheduling of knowledge sources are the other features of the system. More details about the system are presented by Sakthivel et al. [32].

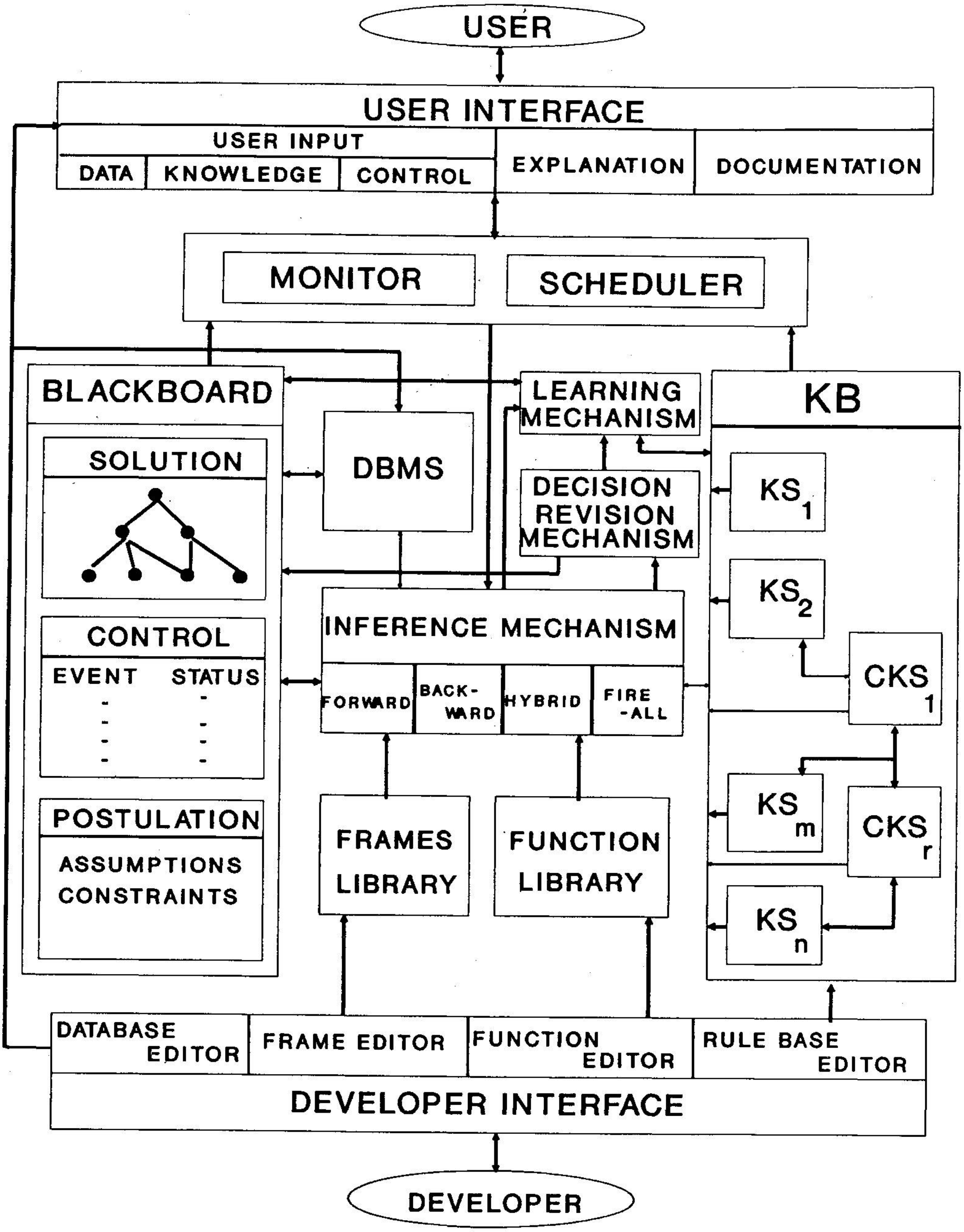


FIG.1. INTEGRATED ENGINEERING SYSTEM

4.2 Scope for Machine Learning

In large systems it is neither desirable nor feasible to consider machine learning as the backbone of the system. In engineering domain whenever problem solving steps and the engineering fundamentals that form the basis of the problem are well understood, it is efficient to represent such knowledge algorithmically in procedural programs. These are segments where governing knowledge is clearly defined or is easily acquired. However, the sequence of application of the knowledge during problem solving may be either not clear or has to be flexible. Knowledge based approach is more appropriate under such circumstances wherein the knowledge may be represented in production rules, frames, semantic networks, etc. Trying to acquire such knowledge for problem solving through machine learning is devious, unproductive and inefficient. However, scope exists for machine learning in engineering problem solving. Frequently knowledge or expertise is difficult to acquire and codify in which case machine learning from earlier problem solving experience can be of immense help. Besides, adaptation and modification of theory and practice is a continuing process in engineering problem solving. Engineering solution is affected by temporal, geographical and economic factors in non-obvious ways. Machine learning capabilities could synthesize such knowledge from past experience and help the system to adapt to changes in the theory and practice.

4.2.1 Engineering Tasks and Learning Strategies

In this section we discuss specific machine learning strategies that are being tried at various stages of engineering problem solving using IES. The preliminary specification of a structural engineering problem is brief open ended and ill-structured. Conceptual design based on this problem statement leads to an appropriate structural system, such as the type of bridge appropriate for a given specification being a cable stayed bridge or a truss bridge, etc. Decisions made at this stage have probably the greatest impact on the final economy of the engineering solution. However, the knowledge that drives the conceptual design and the application of the knowledge to arrive at appropriate decisions are not well understood. It is usually difficult to acquire the conceptual design knowledge. Analytic paradigms, such as case based learning or derivational analogy, are appropriate strategies for learning conceptual design. A few learning examples along with rich underlying domain theory support the learning task. Conceptual design is highly sensitive to temporal and geographical conditions. Hence a continuous learning system which could pursue multiple solution path and learn from each problem solving episode would be more robust. The frame based representation of objects, the dynamic instantiation of the objects in the solution space and events in the control blackboard, as well as knowledge based control strategy are the features of the IES system, which readily support the analytic learning strategy.

