SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION	UNCLASSIFIED
1b. RESTRICTIVE MARKINGS	NONE
2a. SECURITY CLASSIFICATION AUTHORITY	
3. DISTRIBUTION/AVAILABILITY OF REPORT	APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
5. MONITORING ORGANIZATION REPORT NUMBER(S)	AFIT/CI/CIA- 89-100
6a. NAME OF PERFORMING ORGANIZATION	AFIT STUDENT AT CLEMSON UNIV
6b. OFFICE SYMBOL *(If applicable)*	
7a. NAME OF MONITORING ORGANIZATION	AFIT/CIA
6c. ADDRESS *(City, State, and ZIP Code)*	
7b. ADDRESS *(City, State, and ZIP Code)*	Wright-Patterson AFB OH 45433-6583
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	
8b. OFFICE SYMBOL *(If applicable)*	
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS *(City, State, and ZIP Code)*	
10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.	
11. TITLE *(Include Security Classification)*	(UNCLASSIFIED) DEVELOPMENT OF A FUZZY LOGIC EXPERT SYSTEM FOR PILE SELECTION
12. PERSONAL AUTHOR(S)	MICHAEL L. ULSHAFER
13a. TYPE OF REPORT	THESIS/~~DISSERTATION~~
13b. TIME COVERED	FROM ____ TO ____
14. DATE OF REPORT *(Year, Month, Day)*	1989
15. PAGE COUNT	136
16. SUPPLEMENTARY NOTATION	APPROVED FOR PUBLIC RELEASE IAW AFR 190-1 ERNEST A. HAYGOOD, 1st Lt, USAF Executive Officer, Civilian Institution Programs
17. COSATI CODES: FIELD / GROUP / SUB-GROUP	
18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*	
19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*	

90 02 14 044

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS
21. ABSTRACT SECURITY CLASSIFICATION	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL	ERNEST A. HAYGOOD, 1st Lt, USAF
22b. TELEPHONE *(Include Area Code)*	(513) 255-2259
22c. OFFICE SYMBOL	AFIT/CI

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

AFIT/CI "OVERPRINT"

May 5, 1989

To the Graduate School:

This thesis entitled "Development of a Fuzzy Logic Expert System for Pile Selection" and written by Michael L. Ulshafer is presented to the Graduate School of Clemson University. I recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Civil Engineering.

C. H. Juang
Thesis Advisor

We have reviewed this thesis
and recommend its acceptance:

R. E. Elling

Wm Baron

Accepted for the Graduate School:

Farrell B. Brown

ABSTRACT

This thesis documents the development of a prototype expert system for pile selection for use on microcomputers. It concerns the initial selection of a pile foundation taking into account the parameters such as soil condition, pile length, loading scenario, material availability, contractor experience, and noise or vibration constraints. The prototype expert system called Pile Selection, version 1 (PS1) was developed using an expert system shell FLOPS. FLOPS is a shell based on the AI language OPS5 with many unique features. The system PS1 utilizes all of these unique features. Among the features used are approximate reasoning with fuzzy set theory, the blackboard architecture, and the emulated parallel processing of fuzzy production rules. A comprehensive review of the parameters used in selecting a pile was made, and the effects of the uncertainties associated with the vagueness of these parameters was examined in detail. Fuzzy set theory was utilized to deal with such uncertainties and provides the basis for developing a method for determining the best possible choice of piles for a given situation. Details of the development of PS1, including documenting and collating pile information for use in the expert knowledge data bases, are discussed. (SCW)

A

DEVELOPMENT OF A FUZZY LOGIC EXPERT SYSTEM FOR PILE SELECTION

A Thesis

presented to

the Graduate School of

Clemson University

In Partial Fulfillment

of the Requirement for the Degree

Master of Science

Civil Engineering

---

by

Michael L. Ulshafer

May 1989

ACKNOWLEDGEMENTS

I'd like to express gratitude to my advisor, Dr. C. H. Juang, whose direction, suggestions, time and effort, made this work possible, particularly in the developement of the expert system PS1.

I'd like to sincerely thank Dr. W. Baron and Dr. R. Elling for reviewing this manuscript and offering suggestions for its improvement.

Finally, I want to dedicate this work to my son, Kyle, who was born during the research, and whom I love very much.

TABLE OF CONTENTS

Table of Contents (Continued)

LIST OF FIGURES

CHAPTER I

INTRODUCTION

As stated by Professor R. B. Peck (1), "driving piles for a foundation is a crude and brutal process". The interactions among the piles and the surrounding soil are complex. Placement of piles generally alters the character of the soil and intense stresses are developed locally near the piles. The nonhomogenity of soils, along with the effects of adjacent piles and pile shape, add further difficulties to the understanding of the soil-pile interaction. Despite the extensive pile test data, analysis, and irrefutable value of soil mechanics, prediction of pile bearing capacity involves considerable guesswork because it represents the contributions of so many unevaluated factors.

Piling is a form of construction of great antiquity, and an almost instinctive trust in piles for overcoming difficulties runs throughout foundation work. This attitude still exists today, fostered no doubt by the lack of collated knowledge of how piles really behave. It has often led to piles being installed where another type of foundation, particulary a shallow foundation, might have been preferable (2).

Although piling is still largely an art, and there are circumstances where dependence must be placed on experience or even on rule-of-thumb, the engineer endeavors as far as possible, to apply the methods of mechanics to pile foundation design. But currently, more than any one factor, experience might play the biggest role in selecting the pile that's best for the job. The main goal of this thesis is to provide reasonable initial weighted pile foundation choices, by developing a prototype expert system, Pile Selection One (PS1). This program is considered a prototype because the expert knowledge data base was developed largely through a comprehensive literature search and partly based on the author's own knowledge and judgement. No expert in pile foundation design or construction was actually consulted during the development of the current version of PS1. Once PS1 has recommended initial pile selections, they can then be used in the pile design process. The user need only have a general knowledge of the loading and site conditions.

Many structures are supported on pile foundations. Engineers face the challenge of designing a safe and yet economically feasible structure despite the numerous uncertainties involved in areas such as: loading conditions, soil conditions, material properties, design approaches and methods of construction, nuisance effects, and space and time constraints. Choosing an initial pile for a pile design is guess work based on experience. The design of pile foundations should ensure adequate safety margin against potential failure. The factor of safety for pile foundations is generally larger than those of shallow foundations because of the greater uncertainty of soil-pile interaction. Typically, larger factors of safety results in higher costs. PS1 will provide several weighted pile types that best fit the site scenario. The top pile choices should be the optimum pile type and thus provide reasonable factors of safety at minimum costs. Using these suggested pile types, further detailed analyses can be done using approved equations or computer programs for such things as bearing capacity, settlement analysis, cost comparisons, etc.

There exists a vast quantity of useful information on the subject of pile foundations which is scattered throughout literature. The purpose of this thesis is two fold: (1) compile useful information concerning pile foundations and their design and incorporate this information into a reliable expert knowledge data base for use by PS1, and (2) develop and discuss the prototype expert system, PS1. Therefore, this thesis is divided into two main sections: collating information concerning pile foundations and the evaluating the parameters used to create the prototype expert system PS1.

CHAPTER II

PILE FOUNDATIONS

General Background

Until this century a "pile" was a straight log of timber about 12 inches in diameter and some 33 feet long that was driven into the soil by the blows of a hammer. Piles that projected above the ground formed the supports for bridges and jetties and when driven entirely below the surface they were used to carry the walls and columns of buildings. Today, reinforced concrete and steel have mostly taken the place of timber. Although piles of these materials are driven like the timber log was driven, piling by another art has developed. Bored piles are made by making a tubular hole in the ground and pouring in concrete, which is allowed to harden (2).

A foundation is the interfacing element between the structure and the underlying soil or rock. Essentially a pile is an elongated or columnar body installed in the ground for the purpose of transmitting forces to the ground without excessive settlement. The loads transmitted from the structure to the underlying soil must not cause soil shear failure or damaging settlement. A pile foundation is used where adequate shallow foundations are impractical. When it is necessary to provide support and carry the load to an underlying stratum, such as through a layer of weak or compressible material, or through water, or in close proximity to existing structures, a pile foundation may be required. It also provides uplift resistance and/or lateral load capacity. Although capacity aspects may be emphasized in design, the foremost reason for using piles is to reduce deformation, normally settlement. Pile foundations are deep and usually costs more than a shallow foundation. Despite the cost, the use of piles often becomes necessary to ensure that the structure under consideration is safe. Following is a list of some of the conditions that require pile foundations (1, 3):

1. When the upper soil layer(s) is highly compressible and too weak to support the load transmitted by the structure, piles are used to transmit the load to underlying bedrock. When bedrock is not located at a reasonable depth below the ground surface, piles are used to gradually transmit the structural load to the soil by frictional resistance developed at the soil-pile interface. These are usually referred to as bearing piles as in Figures 1a and 1b.

2. Foundations of some structures, such as transmission towers, offshore platforms, and basement mats below the water table, are subjected to uplifting forces. Piles are sometimes used for these foundations to resist the uplifting force. These may be called tension or anchor piles as in Figure 1c.

3. When subjected to horizontal forces as in Figure 1d, pile foundations can resist bending, while still supporting the vertical load transmitted by the structure. This type of situation is generally encountered in the design and construction of earth-retaining structures and foundations of tall structures that are subjected to high wind and/or earthquake forces.

4. Bridge abutments and piers are occasionally constructed on pile foundations to avoid the possible loss of bearing capacity that a shallow foundation might suffer because of soil erosion of the foundation at the ground surface by scour of the stream bed during flood flow as in Figure 1e.

5. Piles may be considered as an alternative in expansive and collapsible soils such as loess. These soils may extend to a great depth below the ground surface. Due to an increase or decrease in moisture content, the swelling pressure of such soil can be considerably high and cause considerable damage to shallow foundations. In such cases, piles can be extended into stable soil layers beyond the zone of possible moisture change as in Figure 1f.