Having decided on the structural system, planning and configuring the structural subsystems is the next step in the engineering process, which offers opportunities for the machine learning process. Maher and Li [23] have demonstrated conceptually, a learning system for configuring cable stayed bridges based on the dependency network of the design experience. Inductive paradigms such as conceptual clustering using a sequence of known examples and counter examples from previous problem solving sessions support this process. This paradigm is being tried for the structural configuring activity in IES.

Let us consider the task of learning the configuration generation of a cable stayed bridge from a number of cases already engineered. The attributes that define the configuration of a cable stayed bridge may be subdivided into problem specification attributes and configuration attributes to be generated by the system. The specification attributes are the total length of bridge, number of lanes of traffic, geotechnical details of the site, navigational requirements under the bridge, wind and earthquake load at the site, approach alignment, and aesthetic requirements. The configuration attributes are the number of cable stayed spans, maximum span length, side span length, drop span, tower type, tower height, number of cable planes, inclination of cable planes, number of cables per span, cable arrangement, girder type, girder depth, and foundation type. Cases of cable stayed bridges are available in the literature [35] which could be used as learning examples in induction. Inductive paradigms based on concept acquisition [6] require tutoring and would not serve the requirements. Conceptional clustering CLUSTER/2 [36] and other similar algorithms can generate only a hierarchical organisation of objects classified by conjunctive statements. The learning process in the configuration generation should be able to represent many to many relationship between objects derived using operators expressing other logical implication in addition to conjunction. An induction algorithm which can create a network structure between attributes of the domain. This would involve creation and use of fuzzy definition of attribute values.

Decisions regarding trial shapes and sizes for members are made at the stage of preliminary sizing. Past experience plays a major role at this stage. Maher and Li [23] and Adeli and Yeh [24] have demonstrated machine learning in this domain using induction and perceptron, respectively. Neural network with hidden layers could learn from earlier design experience and thus enhance preliminary design capability.

Detailed design is the iterative process of checking the adequacy of trial sections to meet all the constraints of the design. This falls under the category of routine design. The detailed design has to be repeated for many member in the structural system such as tension and compression members of the truss bridge as well as their connections. Knowledge chunking algorithm helps in speeding up this process [33].

Time and cost overrun in large projects are frequently due to the difficulty in planning and managing such construction projects. Technical, social and environmental uncertainties influence the construction process. Construction planners and managers learn to tackle these activities under uncertainties, based on their past experience on similar projects. Cause effect relationship in these activities is not well documented. Neural network system can be trained using past cases to learn the implicit knowledge associated with the process. The trained neural network serves as the transfer function relating inputs and outputs of the construction planning process. The self organisation, generalisation, fault tolerance, and massively parallel processing properties of the neural network systems are useful in this activity.

The process of solving any major engineering problems is an open ended problem. Many agents cooperating opportunistically, and interacting in a non-deterministic and non-trivial way contribute to the solution in an incremental but non-monotonic fashion. Computing systems such as DICE [34], attempt to facilitate such a cooperative problem solving process in real time. IES based on opportunistic knowledge scheduling, models such a problem solving strategy. In IES without learning capabilities, the choice of one rule from among many competing rules or one knowledge source from among the competing knowledge sources is predetermined by the priorities set in advance, based on the experience of the developer. In IES with learning capabilities such priorities can be continually updated based on the past problem solving experience. The process and not the product of the past experience is used in learning. Induction paradigms provide algorithms for learning problem solving process.

IES also has a rich user interaction facility. Control is given to the user whenever a new input is required or a new knowledge source is to be scheduled in addition to pauses at pre-defined points depending upon the domain requirements. During such interruptions the user can review the solution and make modifications to any value already inferred or change the event to be pursued which may be different from that dictated by control knowledge. Such user inputs serve as a rich source for learning the problem solving process. An abstraction of the entire problem solving trace is stored in IES as a dependency network. The dependency network also serves as a source for learning the problem solving process. An induction learning algorithm could be used to achieve this learning. IES handles the non-monotonic problem solving process in engineering, using the dependency network and the consistency maintenance mechanism. Whenever a design failure or a constraint violation occurs, the knowledge based backtracking mechanism takes over and restarts from an earlier state after appropriate modifications to the solution state and dependency network. The knowledge for the backtracking may be available in the knowledge base, if the episode has been already envisaged. Otherwise the advice is obtained from the user. Such backtracking knowledge with accompanying explanation is to be used to minimize or eliminate unnecessary problem solving cycles in the subsequent sessions in the IES, using an explanation based learning algorithm.

5. SUMMARY AND CONCLUSIONS

It is seen that no single strategy could effectively serve the machine learning requirements of large applications. IES requires implementation of different learning strategies and the engineering application developer can make a choice depending upon the domain requirements. The opportunistic knowledge scheduling and maintenance of dependency network in the IES system based on extended blackboard architecture are features which aid the implementation of the learning strategies. The learning strategies as discussed are being currently implemented and tested in the IES system. For brevity, implementation details are not presented in this paper.

6. ACKNOWLEDGEMENT

The work reported in this paper is supported by the sponsored research project on "Expert System Based Computer Aided Engineering of Steel Structures", supported by Department of Science & Technology.