The selection of the most appropriate pile type for any given set of circumstances depends upon many variables, particularly the type of subsoil, the topography of the site, the location of the site in relation to

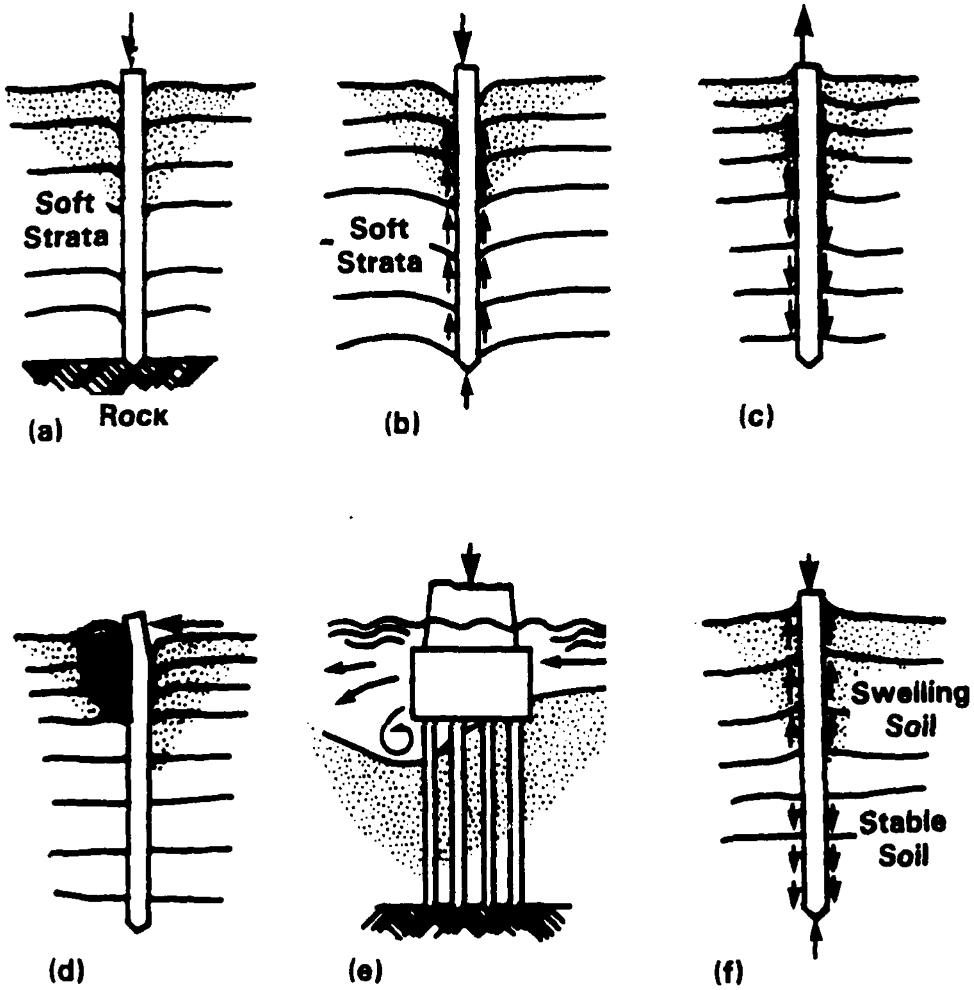


Figure 1. Various pile uses.

nuisance effects, transport and materials and the size and form of the proposed structure. There are, however, certain clear indications as to the suitability of a given pile type in particular conditions. Of all considerations, the type of subsoil existing on the site usually has the greatest influence on the type of pile to be selected. The type of structure to be erected has less influence than might be expected on the selection of pile type.

It should be noted that although this thesis concerns single pile selection for use in further analysis, rarely is a single pile used; rather two, three of more piles are used in a group. However, most of the accepted pile-capacity equations are for a single pile.

The Pile Design Process

To arrive at the optimum foundation solution, the foundation engineer must have thorough information and understanding of (1) foundation loads, (2) subsurface conditions and soil/rock properties, and (3) current practices in foundation design and construction.

Generally, the design process of a foundation system usually follows the steps as shown in Figure 2. Based upon site conditions, the designer must select an initial pile selection. This initial pile selection is then analyzed for design using accepted bearing equations or programs. Settlement analysis and cost comparisons are made, then the designer goes back and selects another initial pile choice and again analyzes this pile. This process can continue several times in an effort to find the pile which provides the factor of safety required at the minimum cost. The design analyses process can be taught and learned, but the initial pile selection is based upon experience, which may take many years to

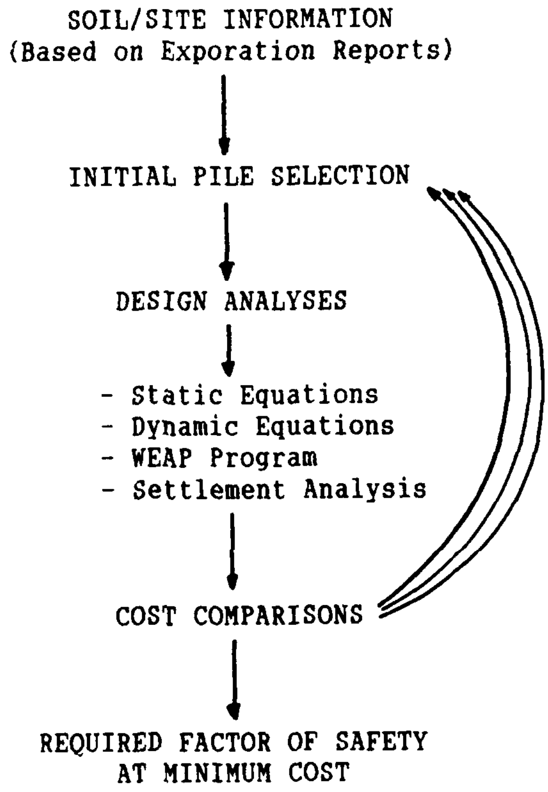


Figure 2. General design process.

acquire. This is where PS1 assists the designer in providing optimum pile selections and thus reducing the iterations needed to find the pile which gives the desired factor of safety at the minimum cost.

In most cases, the bearing capacity of a deep foundation unit is governed by geotechnical consideration, rather than by the structural strength of the unit. Therefore, a proper design of a structure foundation requires adequate knowledge of the subsurface conditions at the structure site. If the designer has comprehensive data, then an economical foundation system can be designed. The absence of a thorough geotechnical investigation or adequate data generally leads to a foundation system with a large factor of safety which is generally a more expensive foundation, or to an unsafe foundation, or to construction disputes and claims.

Subsurface Exploration

To design a foundation that will adequately support a structural loading, understanding the nature of the soils that will support the foundation is critical. Subsurface exploration is basically the process of determining the layers of natural soil deposits under the proposed site.

The geotechnical engineer or the foundation engineer has the responsibility for determining the type and capacity of foundations and should be involved in every phase, i.e. preliminary explorations, preliminary design, final design, construction, and post construction. This practice provides continuity of the design personnel through the construction stage (1).

Subsurface exploration comprises a few steps, such as gathering preliminary information, reconnaissance, and site investigation. Each phase adds to or supplements the information from the previous phase. Gathering preliminary information can be obtained from several sources such as United States Geological Survey maps, state government geological survey maps, United States Department of Agriculture's agronomy maps, agronomy maps published by the agriculture departments of various states, hydrological information published by the United States Corps of Engineers, which include the records of stream flow, high flood levels, tidal records, etc., highway department soils manuals published by several states, soil exploration reports for nearby completed construction projects, etc.

Reconnaissance involves a personal visit to the site to get a feeling of the general topography of the area, soil stratification from any deep cuts nearby, type of vegetation, and any potential problems such as abandoned dumps, drainage ditches, underground streams (with the help of the reports above), underground obstructions such as tanks or tunnels, creep of local slopes, underground and overhead services, both obsolete, current, and proposed, or cracks in nearby existing buildings. All of this information is usually presented in a preliminary report.

Site investigation consists of planning, making test boreholes, and collecting soil samples at various levels for subsequent observation and laboratory tests. Sowers and Sowers (4) recommend that the depth (D), in feet, of borings for light steel or narrow concrete structures equals

$$10(S)^{0.7}$$

and for heavy steel or wide concrete buildings equals

$$20(S)^{0.7},$$

where

S = number of stories.

They also state that when deep excavations are anticipated, the depth of boring should be five to fifteen feet below the depth of excavation and deeper if soft clay of loose sand and silt are encountered. Others have stated that the soil should be examined to a distance of between one and one-and-a-half times the width of the structure below the pile points, unless there is definite evidence from other sources that no compressible materials are present. In situ tests are needed to provide soil parameters for the design of structure foundations especially where standard drilling and sampling methods can't be used to obtain high quality undisturbed samples for lab analysis. The most common in situ test is the Standard Penetration Test (SPT) which is normally performed during the subsurface investigations. The SPT test provides the blow count number (N) values which are used to estimate the density of cohesionless soils and consistency of cohesive soils. Other in situ tests which provide data for foundation design are the static cone test, pressuremeter test, vane shear test and borehole shear test. When high quality undisturbed samples can be obtained, lab analysis is the most accurate, but the most expensive.

There are numerous in situ tests because no single in situ test provides all the answers to all the problems. An unfortunate characteristic of most in situ tests is that in general, they do not measure real soil properties; instead they provide some intermediate parameter, such as stress or torque, which is then used to generate a desired soil property by an empirical, semiempirical or, theoretical transformation. Simplifying assumptions are generally associated with the transformation, and therefore the accuracy of individual test results may be directly related to the assumptions. The fact that an in situ test predicts field performance to a high degree of precision may be accidental or predictable, depending on the number of offsetting errors and the quality of the algorithm (1).

Accurate ground water level information is needed for the estimation of soil densities, determination of effective soil pressures and for the preparation of effective soil pressure diagrams. This information is vital for performing foundation design. Water levels will also indicate the construction difficulties which may be encountered in excavation and the level of dewatering effort required.

There are many different types of soils and related problems. Some of these will be discussed next.

Cohesionless Soils

The load capacity of piles driven into cohesionless soil depends primarily on the relative density of the soil. During driving, the relative density is increased close to the pile due to vibrations and lateral displacement of soil. The effect mostly occurs in the immediate vicinity of the pile shaft and extends in gradually diminishing intensity over a zone one to two pile diameters around the pile shaft.

The increase in relative density increases the load capacity of single piles. The pile type also affects the amount of change in relative density. Piles with large displacement characteristics like closed-end pipe piles and precast concrete piles increase the relative density of cohesionless material more than small displacement steel H piles.

The driving process generates high pore water pressures in saturated cohesionless silts which temporarily reduce the soil shear strength and the pile capacity. The gain in capacity with time (setup) is generally quicker for silts than for cohesive soils because the pore pressures may dissipate more rapidly (1).

Cohesive Soils

When piles are driven into saturated cohesive materials, the soil near the piles is disturbed and radially compressed. For soft or normally consolidated clays the zone of disturbance is within one pile diameter around the pile. For piles driven into saturated stiff clays, there are significant changes in secondary soil structure (closing of fissures) with remolding and complete loss of previous stress history effects in the immediate vicinity of pile.

The disturbance and radial compression generate high pore water pressures which temporarily reduce soil shear strength and, therefore, load capacity of piles. As reconsolidation of clay around the pile occurs, the high pore water pressures are diminished which leads to an increase in shear strength and pile load capacity (setup). The zone and magnitude of disturbance is dependent on soil properties of soil sensitivity, driving method and the pile foundation geometry. Limited data suggests that for partially saturated cohesive soils, pile driving does not generate high pore water pressures and hence setup does not occur (1).

When piles are driven in clays, the volume of soil displaced by the pile generally causes a heave of the soil surface. The heave of adjacent piles may also occur, possibly resulting in a reduction in the capacity of these piles. This problem is of particular significance when large pile groups are driven. Experience has shown that the heaved volume at the ground surface is normally of the order of 40 to 60% of the pile volume. If such heave is unacceptable, preboring is the method usually applied to reduce it (5).

Rock

If a foundation can be driven or bored to bedrock, experience shows that a satisfactory foundation will usually result, provided the nature of the rock has been correctly assessed. Pile foundations on rock are normally designed to carry large loads. Usually, the allowable loads on piles driven into rock are based on pile structural capacity while the allowable bearing pressures for bored piles on rock are based on a nominal values of allowable bearing capacity. For pile foundations which are driven to rock, which include steel piles or precast concrete piles, the exact area of contact with rock, the depth of penetration into rock as well as the quality of rock are largely unknown. Therefore, the determination of load capacity of driven piles on rock should be made on the basis of driving observations, local experience and load tests. Rocks may be divided broadly into "soft" rocks such as chalk, weakly cemented sandstones, shales and mudstones, into which the lower ends of piles can be driven and which can be bored by auger rigs, and "hard" rocks that resist pile penetration. The ease with which piling operations can be performed and the ultimate bearing capacity of the resulting pile depend on the intrinsic strength of the rock and on the extent to which the rock mass is fissured. There are unfavorable rock conditions such as cavernous limestone, which can result in excessive settlement and/or failure. A site investigation should show the slope of the rock surface and the rock itself should be sampled by core drilling. The cores will show the rock type and give some indication of the degree to which the rock is broken up by fissuring. Considerably skill is required to avoid overdriving when the pile point reaches the rock surface. Bored piles are normally drilled a nominal depth into the rock (usually one to three times the diameter of the pile (5, p. 271) to ensure the pile bearing is entirely on rock and to extend the pile through the upper, more fractured zones of the rock. If the rock surface is sloping, then attaching an "Oslo" point (a protruding round steel bar 2.9 to 3.9 inches in diameter, with the lower end hollow-ground and hardened) may prevent the pile tip from skidding down the slope (2, pp. 101-103).