REFERENCES

1. SIMON H.A., Why Should Machines Learn, in Michalski, R.S. Carbonell J.G. and Mitchell T.M. (Eds), Machine Learning, Tioga, Palo Alto CA, 1983, pp. 25-37.
2. CARBONELL J.G., Introduction: Paradigms for Machine Learning, Artificial Intelligence, Vol. 40, No. 1-3, 1989, pp. 1-9.
3. DIETTERICH T.G. and MICHALSKI R.S., Inductive Learning of Structural Descriptions, Artificial Intelligence, Vol. 16, No. 3, 1981, pp. 257-294.
4. HUNT E.B., Concept Learning:An Information Processing Problem, Wiley, New York, 1962.
5. QUINLAN J.R., Induction of decision trees, Machine Learning, Vol. 1, 1986, pp. 81-106.
6. GENNARI J.H., LANGLEY P. and FISCHER D., Model of Incremental Concept Formation, Artificial Intelligence, Vol. 40, No .1-3, 1989, pp. 11-61.
7. HALL R.J., Computational approaches to analogical reasoning:A comparative analysis, Artificial Intelligence, Vol. 39, No. 1, 1989, pp. 39-120.
8. WINSTON P.H., Learning and Reasoning by Analogy, Commn. ACM, Vol. 23, No. 12, 1980, pp. 689-703.
9. MOSTOW J., Design by Derivational Analogy:Issues in the Automated Replay of Design Plans, Artificial Intelligence, Vol. 40, No. 13, 1989, pp. 119-184.
10. KOLODNER J., SIMPSON R. and SYCARA K., A process model of case- based reasoning on problem solving, in Joshi A. (Ed)., Proc. of IJCAI-85, Los Angeles, CA, 1985, pp. 284-290.
11. SCHANK R.C. and LEAKE D.B., Creativity and Learning in a case- based Explainer, Artificial Intelligence, Vol. 40, No. 1-3, 1989, pp. 353-385.
12. DEJONG J. and MOONEY A., Explanation Based Learning - An alternative view, Machine Learning, Vol. 1, 1986, pp. 145-176.
13. MINTON S., CARBONELL J.G., KNOBLOCK C.A., KUOKKA D.R., ETZIONI O., AND GIL Y., Explanation Based Learning:A Problem solving perspective, Artificial Intelligence, Vol. 40, No. 1-3, 1989, pp. 63-118.
14. CARBONELL J.G., Learning by Analogy:Formulating and generalizinG plans from past experience in Michalski R.S., Carbonell J.G. and Mitchell T.M. (Eds.), Machine Learning , Tioga, Palo Alto, CA, 1983, pp. 137-161.
15. BOOKER L.B., GOLDBERG D.E. and HOLLAND J.H., Classifier systems and Genetic Algorithms, Artificial Intelligence, Vol. 40, No. 1-3, 1989, pp. 235-282.
16. HOLLAND J.H., Genetic algorithms and the optimal allocation of trials, SIAM Jl. Comput., Vol. 2, 1973, pp. 88-105.
17. HINTON G.E., Connectionist Learning Procedures. Artificial Intelligence Vol. 40, No. 1-3, 1989, pp. 185-234.
18. AMARI S.I., A theory of adaptive pattern classifiers, IEEE Trans. Electron. comput., Vol. 16, 1967, pp. 299-307.
19. LENAT D.B., The role of heuristics in learning by discovery:three case studies in Michalski R.S., Carbonell J.G. and Mitchell T.M. (Eds.), Machine Learning , Tioga, Palo Alto, CA, 1983, pp. 243-306.

20. MITCHELL T.M., UTGOFF P.E. and BANERJI R., Learning by experimentation:Acquiring and refining problem solving heuristics, in Michalski R.S., Carbonell J.G. and Mitchell T.M. (Eds.), Machine Learning, Tioga, Palo Alto, CA, 1983, pp. 163-190.
21. MOSTOW D.J., Machine transformation of advice into a heuristic search procedure, in Michalski R.S., Carbonell J.G. and Mitchell T.M. (Eds.), Machine Learning, Tioga, Palo Alto, CA, 1983, pp. 368-403.
22. ROONEY M.F. and SMITH S.E., Artificial intelligence in engineering design, Computers & Structures, Vol. 16, No. 1-4, 1983, pp. 279-288.
23. MAHER M.L. and LI H., Automatically Learning preliminary design knowledge from design examples, Microcomputers in civil engg., Vol. 7, No. 1, 1992, pp. 73-80.
24. NAVINCHANDRA D., SRIRAM D. and KEDAL-CABELLI S.T., Analogy based engineering problem solving:An overview, in KBES in engineering:Tools and techniques, 1987, pp. 273-285.
25. ZHAO F. and MAHER M.L., Using analogical reasoning to design buildings, Engg. with computers, Vol. 2, 1988, pp. 107-119.
26. MURALIDHARAN T.L., ARAVIND H.B., SURYAKUMAR G.V., AND RAMAN N.V., Expert tower analysis and design systems:Part II. Search strategies and learning, Jl. of comput. in civil engg., Vol. 5, No. 2, 1991, pp. 193-210.
27. ARCISZEWSKI T. and ZIARKO W., Inductive learning in civil engineering:Rough sets approach, Microcomputers in civil Engg, Vol. 5, No. 1, 1990, pp. 19-28.
28. YEH, Y.C., Kuo Y.H. and Hsu D.S., Building KBES for diagnosing PC Pile with inductive learning, Jl. of comput. in civil engg., Vol. 6, No. 2, 1992, pp. 200-219.
29. ADELI H. and YEH C., Perceptron learning in engineering design, Microcomputers in civil engg., Vol. 4, No. 4, 1989, pp. 247-256.
30. KAMARTHI S.V., SANVIDO V.E. and KUMARA S.B.T., Neuroform - neural network system for vertical formwork selection, Jl. of comput. in civil Engg., Vol. 6, No. 2, 1992, pp. 178-199.
31. MOSELHI O., HEGAZY T. and FAZIO P., Neural networks as tools in construction, Jl. of constrn. engg. & mgt., Vol. 117, No. 4, 1991, pp. 606-625.
32. SAKTHIVEL T.S. and KALYANARAMAN V., A KBES for integrated engineering (Accepted for publication in Engineering with Computers)
33. REICH, Y. and FENVES S.J., Floor-system design in SOAR:A case study of learning to learn, Technical report EDRC-12-26-88, Engg. Design Research Center, Carnegie Mellon University, Pittsburgh, PA, 1988.
34. AHMED S.M., SRIRAM D., AND LOGCHER R., Transaction Management issues in collaboration engineering, Jl. of comput. in civil engg., Vol. 6, No. 1, 1992, pp. 85-105.
35. PODOLNY W. and SCALZI J.B., Construction and design of cable- stayed bridges, Wiley Interscience, U.S.A, 1986.
36. MICHALSKI R.S. and STEPP R.E., Learning from observation:Conceptual clustering, in Michalski R.S., Carbonell J.G. and Mitchell T.M. (Eds.), Machine Learning , Tioga, Palo Alto, CA, 1983, pp. 331-363.