In soil containing boulders it is often difficult to differentiate between large boulders and beds of rock, so that caution is needed when interpreting the results if only one or two borings are put down. In such cases local knowledge, well drilling records and data from any geological survey made of the area will give guidance as to the nature of the strata likely to be encountered. When obstructions such as old foundations, boulders, rubble fill, etc., are too deep to excavate, the use of temporary casings or drive shoes and reinforced tips on piles which are strong enough to be driven through the obstruction are recommended.

Other Situations

Piles are commonly driven through alternating layers of competent and non-competent soils. In such cases, the pile foundation is generally designed with the relative stiffness and strength of the different layers penetrated by the pile kept in mind. This provides an idea as to the probable relative contribution of these layers to the pile capacity. The soil profile immediately below the pile toe, which influences the stability and the settlement of the pile is another key to look at in multilayered profiles. The relative contribution of the various layers penetrated by the pile to the capacity of that pile is primarily a function of the relative stiffnesses of these layers and of the type of pile. Piles driven through a multilayer deposit derive their capacities from both shaft and toe resistance. Whenever possible, piles in multi-layered deposits should be driven to a layer of sufficient strength and thickness that it may be assumed that they derive their load capacity entirely from that layer. It is essential to check that the bearing layer extends below the proposed pile toe elevation to a depth sufficient to ensure safety against a punching failure of the bearing layer into a lower weaker material. The design process remains the same, but modifications can be made based on the prevailing subsoil conditions.

When driving a pile near slopes, caution must be taken, particularly in sensitive clays. Driving piles in clays causes the pore water pressure in the clay to increase. The induced pore water pressure may spread through the clay mass and over a considerable distance from the piles. If piles are driven in the vicinity of a slope, the increase in pore pressure produced by driving may cause failure of the slope. Analysis of the stability of the slope before and after driving and instrumentation of the clay layer for monitoring of pore water pressures and soil displacements during driving is recommended. If necessary, pore water pressures can be reduced by preboring or the use of vertical premanufactured drains attached to the surface of the piles, or preferably, installed at the site prior to the pile driving (5, p. 305).

Negative Skin Friction

A common and potentially dangerous field condition affecting pile foundations is a reversal of side friction due to settlement of adjacent soil relative to the pile. In describing the condition of negative skin friction, consider a case where the soil is naturally consolidated and the ground water level is at the surface (2, p. 130). A fill is now placed on top of this existing soil. The pore water pressure, plotted as abscissa from a vertical line AB, will be represented by AC in Figure 3. If a layer of filling AD is now placed, the pore pressure will rise as represented by the line AEC, it being assumed that the bearing stratum below B and the fill above A are both free-draining. As this excess pore water pressure dissipates, consolidation of the weak stratum takes place and it settles. At the place where a pile has been driven, the downward motion of the soil to the full distance is resisted by skin friction at the soil-pile interface, and the soil layers form a cusp as indicated in Figure 4. The downward drag of the soil on the pile is called negative skin friction, and is resisted by positive skin friction on the shaft BF in the bearing layer and by the point resistance. This additional loading, ie. the downward drag of the soil, along with the structural load, as seen on the graph of Figure 5, might produce an unacceptable settlement (S' in Figure 5), greater than what was predicted with the structure loading only, and in some cases, cause foundation failure. Negative skin friction results primarily from consolidation of a soft deposit caused by dewatering or the placement of fill such as:

1. If a fill of clay soil is placed over a granular soil layer into which a pile is driven, the fill will gradually consolidate.

2. If a fill of granular soil is placed over a layer of soft clay it will induce the process of consolidation in the clay layer.

3. Lowering of the ground water table will increase the vertical effective stress on the soil at any depth and induce consolidation in the clay layer.

Down-drag on piles caused by negative skin friction is a settlement problem and rarely a capacity problem. This force increases the pile axial load and can be especially significant on long piles driven through compressible soils and must be considered in pile design. The amount of relative settlement between soil and pile that is necessary to mobilize

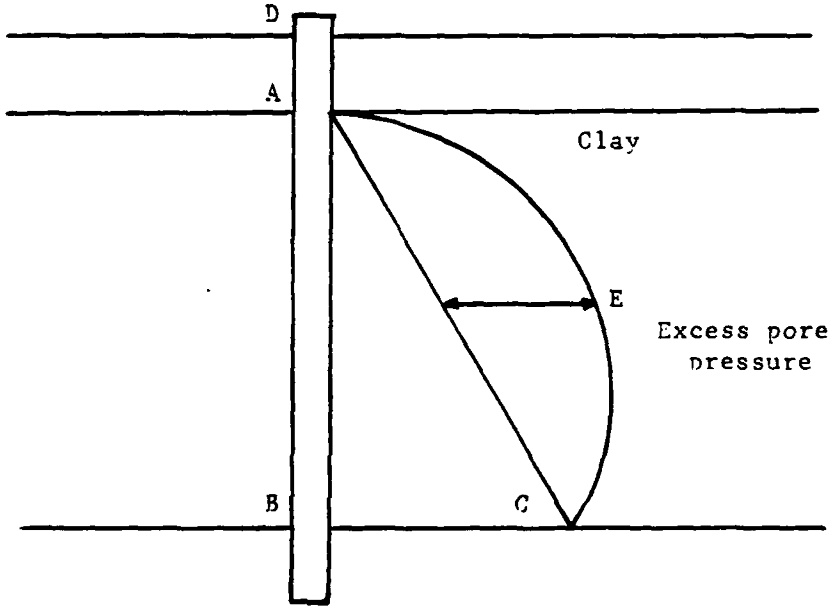


Figure 3. The pore water pressure-depth relationship.

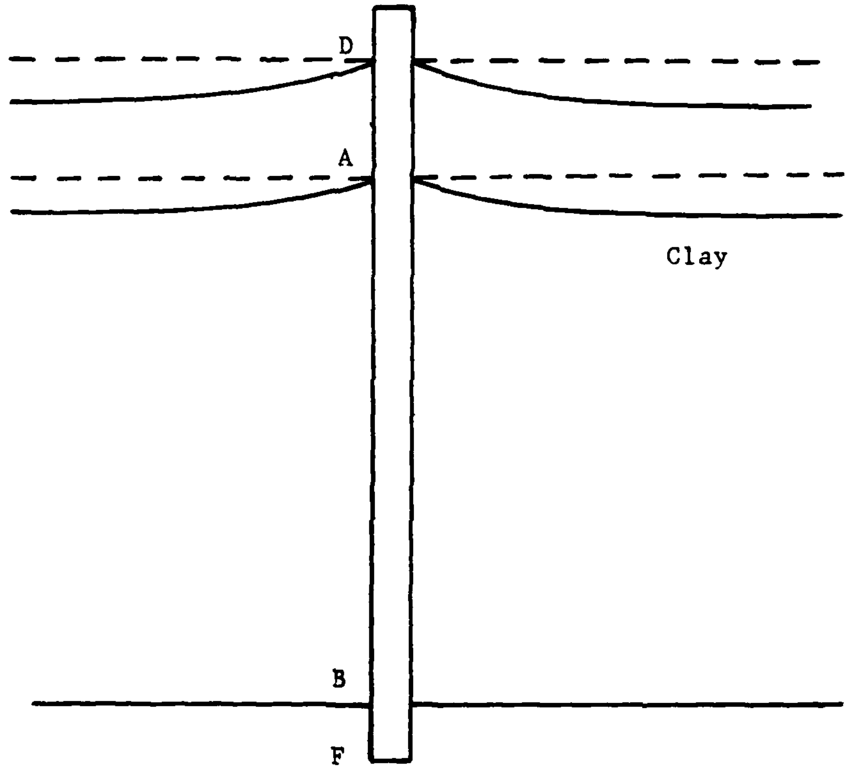
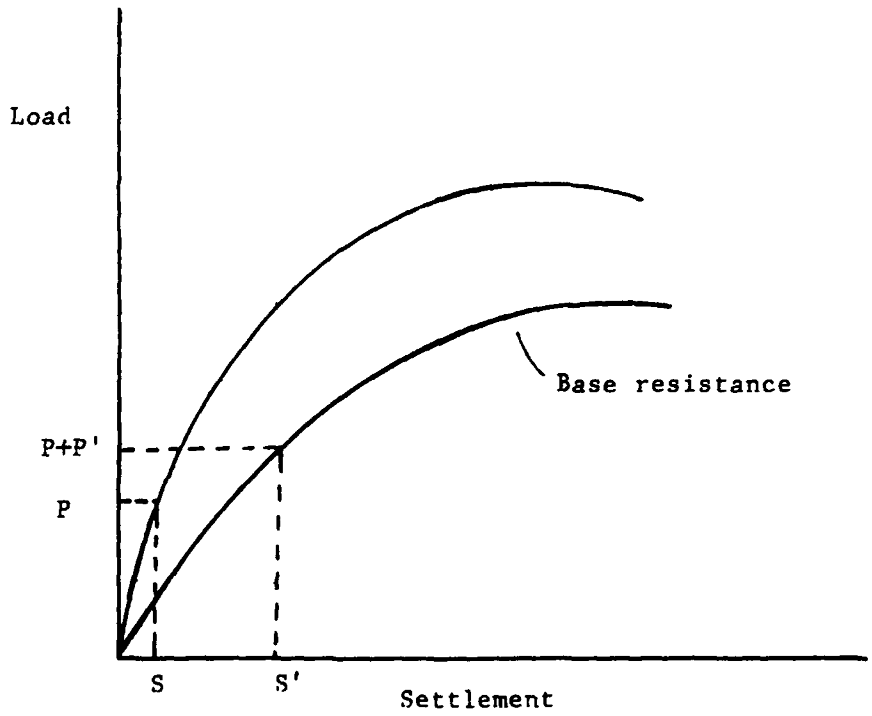


Figure 4. Settlement of the soil around a pile due to consolidation.



P+P' = Applied load plus down-drag load

S' = Settlement due to applied loading and down-drag

Figure 5. The load-settlement diagram.

negative skin friction is about half-an-inch (1, 6). Various methods have been proposed for reducing negative skin friction:

1. the use of slender piles, such as H-sections, to reduce the shaft area subject to drag,

2. increasing the pile length,

3. predrilling an oversized hole through the compressible material prior to insertion of the pile,

4. for bored piles, provide a casing or floating sleeve around the pile to prevent the direct contact with settling soil,

5. coat pile shaft with bitumen to allow slippage.

Bjerrum (7) compared bitumen to an electro-osmosis process for reducing down-drag and reported the latter to be the more efficient. Studies on bitumen, of 80-100 penetration, where a one milimeter thick coating was applied, was found to reduce down drag by more than 50 per cent (6). Fellenius (8), reported a reduction of 90 per cent using the bitumen coating. Fellenius points out that the important factor is to ensure that after installation the coating remains intact. Also, coatings should be applied only to those portions of the pile anticipated to be within a zone of subsidence and the lower portion of the pile (at least ten times the diameter) should remain uncoated so that the full lower shaft and point resistance may be mobilized (2).

Piles driven in swelling clays may be subjected to uplift forces as the result of the swelling process. This is the reverse of negative skin friction. It is best to ensure that the structural resistance of the pile is sufficient to withstand the uplift forces incorporated in the design by providing one or more sections of a diameter larger than the average pile diameter. Expanded base piles, under-reamed and multi-underreamed piles, and screw-piles are typical. If necessary, the uplift forces may be eliminated by isolating the piles from the swelling clay by the use of floating sleeves or bituminous coatings applied to the pile surface (5, pp. 306, 310).

At the end of all subsoil exploration programs, all of the required information is then compiled into a soil exploration report. It contains the scope of the investigation, description of the proposed structure, description of the location site, geological condition of the site, details of the field exploration with a description of the subsoil conditions as determined from lab analysis (ie., standard penetration tests, in situ vane test, etc.), location of the water table, foundation recommendations, and conclusions and limitations of the investigations. Because soil properties cannot be measured with great accuracy and are variable within a building site and the correlation between the soil parameters and the bearing capacity of a pile includes a margin of error and the actual driving or installation conditions vary from pile to pile and cannot be properly taken into account, a large factor of safety is generally used in determining the allowable pile load. The factor of safety depends on the importance of the structure and financial losses in case of failure, and the reliability of the information on soil and environmental conditions. Factors of safety commonly range from 2.0 to 4.0 or higher (9).

The ultimate cost of the structure should also be taken into account while making decisions regarding the extent of the field exploration. The exploration cost generally should be in the range of 0.1 - 0.5% of the cost of the structure (3, p. 65), and the average total foundation costs comprise 3.7% of the total cost of the building (Colin and Steyert study, 1973) (10).

It is not necessary to have a detailed soil exploration report to use PS1, but the more specific the information provided, the better the results. PS1 provides weighted selection of initial pile choices. This initial pile selection is then used to determine the bearing capacity of the soil by any number of well known bearing capacity formulas.

Bearing Capacity

Another important step in the design process is determining the bearing capacity of the soil by any of the following methods: static formula, dynamic formula, WEAP computer analysis, or dynamic load test. It should be noted that design methods for piles in fine-grained soils are in some cases of doubtful reliability. This is particularly so for the bearing capacity of shaft-bearing piles in clays of medium-to-high shear strength. Because of this, pile test loading should be carried out where economically justified or, alternatively, an adequate factor of safety should be used.

Static methods which are primarily based on the principles of soil mechanics are often used to determine the ultimate bearing capacity of the soil. There are several static formula to choose from. According to the U.S Department of Transportation, Pile Group Prediction Symposium, Oct 1987 (11), the two best methods, based on the Standard Penetration Test (STP), for predicting the bearing capacity of a single pile were Coyle-Castello (12) and FHWA (13). The Navy's DM 7.2 (6) method was not included in the U.S. Dept of Transportations symposium and is believed (14) to be as good a method as the FHWA method. The symposium also recommended Schmertmann's (15) method which is based on the Cone Penetrometer Test (CPT).

The FHWA capacity prediction method utilizes Nordlund's (16) procedure for predicting shaft capacity, and Thurman's method (17) for predicting toe capacity. In FHWA, failure is based on the load corresponding to a settlement of one-tenth of the pile diameter (18). It uses a factor of safety of two (14).

The Navy's DM 7.2 method is similar to FHWA, but uses its own charts and tables and some slightly different assumptions. It is based on the Davisson criteria (18) for failure and uses a factor of safety of three (14).

The Coyle-Castello method uses design charts for the unit side resistance and point resistance. These unit resistances are correlated with relative depth and friction angle. Failure is described as the load at which the settlement first reaches 0.05 inch/ton (18). It uses a factor of safety of three (14).

Other static methods (11) for determining bearing capacity are Meyerhof's method (STP) (19), the Briaud-Tucker method (STP) (20), the API method (STP) (21), the Bustamante/Gianeselli method (CPT) (22), and the DeRuiter/Beringen method (CPT) (23).

Dynamic analyses use empirical equations along with hammer data to approximate the ultimate bearing capacity. There are two recommended methods to use (9, 14): the modified ENR (24) and the Hiley formula (25), both use factors of safety of six.

Pile load testing is the actual driving of a test pile at the site location. The usual reason a pile load test would be performed is to verify the design during construction. It is not usually done as a design tool since it is a costly procedure. Typically the test pile is loaded with up to 200% of the design load. It is used to determine the compressive and tensile axial load capacities and lateral load capacities. It is also used to determine compressive axial capacities, internal stresses, pile driver parameters, and assess damage during driving. Butt displacement is also measured.

The Wave Equation Analysis Program (WEAP) (26) is a computer program which is based on the theory of one dimensional wave propagation to analyze piles. It uses the relationships between pile weight, hammer and set, together with stress determination during pile driving. Pile behavior during driving is essentially that of a rod vibrating from being hit on the end with a hammer. That is, the hammer blow causes a momentary longitudinal compression at the end, which moves as a compression wave down the rod. When the wave arrives at the opposite end of the rod or pile, that end momentarily extends, then snaps back, and starts an upward return compression wave. It is the momentary extension at the lower end which pushes the soil aside and drives the pile. Traveling immediately behind each compression wave is a zone of tension. The waves are damped by soil shear along the sides of the pile, and tip resistance.

For the analysis the pile is divided into a series of masses connected by springs which characterize the pile stiffness, and dashpots which simulate the damping below the pile tip and along the pile embedded length. One advantage of a wave analysis is a better prediction of the side shear and end-bearing forces during driving, both being relevant to determination of a design load for the pile as well as the drivability. This method could be used in relating the static bearing capacity of a pile with its dynamic behavior. By inserting various soil properties, pile properties, and hammer properties, the WEAP program will calculate the blow count versus ultimate capacity, as well the stress in the pile, and pile displacement.

It is recommended that the WEAP program be utilized along with those static methods recommended by the 1987 Pile Group Prediction Symposium to approximate the ultimate capacity of the pile type chosen.

Dividing the ultimate capacity calculated by its factor of safety produces the design load capacity. Dividing the design load capacity into the total structure load indicates the number of piles required to sustain the structure.

Sometimes overlooked is pile settlement analysis since piles are selected over shallow foundations because of minimal settlement criteria. There are a few methods for predicting pile settlement, but two are recommended (14): Vesic's method (27) and Briaud-Tucker method (20).

Vesic's method divides the total pile head movement into three components: the elastic compression of the pile, the settlement beneath the toe, and the settlement caused by the load transmitted along the pile shaft. This is valid only for the ultimate loading.

The Briaud-Tucker method is based on a 33-pile data base, and on the results of standard penetration tests, to produce a load versus pile movement curve. This method can also be used as a static formula to check the bearing capacity of the soil.

A pile loaded by lateral thrust and/or moment at its top, resists the load be deflecting to mobilize the reaction of the surrounding soil. The magnitude and distribution of the resisting pressures are a function of the relative stiffness of pile and soil. Design criteria is based on maximum combined stress in the pile, allowable deflection at the top or permissible bearing on the surrounding soil. Although 1/4-inch at the pile top is often used as a limit (6, p. 234), the allowable lateral deflection should be based on the specific requirements of the structure. Lateral loads commonly are imparted to pile foundations from earthquakes, bridge pier supports, wharves, piers, and other offshore structures, railway bridges, and machinery vibrations. The design of vertical piles to resist lateral forces is complex, and covered in standard references (29, 30, 31, 32).

CHAPTER III

DEVELOPMENT OF PS1

Basic Program Structure

This chapter documents the development of PS1, and expert sytem, for initial pile selection. The system PS1 was written in FLOPS (33), a Fuzzy Logic Production System shell.

Becasue the expert system PS1 was written in FLOPS, it is often referred to as a FLOPS program in this thesis. A FLOPS program may be grouped into three sections: the declaration section, similar to that of the C or PASCAL language; the rules section where the actual rules appear; and the input section, in which actual values are assigned to the attributes described in the declaration section.

The program uses a blackboard system for input data, intermediate data and final output. Production rules are written in FLOPS with automated generation scheme. This provides a powerful tool to generate hundreds of rules from a few rules written in PS1. External routines written in the C programming language are linked to the FLOPS program for faster computation.

To run the program PS1 using FLOPS, first type "pflop" at the A>, B>, or C> prompt. Then at the >> prompt, type "open PS1.c;". The program PS1 will now begin to run.

The content of PS1 is listed in Appendix A for easy reference, and its basic structure is shown in Figure 6. PS1 first reads external expert knowledge data files, DLKB.c (Design Load knowledge base), LTHKB.c (Length knowledge base), SCKB.c (Soil Condition knowledge base), NEKB.c

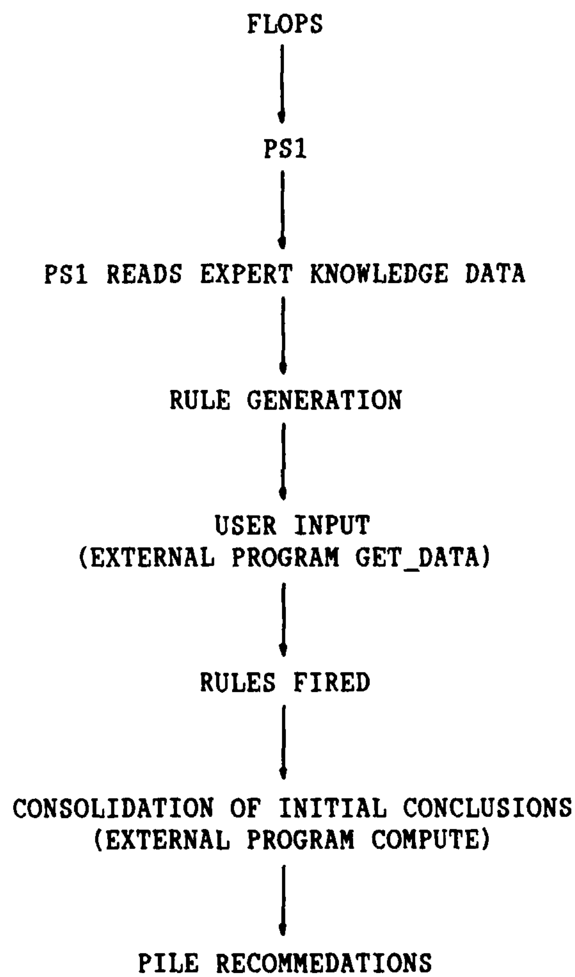


Figure 6. Basic PS1 structure.

(Noise Effects knowledge base), VEKB.c (Vibration Effects knowledge base), AOMKB.c (Availability of Material knowledge base), and LEKB.c (Local Experience knowledge base) and begin to generate new rules. All of the expert knowledge data files can be seen in Appendicies D through J. Using these knowledge and some "metarules", the program generates all possible production rules. After the rules are generated, the program will begin to ask for user input data by calling an external program "GET_DATA" (Appendix B). The program GET_DATA was written in the C language and compiled with Microsoft C5.1 compiler (50). It serves as a user interface which allows for the problem-specific data to enter the FLOPS program. The external program GET_DATA will create a data file called "user.dat" (Appendix K, L, and M) to store user input. This "user.dat" file is then transfered to PS1 and matched with generated rules (when input matches a generated rule, it causes that rule to fire). Execution of the production rules now begins, which produces several preliminary conclusions. These conclusions are stored in the file "pile_type.dat" (Appendix K) and contain the confidence levels of the fired rules. This new file will be transfered to "COMPUTE" (Appendix C), an external program called by PS1. COMPUTE, also written in the C language, consolidates the preliminary conclusions and creates two output files, "flops.out" and "class.out". The file "flops.out" will be printed on the screen. The file "class.out" (Appendix K, L, and M) contains more information and may be seen by using the "type class.out" command at the A>, B>, or C> prompt.