Using Case-Based Reasoning for the Synthesis of Structural Systems
Raisonnement rapporté au cas spécifique dans la synthèse des systèmes structuraux
Fallgestütztes Schliessen für die Synthese von Gebäudesystemen

Dong Mei ZHANG
Univ. of Sydney
Sydney, Australia

Mary Lou MAHER
Univ. of Sydney
Sydney, Australia

SUMMARY
Design synthesis is defined to be the generation of alternative design solutions. Domain knowledge provides design principles and performance theoretical guidance, design episodes serve as resources in the design synthesis process because they record experience and reasoning steps. The main aim of our research is to explore a design process model incorporating both episode-based design situations and generalized domain knowledge for design synthesis. This paper presents an approach to combining case-based reasoning and decomposition to derive a new design solution by the transformation of previous design situations. The issues of the representation of realistic structural designs in a case base and the transformation of previous design situations are addressed in the paper.

RÉSUMÉ
On entend par synthèse d'études la génération de solutions de rechange dans l'établissement des projets. Un modèle de processus d'études, en cours d'évaluation, combine un domaine de connaissances, englobant des règles d'études et des caractéristiques de performance, avec des situations d'études épisodiques, dans lesquelles sont mémorisées les expériences et les conclusions de raisonnements déductifs. Les auteurs présentent une méthode qui, à partir d'études mises en archives, permet d'effectuer des analyses et des déductions et, de la sorte, conduit à de nouvelles solutions par transformation de situations d'études précédentes. L'article développe d'une part les questions de la représentation d'études spécifiques de bâtiments et d'autre part, le processus de transformation des situations d'études précédentes.

ZUSAMMENFASSUNG
Unter Entwurfssynthese wird die Generierung alternativer Lösungen der Entwurfsaufgabe verstanden. Es wird an einem Entwurfsprozessmodell geforscht, das ein Wissensgebiet aus Entwurfsregeln und Leistungsmerkmalen mit episodischen Entwurfssituationen kombiniert, in denen Erfahrungen und Schlussfolgerungen gespeichert sind. Der Beitrag stellt eine Methode vor, wonach aufgrund archivierter Entwürfe Schlüsse und Analysen ermöglicht, und durch Transformation früherer Entwurfssituationen neue Lösungen gefunden werden. Dabei wird auf Fragen der Darstellung realistischer Gebäudeentwürfe in einer Fallsammlung und des Transformationsprozesses näher eingegangen.

1 INTRODUCTION

Design is a process in which the experience and knowledge of designers and the design specifications are combined, during which a design description is generated to satisfy the design intentions. In the synthesis of design solutions, alternative configurations are generated and evaluated. During design synthesis, domain knowledge provides design principles and performance theoretical guidance, design episodes serve as resources because they record experience and reasoning steps.

There is no standard method of synthesis suitable for all design problems. The case-based reasoning (CBR) paradigm provides a model for applying prior experience to new problems. It involves retrieving relevant previous cases, adapting the solution from a previous case to solve new problems, and storing the current episode as a new case to be used in the future. CBR as a process model of design synthesis is appealing intuitively because much of design knowledge comes through experience of multiple, individual design situations. For many domains where design knowledge is difficult to acquire and may not be objectively applicable, the case-based paradigm presents a model for the acquisition, organization and reuse of specific design knowledge. Using CBR as a design process model raises the following issues: the identification of what is in a design episode in order to reason about its applicability in a different design context, and the transformation of previous design situations from an original context to a new context.

For design, what is stored in a case reflects the characteristics of design knowledge, as design case retrieval and transformation are based not just on surface features such as the description of design solution, but also on the causal relations between function, behavior, and performance etc. This increases the complexity of the representation and organization of design cases. Whether to include the relational knowledge and governing constraints for a design case within the case or to represent this knowledge outside case memory is still an open research question.

Transformation of a case plays a crucial problem solving role in the CBR paradigm. Transformation includes identifying the difference between the retrieved cases and the new problem and modifying the solution stored in the retrieved case to take those differences into account. The issues raised by transformation are: the representation of domain knowledge about transformation; the maintenance of consistent modification; and the verification of a feasible solution. Previous designs can not be reused without substantial changes. A previous design is either proprietary or customized for a specific context. Proprietary designs (such as Xerox copier) can not be used again without violating laws. Customized designs (such as buildings) can not be used again because the exact context will rarely arise again.

As a result of many efforts toward using CBR for design problem solving, it is found that certain generalized or compiled domain knowledge are essential to address some of the issues in the case-based design model. A hybrid model, therefore, becomes a common approach in many implementations of case-based designs.

In recent years many CBR computer models have used the idea of hybrid systems, and have been developed in engineering design domains. In Wang and Howard's [1988] integrated system for structural engineering design, case-based and rule-based reasonings are combined. A past design can be applied to a similar design problem by replaying its previous design plans. A conventional rule-based module applies design codes and analysis procedures to create the design solutions when case-based design actions are not available. In Faltings et al's [1991] case-based architectural design, the representation of a case involves specific design knowledge and domain dependent knowledge including transformation rules. Case transformation deals with dimensional and topological discrepancies in which a specific design is treated as a starting point of a new design.

As another typical range of case-based designs, some integrated design systems combine model-based reasoning with case-based reasoning. KRITIK [Goel and Chandrasekaran 1989] is an example of this method for the design of small mechanical assemblies. Causal understanding of structure, function and behavior about a device resides in a design model as a functional representation schema, whereas individual mechanical devices are represented an instance of relevant design models. In [Sycara et al 1988], a design case is a graph-based behavior model about a particular device. Case-based reasoning in this approach is viewed as a methodology for selecting and applying various design models rather than specific episodes.

2 CADSYN : COMBINING CBR & DECOMPOSITION

CADSYN [Maher and Zhang 1991, 1993] provides a process model for design in which case-based reasoning is combined with a generalized decomposition approach, where the CBR and decomposition approaches complement each other to provide a flexible and comprehensive model of design. In this paper, we will focus on the approach where CBR is adapted to provide a model for selecting and transforming previous design situations to fit a new context using decomposition and constraints knowledge. The process model integrates three distinct types of knowledge: specific design situations, generalized decomposition of a design domain into systems and components, and design constraints. The components of knowledge and main processes for the CBR approach are illustrated in Fig.1.

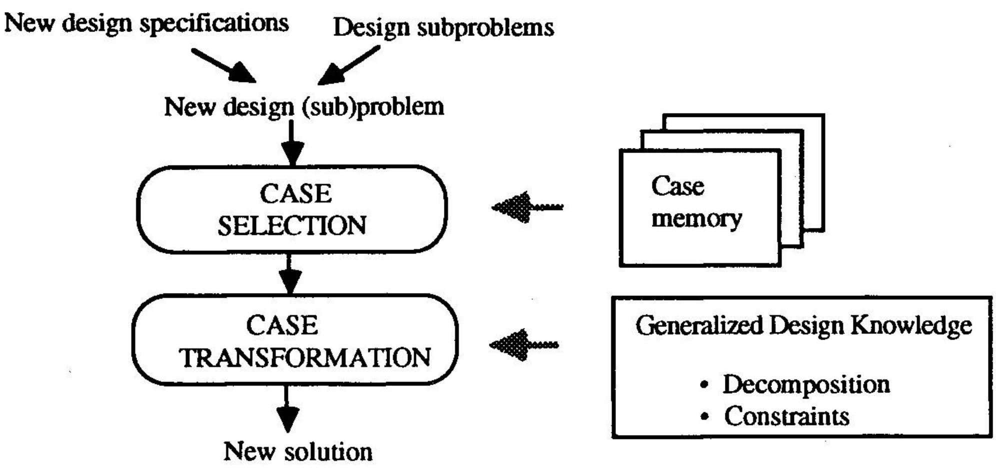


Fig.1 The overall architecture of CADSYN

The problem solving process in CADSYN is primarily divided into case selection and case transformation. Given a new design problem or subproblems, a case or subcase which was designed for a similar context is selected from case memory. The selected case or subcase is then transformed to the new context through modifications which resolve the conflicts caused by difference between the original and the new contexts. A solution, thus, is derived based on (1) the most relevant previous design situation being selected, i.e a close match is found; and (2) transforming the potential solution to fit the new design situation using a domain specific constraint satisfaction approach.

Case selection The selection of the most similar design case consists of two steps, retrieval and selection, as illustrated in Fig.2. The retriever traverses the case memory according to the new problem definition, and identifies the similarities between cases and the new problem. The selector then compares the similar design cases to choose the most relevant one. Case selection involves assessing not only how close the past cases are to the new problem, but also the relative importance of the relevant similarities and differences. To model this selection process, a weighted count of matching features is applied.

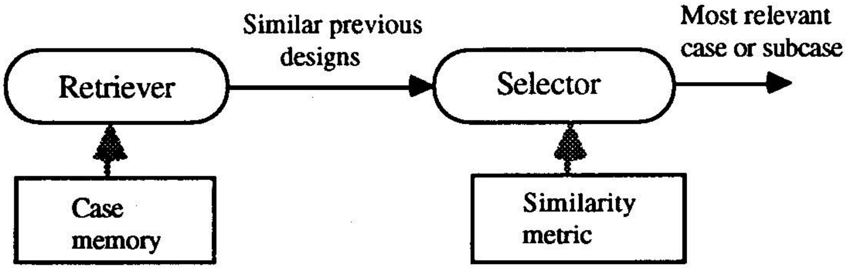


Fig.2 Design case selection

Case transformation Transformation in CADSYN forms the essence of design synthesis, using a holistic approach to design by starting with a solution and adapting it to fit a new context. Transformation in our model assumes that case selection provides a description of a specific design solution that is close to the acceptable final solution, and transforms those aspects of the design solution that are inconsistent. Transformation can be divided into three logical phases: adapt, verify, and repair, as shown in Fig.3. First, a potential solution to a new problem is proposed as the solution from the selected cases. This potential solution is adapted to change the difference in specifications and the design description, which introduces some inconsistencies between the design specifications and the design description. This step is then followed by verification, the process of evaluating the new solution by checking design constraints, and modification, the process of fixing an inconsistent design to satisfy the violated constraints. Once the solution has been revised, it is verified again. This process proceeds until all constraints are satisfied and a feasible solution is found.