To get the best results from PS1, it is best to have an idea of the foundation loads to be supported, soil properties, approximate pile length, availability of material, local contractor experience, and noise or vibration constraints.

The key of pile types used in the expert knowledge data files and used to describe each file on the following pages is shown in Figure 7.

KEY

A = PRECAST CONCRETE PILES
B = PRESTRESSED CONCRETE PILES
C = CAST-IN-PLACE (W/MANDREL)
D = CAST-IN-PLACE (W/OUT MANDREL)
E = STEEL PILE (H OR PIPE)
F = WOOD PILE
G = COMPOSITE WOOD PILE
H = COMPOSITE STEEL PILE
I = PRESSURE INJECTED PILES
J = AUGER PLACED CONCRETE
K = BORED/CASSION PILE

Figure 7. Pile classification key.

It is noted that the expert system PS1 was written in parallel FLOPS. Thus, it does not involve backtracking of the rules. Instead, at any stage of rule firing, all fireable rules are fired at the same time. The problem of memory conflicting was resolved using weakly monotonic logic, a type of fuzzy logic in which the value of a datum may be replaced by a new value, if the confidence in the new value is greater than or equal to the old confidence (33). However, the preliminary conclusions reached at different stages were treated as evidences, each based on a particular knowledge source. They were not combined with the weakly monotonic logic. Instead, an external program COMPUTE was used to combine these evidences.

Flops Features

FLOPS is an expert system shell written in C language by Siler and Tucker (33) for use in the MS-DOS or compatible DOS environment on microcomputers. It is based on the Artificial Intelligence (AI) language

OPS5 by Forgy (34). Thus, FLOPS basic syntax is the same as that of OPS5. However, FLOPS has several unique features which provide a great deal of power. A summary of the FLOPS features employed in the development of PS1 follows.

Approximate Reasoning with Fuzzy Logic

FLOPS uses fuzzy logic invented by Zadeh (35,36). Since Zadeh's (35) pioneer work, the theory of fuzzy set has developed into an important branch of mathematics. The subject of civil engineering applications of fuzzy sets is beyond the scope of this paper, but many of them have been documented elsewhere (37-48). FLOPS employs fuzzy set theory in three aspects:

1. fuzzy logic to handle approximate reasoning,
2. fuzzy number to represent number with uncertainty, and
3. fuzzy set to deal with ambiguity.

Using fuzzy sets, FLOPS can represent uncertainties and ambiguities in the data as well as in the knowledge itself in a more natural way. In addition, fuzzy logic permits a proposition to be true to a wide range of degrees of certainty, which comes much closer to real-world reasoning than that using classical logic.

In binary terms, a statement is either true (1.0) or false (0.0). In fuzzy logic this concept of true or false is generalized. A statement can be partially true, with a degree of trueness of 0.6 for example. This generalization of degree of trueness allows for a more realistic modeling of many real world situations, since few situations of a complex nature are absolute. With all the uncertainties involved in pile selection, it becomes difficult to absolutely eliminate a pile choice from a given construction scenario. PS1 incorporates an expert knowledge data base with potential pile choices ranging in trueness or membership from 0.5 to 1.0 for specific site conditions, and represented as 500 to 1000 in FLOPS. Currently FLOPS and PS1 use all the memory capacity of a 640 K personal computer. This is unfortunate since a better pile selection data base would have memberships which range in value from 0 to 1000. The current expert knowledge data base was developed from personal knowledge and current literature on pile selection and design.

Deductive and Inductive Reasoning

FLOPS is a production system and as such, its basic element is the "rule". The deductive logic implemented in FLOPS is no different from most expert systems. It fires the production rules sequentially. If the data permits more than one rules to be "fireable", deductive systems select one rule for firing; other fireable rules are stacked for backtracking later. FLOPS on the other hands, also implements inductive reasoning which considers many possible outcomes at once. FLOPS parallel rule-firing scheme for implementing the inductive reasoning is rather unique. All fireable rules are fired concurrently, and thus no rules remain to be stacked for backtracking. Instead of implementing a rule conflict algorithm, FLOPS adopts a weakly monotonic fuzzy logic for its truth maintenance to resolve a memory conflict problem. When applicable, the inductive mode of FLOPS is much faster than the deductive mode to reach a conclusion. Both forward chaining and backward chaining inference mechanism are available in FLOPS.

Blackboard Architecture

FLOPS employs a relational structure for data stored on a blackboard, a disk on microcomputers. The ability of one FLOPS program to call another and to exchange data through the blackboard could overcome memory limitations of small microcomputers.

Expert knowledge is divided into two classes in FLOPS. One is factual knowledge, which belongs in a data base. The other is expert skills, which belongs in rules. One of these expert skills knows how to use the expert factual knowledge; in FLOPS, this means writing rules to generate other rules. In other words, the programmer-written rules can generate the production rules based on the factual knowledge during the program execution.

Two methods of communicating with external programs are available in FLOPS. One type of call transmits a command string to the called program in the DOS environment. The other is a call by reference to a C program, and thus requires to follow the calling convention used in the C language.

Data Type

FLOPS implements seven data types in two major classes. One is to be declared by the programmer. It includes character string, integer, floating point number, fuzzy number, and fuzzy set. The second one is reserved to be used by the system itself. It includes the confidence level of an attribute and the time tag of an instance of a memory element.

Details of the above features as well as others can be found in FLOPS manual (33).

Detailed Program Structure

Detailed comments on the development of PS1 will follow, with reference being made to Appendix A. The command literalize declares "start" as a memory element with 4 attributes of the type atm. In FLOPS, the data type atm is for character string, and the data types flt and int are for floating point number and integer respectively. The syntax for the declaration is very similar to that of a structure in the C or PASCAL language. An attribute is analogous to a variable in C or PASCAL. Notice that a semicolon is only needed at the end of the entire literalize command, and that separation of the literalize command into several lines is a programming style for ease of reading and maintaining of the code.

Memory element "xdata" is needed to store the user-supplied data. Elements "DL", "LTH", "NEX", "VEX", "AMOX", "LEX", and "SOIL" are used to store expert knowledge data. Element "pile_type" has seven attributes: criterion1, criterion2,..., and criterion7. These seven attributes are declared to be of data type fzset, which stands for fuzzy set. The fuzzy set data type in FLOPS is very unique. In common fuzzy set notation, using rating1 as an example, it may be expressed as:

```
criterion1 = {m1/PC, m2/PSC, m3/CIPM, m4/CIP, m5/STL, m6/TM,
              m7/CPW, m8/CPS, m9/PIC, m10/APC, m11/bp},
```

where

m1,..., m11 are the membership grades for the corresponding members "PC",..., "BP", respectively.

In FLOPS, these membership grades appear in the form of a confidence level. The confidence level is an unique data type, which is used to store the confidence toward a datum or a member of a fuzzy set. The attribute criterion1 is created to store the preliminary conclusion on of pile selection based on design loads. The attributes criterion2,..., and criterion7 are declared in the same way. Although these seven attributes look alike, use of different attributes is necessary to preserve multiple preliminary conclusions reached at different stages in the inductive reasoning process. Otherwise, FLOPS weakly monotonic logic may eliminate the desired membership values of the these fuzzy set members.

For this particular system it was determined a parallel (inductive) FLOPS program is more effective than a sequential (deductive) one. It was also decided to set up a block firing control scheme. This scheme ensures the sequential firing of each block of rules. Within each block, however, the parallel processing ensures all rules that are fireable are fired at once.

When block #0 rules are fired, the system reads in expert knowledge files. With this knowledge, part or all of the block #0 rules become fireable and are fired at once. The actions of firing these rules generate the rules of blocks #2 through #8. It is noted that without proper initiation of data, no rules can actually be fired.

Rule Generation

As an example to explain how rules are generated, focus on rule r0. The expert factual knowledge was stored in a file named "DLKB". As soon as it is transferred to the system (using the command open, see comments on input section), the left-hand-side (LHS) of the rule r0 will be satisfied. In other words, the rule r0 becomes fireable; and when it is fired, the right-hand-side (RHS) of the rule r0 will be executed. The content of the file "DLKB" can be seen in Appendix D. It basically consists of a set of make commands. This command initiates a memory element and assign values to its attributes. For example, when the file "DLKB" is open, the first make command assigns the following data:

```
^lower1 = 0
^upper1 = 29
^fsmember = CIPM
^confidence = 600.
```

Notice that ^X is the symbol used in FLOPS for the value of the attribute X. With these values transferred to the system, the variables in the LHS of the rule r0 take on these values:

```
<LB> = 0
<UB> = 29
<FSM> = CIPM
<CONF> = 600.
```

When the rule r0 is fired, the action part (i.e., the RHS) of the rule yields a new rule:

```
rule 600 2 (xdata ^design_load >= 0
           ^design_load < 21)
           (pile_type ^criterion1.cipm = 0)

      ---> Modify 2 ^criterion1.cipm;.
```

Whether the new rule is fireable depends on the actual attribute values in the elements "xdata" and "pile_type". Notice how a membership grade of a member in a fuzzy set is represented. The term "^criterion1.CIPM" represents the confidence level (membership grade) toward the member "CIPM" in the fuzzy-set attribute "criterion1".

All of the make commands in the file "DLKB" match the pattern of the LHS of the rule r0, thus a set of 75 rules are generated. Such a program design is convenient in maintaining the system. When expert opinions change, we need only to change the content of the knowledge file. We may even create a user interface to facilitate the editing of the knowledge file.

As a final note on rules in FLOPS, let's look at the first rule command shown in block zero. A number 1000 appears right after the key word rule. This number is referred to as the priority of the rule, or the prior confidence level of the rule. In FLOPS, the confidence level is encoded as an integer with a maximum value of 1000. The number 1000 actually means a confidence of 100%. When the LHS is evaluated, it also returns a confidence value. The smaller of the two confidence values is taken as the posterior confidence level. All actions involving memory updating in the RHS of that rule are assigned this posterior confidence value. The PS1 system utilizes this feature to assign the membership grade of a member of a fuzzy set. The modify command in block zero is an example. It is noted that the second rule command has a number 2 beside 1000. This is referred to as a block number. When that number does not appear, as in the case of the first rule command, the system assigns a number of 0. The block numbers are generally used to group rules for some rule firing control. It is a useful feature, especially for inductive reasoning in FLOPS.

The input section basically consists of at least a make command. The make command is used for non-interactive input or initiation of the elements and their attributes. The run command, although can be issued from anywhere in FLOPS environment, is usually placed in the input section. This command causes execution of the rule section. The input section may include other commands for specific purposes. All commands are executed sequentially in the input section.

As mentioned earlier, all production rules are grouped into blocks. By controlling the block firing sequence, the rules may be fired in some planned order. However, no particular order is set for the rules within a block. In fact, with parallel processing, all rules fireable will be fired at once regardless their order of appearance.

The program structure shown in Figure 6 was implemented in this input section. First, the system reads in the knowledge files. It then sets up a control mechanism to execute each block of rules sequentially. The system begins with execution of block #0 under the command run. This action generates all possible production rules.