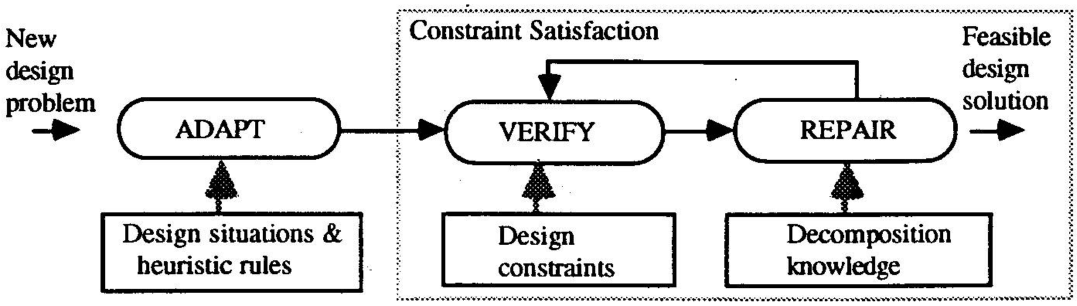


Fig.3 Design case transformation

To explore the case-based reasoning approach for design synthesis, CADSYN addresses two major issues: the representation of design cases and the transformation of a design case. A design case is a description of the design problem and design solution, in which there are no causal relations to support a design decision. The transformation processuses generalized decomposition knowledge and design constraints as the causal relations that justify and verify design decisions.

3 REPRESENTATION OF DESIGN CASES

The formulation of design cases is a prerequisite for a CBR approach to design. In our project, design cases are collected with the cooperation of Acer Wargon Chapman Associates, an engineering consulting firm in Sydney. We were given access to the drawings for the structural design of their building projects. For each building project, there is a set of drawings produced for documentation purposes, primarily for the purpose of communicating the information needed to construct the building. This means, for example, it can be seen on the drawings how much reinforcement each beam has, however, no or very little design information such as lateral load resistance or system design reside on the drawings. To acquire design cases, we augmented the information found on the drawings with interviews with the engineers involved the project. We chose not to put drawings in case memory but to capture the essential design information.

In CADSYN, a design case is represented in a case hierarchy in the form of attribute-value pairs. A design case consists of a supercase part and multiple levels of subcases. This case representation provides the process model with a means to use subcases independently of the entire case. The supercase of a case provides the overall design episode context and general description. Each subcase describes the local context and the solution of a design subsystem. Subcases are indexed individually along with links that can be used to construct the whole case.

As both specific design cases and generalized decomposition knowledge are incorporated to derive a new design solution, a correspondence between them is established as follows: the design description of a particular design case is associated with a set of subsystems from the decomposition knowledge, where each subcase matches a generalized subsystem. This ensures that subsystems and their attributes can be recognized during case transformation for constraint checking and repair.

A design case in CADSYN is the description of a design context comprised of design specifications and a design solution, in which there are no causal relations to support a design decision. In structural design, the content of a design case is constructed in three layers: problem specifications as a global context; a grid representation for each function of building as the geometric context; and structural systems as a design solution for each grid level. The problem specifications of a design case include general architectural specifications and loading information, such as the number of stories, the intended use of the building, etc. The grid representation contains bay numbers and sizes in the two principle orthogonal directions, and other functional and geometric information. A design case of a particular building is illustrated in Fig.4. This office provides four functional spaces: parking, retail, office, and service-core. GEN-CASE shows a overall problem description of the building. The local design context for the office space is shown in the GRID3, and the structural design description associated with GRID3 is illustrated in terms of lateral-systems, gravity-system, transfer etc. The attributes in the structural design description are categorized as requirements (req) and design decisions (des).