With the next run command, the system first executes block #1 which gathers problem-specific data by calling an external program GET_DATA. Actions taken in block #1 also make block #2 through #8 fireable. These blocks are then fired sequentially. It is noted that block #1 through block #8 are all executed under this run command. An external data file is created after firing of these blocks. The last run command in the input section causes execution of block #9. This block calls an external program COMPUTE for fuzzy combination of evidences. The result is reported and the program then stops.

Notice that with the implemented structure described above, a rule will be fired only if all of the following conditions are met:

1. the block in which the rule resides is switched on,
2. the elements used in the LHS of the rule have been initiated with proper make command, and
3. the LHS of the rule is evaluated to be "true".

For example, the first run command is proceeded by the elements initiation commands and block enabling commands. These commands plus the run command cause the execution of block #0 and #99. However, whether the rules in the blocks #0 and #99 will be fired still depend on their LHS comparison. In the present case, they are fireable. Although the actions taken in the rules of blocks #0 and #99 do "turn on" block #1, and the run command in parallel FLOPS is supposed to "run" all blocks at once, the rules in block #1 can not be fired because the element "start" has not yet been initiated. The make command and the run command cause firing of rules in block #1.

As a final note on block firing control, let's look at block #99. This block is always enabled. The only rule in this block causes all other blocks to be turned on and off sequentially. This is a simple mechanism on which the PS1 system is based.

External Programs

External program used in the system is treated as a command in the DOS environment and as such, it is communicated with the FLOPS program through a call command with name of the executable program as the only argument. For example, the RHS of the rule consists of two calls to the DOS commands. One is an executable program COMPUTE, treated as a command. The other is a true DOS command pause. Although FLOPS allows for direct call by reference (address) to a program written in C, it is considered to be advantageous to adopt the former method for this particular expert system.

The two external programs used are GET_DATA and COMPUTE, both written in C language and compiled by using Microsoft C5.1 compiler (50) for use in the DOS environment.

The program GET_DATA is used for gathering problem-specific data regarding the pile and site information. It is noted that GET_DATA is itself a complete program, and can be run separately in the DOS environment. In fact, it is often run separately to edit the data file to be used in the PS1 system. The program GET_DATA essentially serves as a user interface to the PS1 system. The program is considered to be very user-friendly.

The program COMPUTE is used for consolidating the preliminary conclusions reached by the system. The data needed for running the program

COMPUTE is created by the system and stored in an external file called "pile_type.dat". The file "pile_type.dat" is an ASCII stream file. The data in this file represents the preliminary conclusions reached by the PS1 system. These data are the degrees of confidence toward each member of the fuzzy-set attributes. As defined in the PS1 system, the members of these attributes are: PC, PSC, CIPM, CIP, STL, TM, CPW, CPS, PIC, APC, and BP. Each preliminary conclusion was reached based on each of the seven geomechanics criteria employed. An example of these conclusions, using criterion1 and a design load of 29 to 39 tons are as follows:

{0.8/PC, 0.8/PSC, 1.0/CIPM, 0.5/CIP, 0.0/STL, 0.7/TM, 1.0/CPW,
1.0/CPS, 0.0/PIC, 0.8/APC, 0.0/BP },

where the values are the confidences toward the individual members.

It is noted that in FLOPS notation, the value 0.8 is stored as 800, and 1.0 as 1000, and so on. In the above example, the PS1 system very strongly supports the statement that cast-in-place concrete pile driven with a mandrel (CIPM), supports precast concrete (PC) piles, weakly supports cast-in-place (CIP) piles, and does not support steel (STL) piles under this design load criteria.

Pile Classification

Piles can be classified in different ways. They can be classified by material type, installation type, bearing type, or displacement type. Figure 8 shows a flow chart of piles classified by material type (1).

Displacement Piles

Displacement piles, or driven piles are piles which are driven into the ground and push the soil out as they descend into the soil. Consequently, driven piles displace aproximately 40 to 60 % of the pile volume

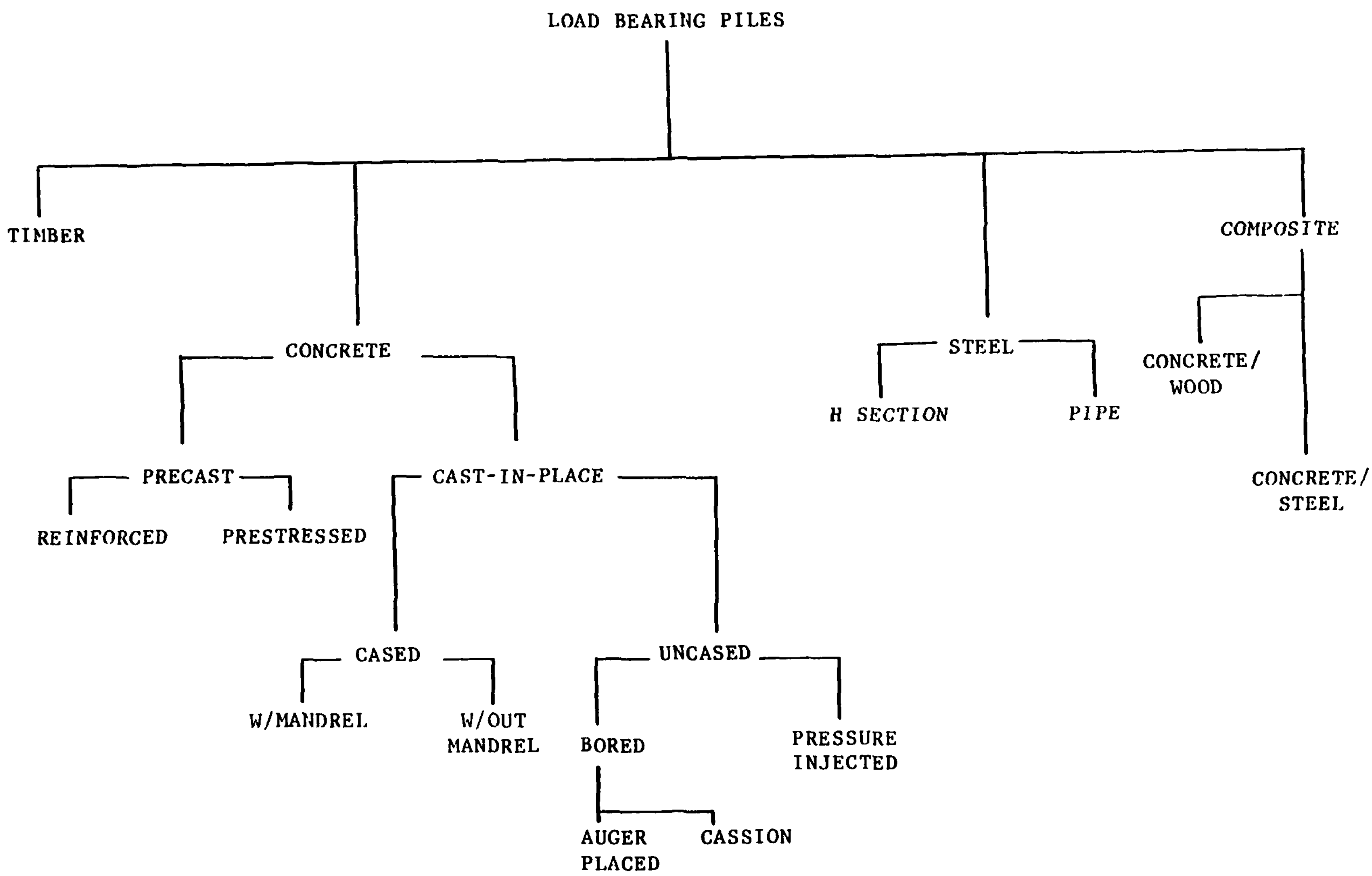


Figure 8. Pile classification.

(5). Driven piles can be divided into several categories; steel piles, concrete piles, wooden piles, and composite piles. Each will be discussed in detail.

Steel Piles

Steel piles are generally either pipe piles or rolled steel H or I-section piles. Pipe piles, which have diameters up to 23.6 inches, can be driven with their ends open or closed. H and I section piles, along with open ended pipe piles which do not form a soil plug, can be thought of as low displacement piles. In many cases, the pipe piles are filled with concrete after driving. Sometimes, they may be filled with a structural shape such as an H-section in addition to the concrete and socketed into bedrock. These steel/concrete piles are considered as composite piles. Close ended pipe piles are considered high displace-ment piles. Steel piles are commonly used for any depth, since they come in various lenghts and sizes and can be easily spliced by full penetration butt welds or riveting when needed. However, long H-piles are prone to bending and doglegging, and the straightness of the H-pile cannot be inspected after driving. Commonly, installation problems with H-piles originate with the use of too small a section of pile. Therefore, caution is recommended when the length to slenderness ratio becomes large. Open ended steel piles may cause high pore-water pressure in fine grained soils. This pressure may cause thin walled pipe piles to collapse or deform. When hard driving conditions are expected, such as driving through dense gravel, shale, and soft rock, the steel piles can be fitted with driving points or shoes. To minimize damage during driving, it is advantageous to use a pile with a high-yield strength when possible.

Experience indicates that corrosion is not a practical problem for steel piles driven in natural soil, due primarily to the absence of oxygen in the soil. However, steel piles may be subject to moderate corrosion in certain conditions. For example, swamps, peats, and other organic soils/fills are corrosive at/or above the water table. Soils that have a pH greater than 7 are not very corrosive. To compensate for the corrosion, either an additional thickness of steel can be added to the design thickness or a factory-applied epoxy coating works satisfactorily. Sometimes concrete encasement of steel piles in very corrosive zones is practiced as a protection against corrosion. Steel piles are suitable for use as friction piles, end bearing piles, and combinations of these two (1, 2, 3, 5, 6, 9, 51, 52, 53).

Concrete Piles

Precast concrete piles can be manufactured by using ordinary reinforcement or prestressed by the use of high-strength steel prestressing cables. The reinforcement is provided to enable the pile to resist the tension wave which follows the compression wave during vertical loading and the bending moments which develop during transportation and lateral loading. Tensile stresses are high when the soil penetration resistance is low. Reinforced concrete piles as compared to prestressed piles are more susceptible to damage during handling and driving. Reinforced concrete piles are rarely used nowadays. Prestressed concrete piles are more durable than reinforced concrete piles. Frequently such piles are cast with a hollow core which may be used for placing instrumentation during construction or for determining pile damage. They can be square or octagonal in cross section and are considered high displacement piles. The piles are cast to desired lenghts and cured before being transported to the work sites. Comparatively, they have a high initial cost. Because of their structural strength and wide choice of possible dimensions, precast and prestressed concrete piles can have a wide range of loading. They are suitable for use as friction piles when driven in sand, gravel, or clays; suitable for use as end bearing piles; suitable for resisting uplift forces, when designed for it; and suitable for driving in soils containing boulders, when correctly designed. They are considered non-corrosive but can be damaged by direct chemical contact with organic soils/fills or industrial wastes. Concrete can be protected from chemical corrosion by use of special concrete or special coatings. Typical depths range from 40-50 ft for precast reinforced concrete piles and 60-100 ft for prestressed concrete piles without splicing devices, and up to greater depths with splicing devices. However, prestressed concrete piles are difficult to splice. Two installation problems commonly arise. Regular horizontal tension cracks may form in the early stages of driving, when the resistance to penetration is low. Such cracks, where visible above ground, frequently indicate severe damage below ground, sometimes even loss of a portion of the pile. In hard driving, the pile toe or pile head may be crushed in compression (1, 2, 3, 5, 6, 9, 51, 52, 53).