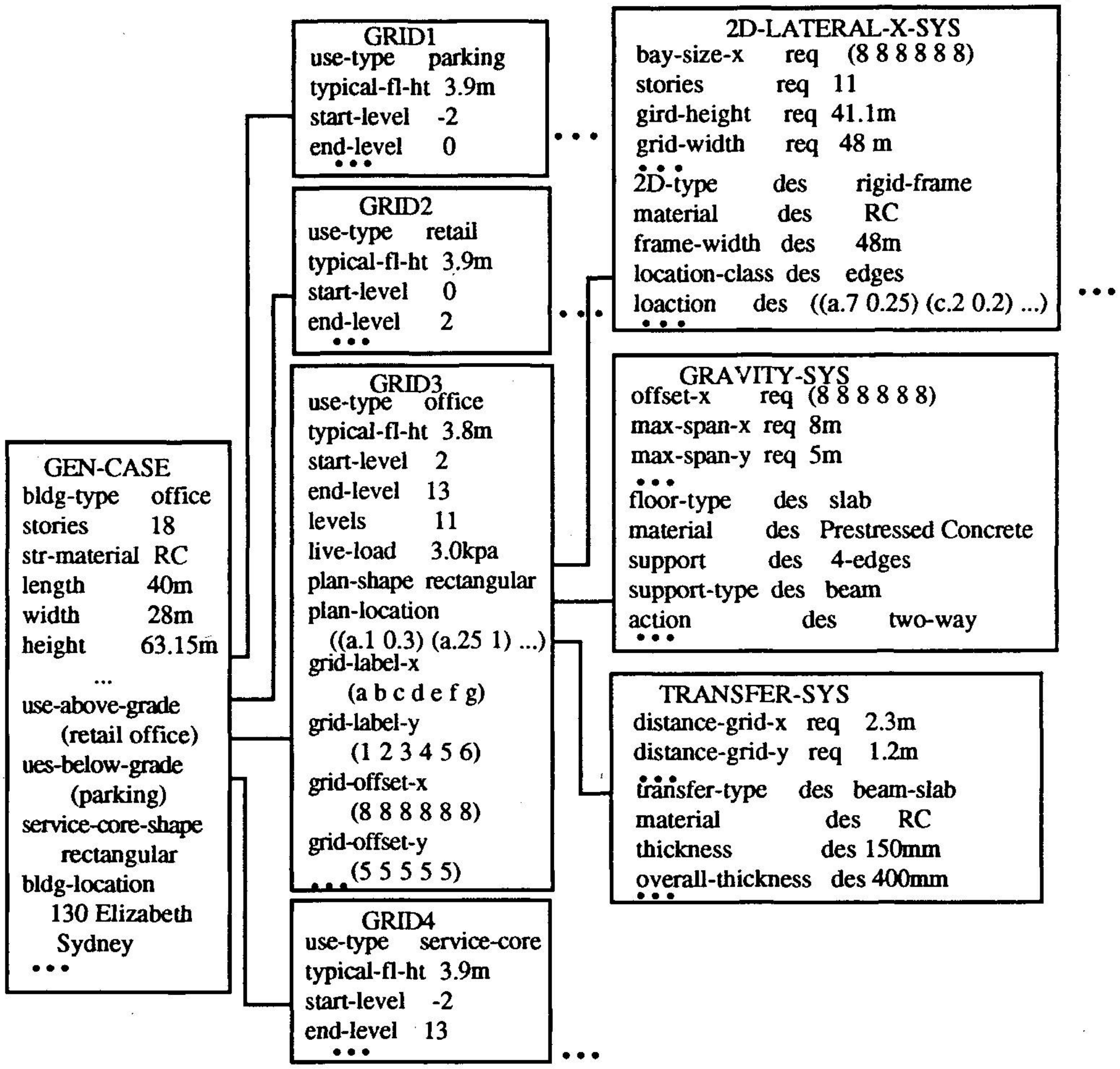


Fig.4 A partial description of a design case for an office building

4 TRANSFORMATION PROCESS IN CADSYN

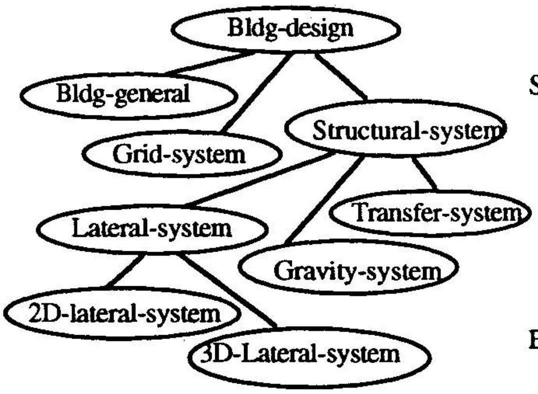
The transformation process of a design case in CADSYN is an adaptation-verification-modification iterative process using constraint satisfaction and decomposition knowledge. In the transformation, the previous design examples act as a starting point for a new design generation and suggests a potential solution to the new problem. An initial solution is firstly constructed by structurally transforming the solution of the most relevant design case. The transformer then adopts a constraint satisfaction approach to check the feasibility of the solution and repair invalid design decisions in the adapted solution.

In this section, the generalized design knowledge used for supporting the transformation process is represented and the strategy for the constraint satisfaction approach in the transformation is described.

4.1 Representation of Generalized Knowledge

The transformation process in CADSYN uses two types of generalized design knowledge: decomposition knowledge and design constraints based on the representation of knowledge in EDESYN [Maher 1989].

Generalized decomposition knowledge provides the transformation process with the search space of possible alternatives for design attributes. The decomposition knowledge describes how a design system is to be decomposed into attributes. Each attribute can be defined in one of three ways: further decomposition through subsystem design; selecting from an enumerated set of discrete values; or the evaluation of a procedural function. A generalized system for the gravity system is as shown in the right hand part of Fig.5, where "floor-type", "material" etc. are selected from sets of discrete values, and "typical-span" with computed values by the procedure "get-span".




```
SYSTEM Gravity-system
  ...
  typical-span  CALCULATION (get-span)
  floor-type    SELECTION (flat-plate flat-slab ...)
  material      SELECTION (steel rein-concrete ...)
  support       SELECTION (cols-only 2-edges ...)
  action        SELECTION (1-way 2-way)
  ...
END-SYSTEM
```

Fig.5 A hierarchy of structural subsystems and the description of the gravity system

Fig.5 illustrates a hierarchy of subsystems for a decomposition of the structural design of buildings. The nodes in the hierarchy represent decomposition systems. At the top level, the bldg-design system is broken into three subsystems, namely, Bldg-general, Grid-system and Structural-system. The structural-system leads the synthesis process to a further decomposition of the structural design solution.

Design constraints are used to identify legal decisions and test a potential solution to a new problem. Each constraint is a declarative statement which eliminates a design alternative. For the purpose of design synthesis, it is appropriate to represent constraints as infeasible combinations of attribute-value pairs or relations, since the role of constraints in the early stages of design is to prune the potentially large number of design alternatives.

Examples of constraints for the structural design of buildings are given in Fig.6. CONSTRAINT-1 specifies that the flat-plate and flat-slab are not used as floor types in heavy load buildings such as an office, parking or institution. CONSTRAINT-2 indicates whether the flat-plate and flat-slab work as one-way or two-way given their typical-span.

CONSTRAINT-1	CONSTRAINT-2	
(bldg-type) in (office parking institution)	(floor-type) in (flat-plate flat-slab)	in : one-of
(floor-type) in (flat-plate flat-slab)	(action) = one-way	< : less than
	(typical-span) < 10m	= : equal to

Fig.6 Examples of elimination design constraints

4.2 Transformation by Constraint Satisfaction

The transformation of cases in CADSYN can be modeled as a constraint satisfaction problem (CSP), where decomposition knowledge defines the domain of possible values associated with each design attribute; and design constraints represent compatibility and selection knowledge. By treating transformation as a CSP, our method takes an initial inconsistent assignment for the design attributes and incrementally repairs constraint violations until a consistent assignment is achieved.

If the potential solution adapted from the previously selected design case violates design constraints, it will be input to the constraint satisfaction process as an inconsistent assignment for the design attributes. The transformation is characterized by searching for a consistent assignment for all design attributes subject to design constraints. A constraint satisfaction process, as illustrated in Fig.7, is applied to find a consistent solution.

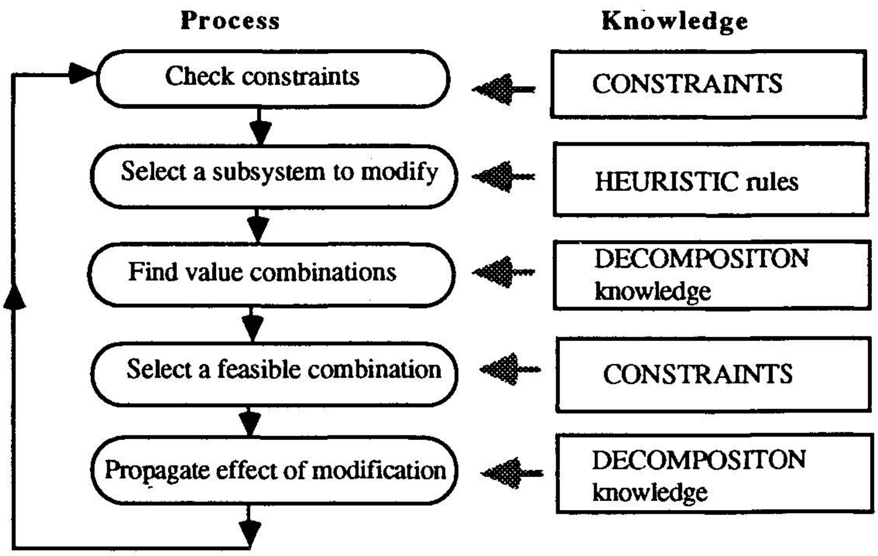


Fig.7 Constraint satisfaction in CADSYN

Check constraints. The potential solution provides a set of attribute assignments based on the adaptation of the retrieved case or subcase. This set of attribute assignments is compared to the constraints in the generalized knowledge base to identify violated constraints and the subsystems associated with the constraints. If no constraints are violated, a feasible solution has been found.

Select a subsystem. An appropriate subsystem is then identified to be focused on based on a set of heuristic rules. During repairing, the strategy is to fix lower level design decisions rather than change subsystems.

Find value combinations. A value for an attribute can be selected from a discrete set of values or it can be computed using a procedure. The possible value combinations are generated by assigning possible discrete attributes, then attributes with computed values are assigned new values by applying relevant procedures based on the value combinations of discrete attributes. The final result of the constraint satisfaction process is a set of possible value combinations in the selected subsystem.

Select a feasible combination. The determination of which value combination is used as a new subsystem description is based on the number of constraints which are satisfied. That is, the value combination which satisfies most constraints is regarded as a new design description for the subsystem.

Propagate effect of modification. Once a selected subsystem is modified, all relevant attributes in other systems associated with this subsystem are updated by recomputing corresponding procedures. The process iterates from this point by identifying new constraint violations until all constraints are satisfied.

5 CONCLUSIONS

The issues raised in this paper are the result of developing the CADSYN process model, and applying the model to the design of structural systems for tall buildings by collecting cases from an engineering consulting company. Direct collection of real world designs leads to difficulties due to the complexity of building design process and the formulation of the case data from design drawings and interviews with designers. One of major issues is that design information is not on the structural drawings. The drawings are used for documentation of the design and contain overwhelming detail on the data needed to construct the object and very little about how the system works. Another issue is capturing the intent of the design so that its adaptation can be consistent following the original context. In general, we augmented the information found on the drawings with interviews with the engineers involved in the project design. The requirements for the design are regarded as a substitution of the designers' intent. A hierarchical representation is used to represent a building design, each subsystem has an associated functional label, set of requirements, and design decisions.

The transformation of a design case in CADSYN is addressed as a design synthesis process using both specific case knowledge and generalized decomposition knowledge. The knowledge about the behavior of structural systems is represented once as generalized decomposition knowledge, rather than repeating it for each case. The knowledge about detecting the design failures is represented as constraints to verify the adapted design and ensure a consistent modification. A new design is generated by adapting cases and performing a constraint satisfaction process in which decomposition knowledge provides a space of possible alternatives for a design attribute, and elimination constraints represent compatibility and selection knowledge. In the development of a constraint satisfaction approach to transformation, how to accumulate the appropriate generalized knowledge is a major issue as there is very little causal knowledge available at the preliminary design stage. Constraints, for example, are used to determine whether an adapted design is feasible, but as knowledge used for preliminary design, such constraints are based on heuristics rather than on an analysis of the behavior of the solution.

Acknowledgements
We would like to thank Acer Wargon Chapman Parters for providing access to their structural design reports and drawings and for participating in discussion. Special thanks to Rita Villamayor for the infinite number of hours she spent looking at the drawings to find some design data. Thanks also to M.Balachandran for his contribution to the development of these ideas.

REFERENCE

1. Faltings, B., and Hua, K. et al (1991). Case-Based Representation of Architectural Design Knowledge. In Proceedings of the DARPA Workshops on Case-Based Reasoning. Washington, D.C. 307-316.
2. Goel and Chandrasekaran, (1989). Use of Device Models in Adaptation of Design Cases. In Proceedings of the DARPA workshop on Case-Based Reasoning. Pensacola Beach, Florida. 100-109.
3. Maher, M. L. (1989). Synthesis and Evaluation of Preliminary Designs. In J. S. Gero (editor), Artificial Intelligence in Design. Springer-Verlag. 3-14.
4. Maher, M.L. and Zhang, D. M. (1991). CADSYN: Using Case and Decomposition Knowledge for Design Synthesis. In J. S. Gero (editor), Artificial Intelligence in Design' 91, Butterworth-Heinemann, UK. 137-150.
5. Maher, M.L. and Zhang, D.M. (1993). CADSYN: A Case-Based Design Process Model. AI EDAM (to appear).
6. Sycara, K.P. and Navinchandra, D. (1989). Integrated Case-based Reasoning and Qualitative Reasoning in Engineering Design. In J. S. Gero (editor), Artificial Intelligence in Design, Springer-Verlag. 231-250.
7. Wang, J. and Howard, H.C. (1991). A Design-Dependent Approach to Integrated Structural Design. In J. S. Gero (editor),Artificial Intelligence in Design'91. Buttorworth-Heinemann, UK. 151-170