Cased Concrete Piles

Cased piles are made by driving a steel casing into the ground with or without the help of a mandrel placed inside the casing. When the pile reaches the proper depth, the mandrel is withdrawn and the casing is filled with concrete. The cased driven shell concrete pile is the most widely used type of cast-in-place concrete pile. Those driven without a mandrel can be redriven because the shell thickness is thicker and less susceptible to driving damage. They are specifically designed for a wide range of loads and they are initially economical. The shell is driven into the soil and then filled with concrete. Unfilled shells are inspected internally for damage. Those driven with a mandrel are susceptible to collapse under hydrostatic pressure before being filled with concrete. Mandrel driven shells can be used in almost any soil except where obstacles such as cobbles and boulders that could rip the shell are present. They are best suited for friction piles in granular material. They cause considerable displacement and are difficult to splice after concreting (1, 2, 5, 6, 9, 51).

Pressure Injected Concrete Piles

Pressure injected concrete piles require the use of special installation equipment handled by persons experienced in the installation work. Damp concrete is rammed through a drive tube into the soil. The pressure injected piles may be reinforced. Pressure injected piles develop their bearing capacity from the densification of soil around the expanded base. They are suited for piles in granular soils, in particular in loose sands, where high capacities can be developed at shallow depths, and for piles subjected to uplift forces provided they are structually designed for this condition. They are usually unsuited for loose granular soils containing more than about 15 to 20 % of fine grained soil, or where special measures are needed to ensure the integrity of the base and shaft. High pore water pressure, either existing or those induced in the soil by the driving, may lead to necking or contamination of the shaft. Also, heave and displacement caused by the driving of nearby piles, cause many failures (1, 2, 6, 51).

Timber Piles

Timber piles are tree trunks that have had their branches carefully trimmed off. The maximum length of most timber piles is 33-66 feet. No special considerations need be given to handling stresses. To qualify for use as a pile, the timber should be straight, sound, and without any defects. They may be used untreated where they are entirely located below the permanent water table. Permanently under water, they are resistant to decay, irrespective of the quality of groundwater. In a fluctuating marine environment, the most common method of protection is pressure creosote treatment. The ASCE's Manual of Practice, No.17 (54), classifies timber piles into three categories (9):

1. Class A piles: These piles carry heavy loads. Minimum diameter of the butt should be 14.0 inches.

2. Class B piles: These are used to carry medium loads. Minimum butt diameter should be 12 - 13 inches.

3. Class C piles: For use in temporary construction work. They can be used for structures on a permanent basis when the entire pile is below the water table. Minimum butt diameter should be 12 inches.

In all three cases, minimum tip diameter should be 5.9 inches. Wood piles are best suited for low-velocity hammer blows. Timber piles cannot withstand hard driving stress; therefore, the pile capacity is generally limited to about 40-80 tons. Steel shoes may be used to avoid damage at the pile tip. A metal band or cap may also be used to avoid damage to the pile top caused by hammer impact. They are considered as high displacement piles. Comparatively, they are low in initial cost. Wood piles are best suited for use as friction piles in sands, silts, and clays. They are not recommended for driving through dense gravel or till, or for toe-bearing piles to rock, since they are vulnerable to damage both at the head and toe in hard driving. Timber pile splices are generally undesirable (1, 2, 3, 5, 6, 51, 52, 53).

Composite Piles

Composite piles are ones in which the upper and lower portions of the pile are made of different materials. Steel and concrete piles generally consist of a steel pipe filled with concrete. An H beam may be placed in the center of the concrete pipe for additional load carrying capacity. Steel and concrete composite piles can also consist of a lower portion of steel and an upper portion of cast-in-place concrete. This type of pile is the one used when the length of the pile required for adequate bearing exceeds the capacity of simple cast-in-place concrete piles. Timber and concrete piles usually consist of a lower portion of timber pile below the permanent water table and an upper portion of concrete. The load capacity of a wood composite pile is limited to the capacity of the wood. Composite piles (with the exception of concrete filled steel pipes) are not widely used because of the difficultly in forming proper joints between two dissimilar materials (1, 2, 3, 5, 6, 51, 52, 53).

Non-Displacement Piles

Non-displacement piles, or bored piles, are being used increasingly because of their high load capacities. Bored piles are installed by drilling or augering a hole in the ground and filling it with concrete as the drive casing or auger is withdrawn. Bored piles are usually reinforced with structural steel cages. They are frequently separated into "normal" bored piles, that is piles up to about 600 mm diameter and "large" bored piles which are greater than 600 mm and sometimes called drilled piers or caissons. Normal bored piles are similar to cased piles without the casing. Uncased piles are made by first driving the casing to the desired depth and then filling it with fresh concrete. The casing is gradually withdrawn in steps. Both normal and large bored ples may have a pedestal at the bottom. The pedestal is an expanded concrete bulb that is formed by dropping a hammer on fresh concrete.

Bored piles are best suited for end bearing, high capacity piles to rock or dense till and successfully used in uniform deposits of firm or stiff clays. Bored piles are commonly used for variable lengths. They have been excavated with bentonite slurry to depths in excess of 100 meter and diameters ranging up to three meters. The borehole would be put down by an auger or by percussion with a core cutter. Concreting should begin quickly after the hole is bored to minimize the deterioration of the sides from the induced migration of pore water from the surrounding clay to the surface of the bore hole. Where deposits of loose cohesionless materials have to be penetrated, or where artesian groundwater conditions prevail, it may be necessary to resort to the use of bentonite slurry or temporary casing.

The use of bored foundations has several advantages:

- A single cassion pile can be used to replace a group of piles and the pile cap.
- It is easier to construct bored piles in deposits of dense sand and gravel than it is to drive piles.
- They are applicable for a wide variety of soil conditions.
- Construction of bored piles can be completed before the completion of grading operations.

- When piles are driven by a hammer, the ground vibration may cause damage to the nearby structures. The use of bored piles does not present such hazards.

- Piles driven into clay soils may produce ground heaving and may also cause previously driven piles to move laterally. Such conditions do not exist in the construction of bored piles.

- There is no hammer noise during the construction of bored piles, as there is during pile driving.

- They can be drilled into bedrock to carry very high loads.

- Because the base of a normal bored foundation can be enlarged, it provides great resistance to an uplifting load.

- The design bore depths and diameters can be readily modified based on field conditions.

- The surface over which the base of the bored pile is constructed can be visually inspected.

- Construction of bored piles generally requires light, mobile equipment.

- Under proper soil conditions, they may prove to be more economical than displacement pile foundations.

- Bored piles have high resistance to lateral loads.

There are also some drawbacks to the use of bored foundations. The disadvantages are:

- The concreting operation depends on more than average quality of workmanship requiring very close supervision and can be hampered by bad weather.

- Danger of lifting concrete when pulling the casing can result in voids or inclusions of soil in the concrete.

- Loose granular soils below the water table can cause construction problems.

- A pedestal usually cannot be formed in granular soils below the water table.

- Small diameter bored piles (less than 30 inches) cannot be easily inspected to confirm bearing and are particularly susceptible to necking problems.

- A more detailed soil exploration is generally required before making decisions about bored piles than in the case of other types of foundations.

- Deep excavations for bored piles may induce substantial ground loss and damage to closely located structures.

A very general rule rule of thumb states that for non-cohesive bearing stratums, the use of a displacement or driven pile is usually the most suitable. For cohesive bearing stratums, replacement piles are usually the most suitable. In rock bearing stratums, driven cast-in-place piles and straight shafted replacement piles are probably the most suitable (1, 2, 3, 5, 6, 9, 51, 52, 56).

Design Load Capacities

The design loading expert knowledge data (DLKB) file is the first file read by PS1. The file is made of a set of make commands, which are special FLOPS commands, with lower and upper limit load constraints for invidual pile types. The pile versus loading chart is presented in Figure 9. The entire DLKB data file can be seen in Appendix A.

Light structures such as one or two story buildings may require the use of small diameter piles and very heavy structures such as high buildings over eight stories, will prohibit the use of timber piles, and small diameter replacement piles. In particular, tall or heavy structures involving large concentrated loading points may strongly indicate the use of large diameter replacement piles. For all other structures including residences up to eight stories, offices up to seven stories, hospitals and other civil engineering structures, virtually any type of pile may be suitable.

Soil Parameters

The primary requirements of a site investigation are that it should describe the ground conditions sufficiently well to enable a suitable bearing stratum to be chosen, and that it should extend to a sufficient

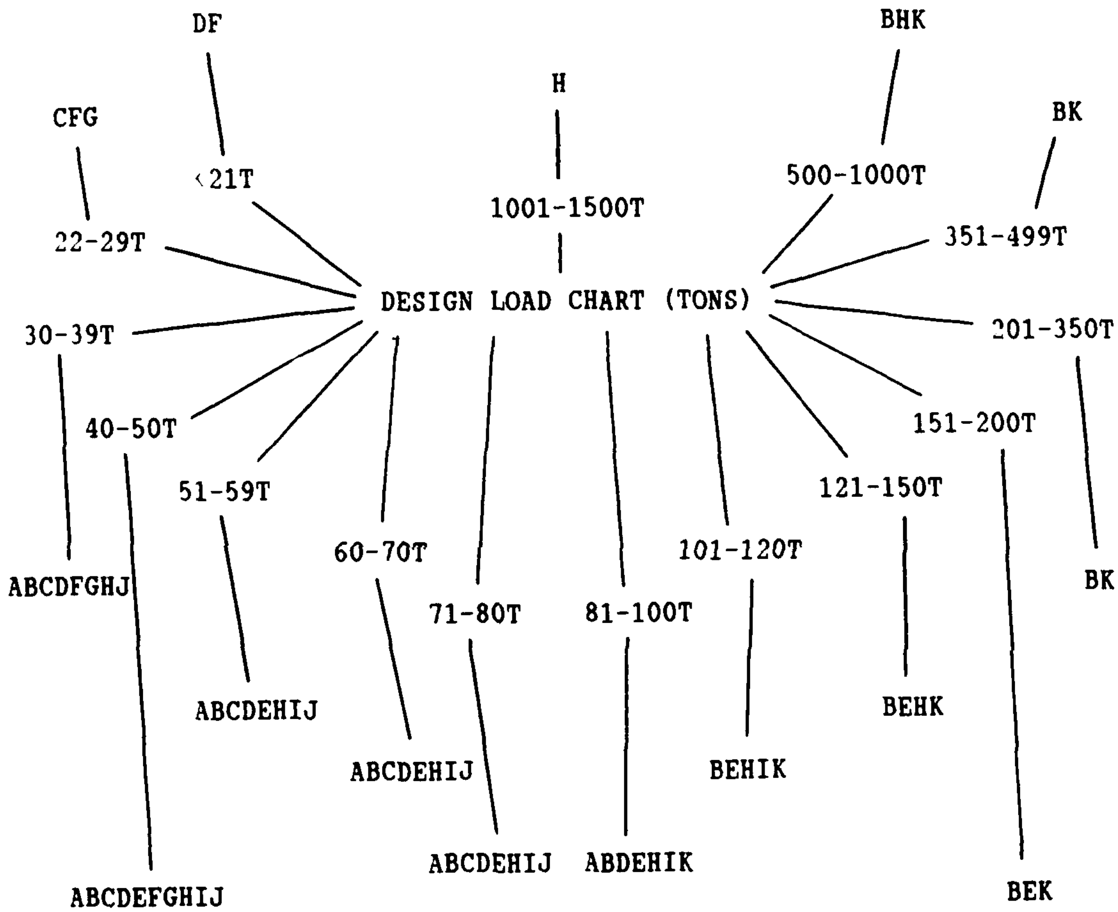


Figure 9. Design loading chart.

depth below the likely level of the pile points to give information on all materials that might affect the foundation.

The soil condition expert knowledge data (SCKB) file consists of a set of make commands with four descriptive factors for soil type, soil stiffness, negative skin friction, and presence of boulders. The entire SCKB file can be seen in Appendix E. The soil condition chart used in PS1 is presented in Figure 10.

Piles embedded in granular (cohessionless) soils, i.e., gravels, sands, and non-cohesive silts, derive their load-carrying capacity from both toe and shaft resistance. The relative contribution of each to the total capacity of the pile depends, essentially, on the density and shear strength of the soil and on the characteristics of the pile.

Piles in cohesive soils generally derive their capacity from shaft resistance. However, in very stiff clays or in cohesive tills, substantial toe resistance may be mobilized, which, for large diameter bored piles, may represent the usable capacity of the pile. The allowable loads on bored piles in cohesive soils are determined from a combination of shaft resistance and toe resistance.

In a structure highly sensitive to settlement, timber piles will be inappropriate but other types may be used. If adjacent structures are highly sensitive to noise, vibration, or ground heave then driven piles might be eliminated from selection. Artesian pressure can cause severe problems to bored, augered, pressure injected, and cast-in-place without mandrel piles due to collapse of the soil or shell wall.

Pile Length

The selection of the approximate pile length is made from a study of the soil profile and the strength and compressibility of each soil

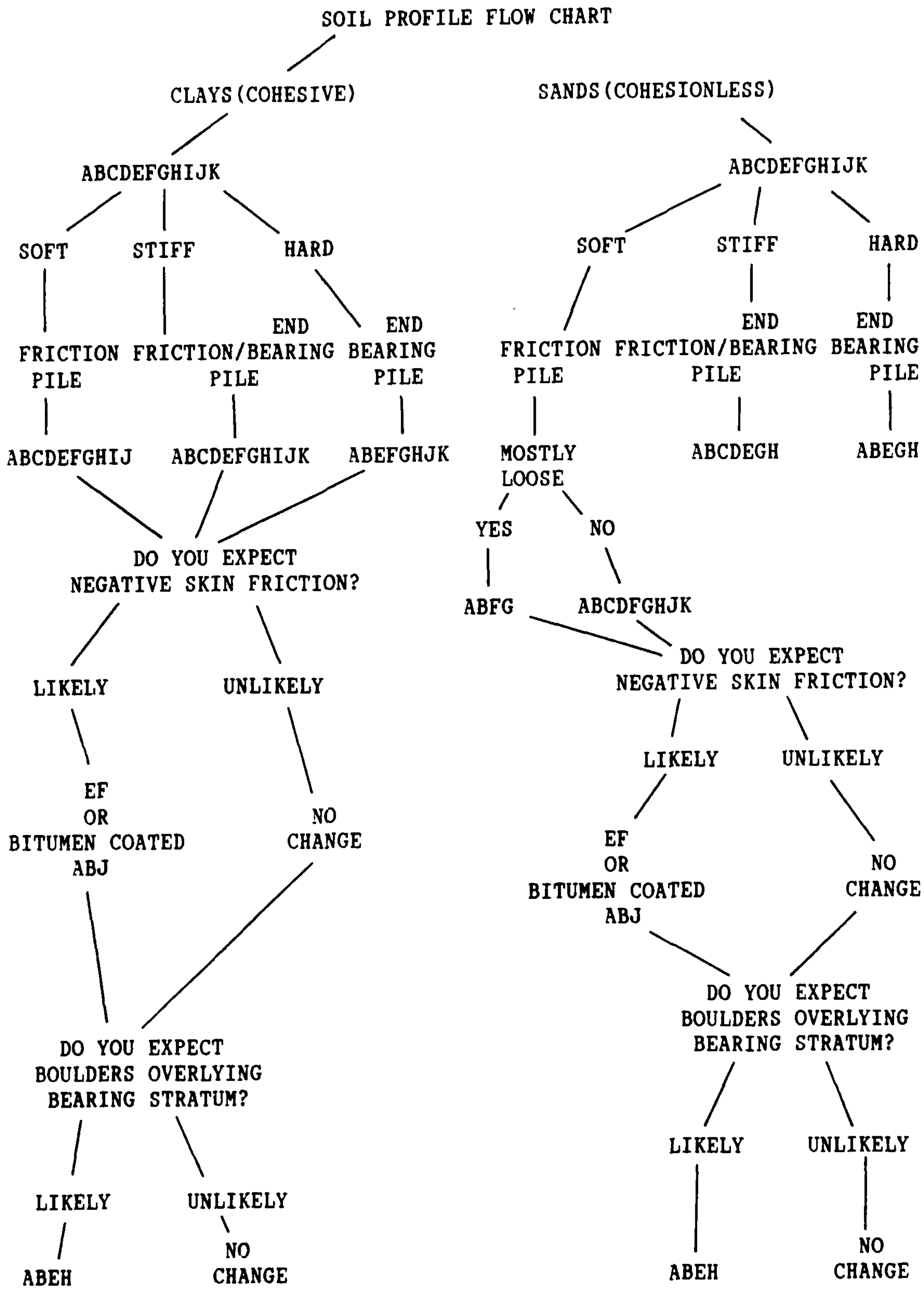


Figure 10. Soil conditions chart.

stratum. End-bearing piles must reach a stratum that is capable of supporting the entire foundation without undue settlement or failure, and friction piles must be long enough to distribute the stresses through the soil mass so as to minimize settlement and obtain adequate safety.

A wide variety of opinion was found on pile type verus pile length. The length criteria used in PS1 was developed by combining what was felt as the most accurate combinations from various authors.

It may be difficult for the user of PS1 to estimate the length required, since this is a fairly difficult task. If the user has no idea of the pile length, then the user may select the "no" option during pile length input. This option simply takes pile length criteria out of the pile selection process. There are occassions when a reasonable estimate can be made based on available knowledge.

The length expert knowledge data (LTHKB) file contains a set of make commands with lower and upper limits for pile lengths versus possible pile selections. The entire LTHKB file can be seen in Appendix F. The length chart used in PS1 is presented in Figure 11.

The ease or lack thereof for splicing a pile for added length was taken into consideration in determining the membership of long piles. No pile length shall be shorter than 10 feet (6).

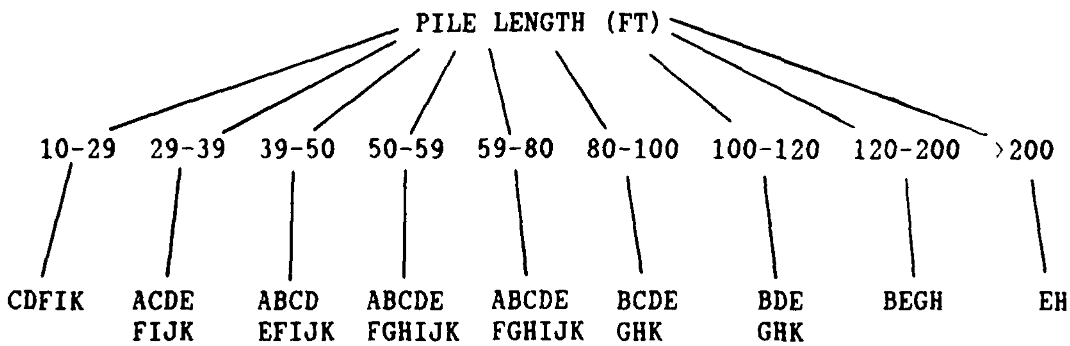


Figure 11. Pile length chart.

Availability of Materials

The availability of materials for certain pile types will vary depending on where in the world the pile construction site is located. The availability of material expert knowledge data (AOM) file consists of a set of make commands with three criteria: economically available, available, and not available. Economically available is considered as material or equipment in which transportation costs are not a factor to the builder. Available is considered as not economic, but still a possibility. Not available can't be resonably attained. The AOM flow chart is the same for every pile type. The membership or confidence (CF) is shown in Figure 12. The entire AOM data file used in PS1 can be seen in Appendix C. FLOPS will generate numerous rules from this AOM data file and can be seen in Appendix G.

AVAILABILITY OF
<PILE TYPE>

ECONOMICALLY AVAILABLE CF=1000

AVAILABLE CF=800

NOT AVAILABLE CF=0.0

Figure 12. Availability of material chart.

Local Experience

The choice of the type of pile, pile design capacity, and installation procedures is highly dependent on local experience and practice. A design engineer unfamiliar with these local practices should contact local building/engineering departments, local foundation contractors, and/or local foundation consultants.

The local experience expert data (LE) file consists of a set of make commands with three criteria of expert, some experience, and no experience. The LE chart is the same for every pile type. The membership or confidence (CF) is shown in Figure 13. The entire LE data file used in PS1 can be seen in Appendix D. FLOPS will generate numerous rules from this LE data file and can be seen in Appendix H.

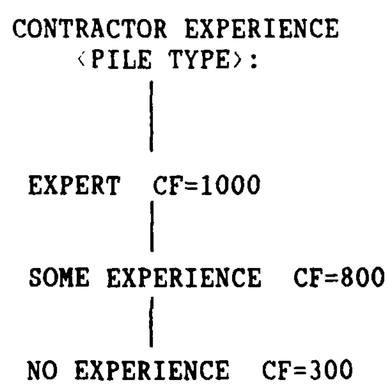


Figure 13. Local contractor experience chart.

Nuisance Effects

Nuisance effects account for both noise and vibration effects. Pile driving vibration may cause unwanted settlement of nearby structures or failures of slopes.

The noise and vibration expert knowledge data (NEKB, VEKB respectively) consists of a set of make commands with three constraints; no constraint, constraint, and extreme constraint. The classification "constraint" is considered as a public nuisance. The entire NEKB and VEKB data file can be seen in Appendix I and J respectively. The charts for both noise and vibration constraint can be seen in Figure 14 and Figure 15.

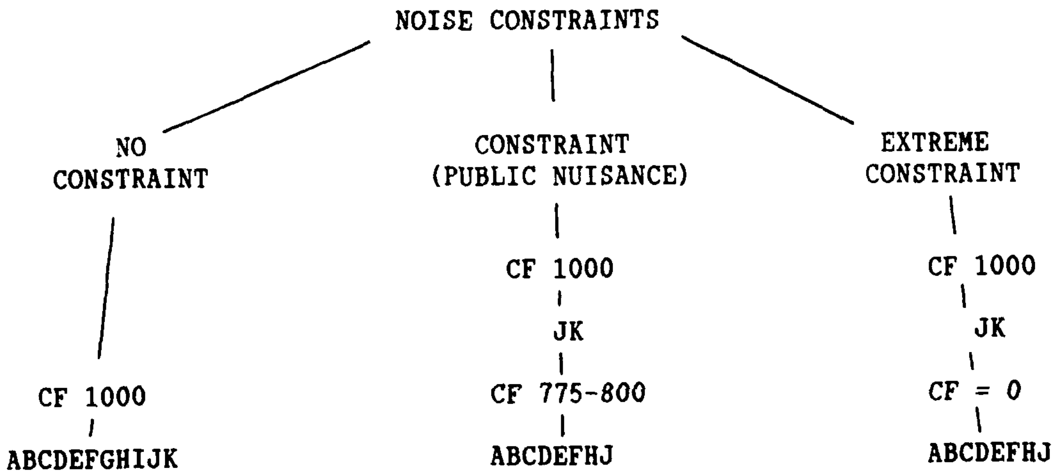


Figure 14. Noise constraint chart.

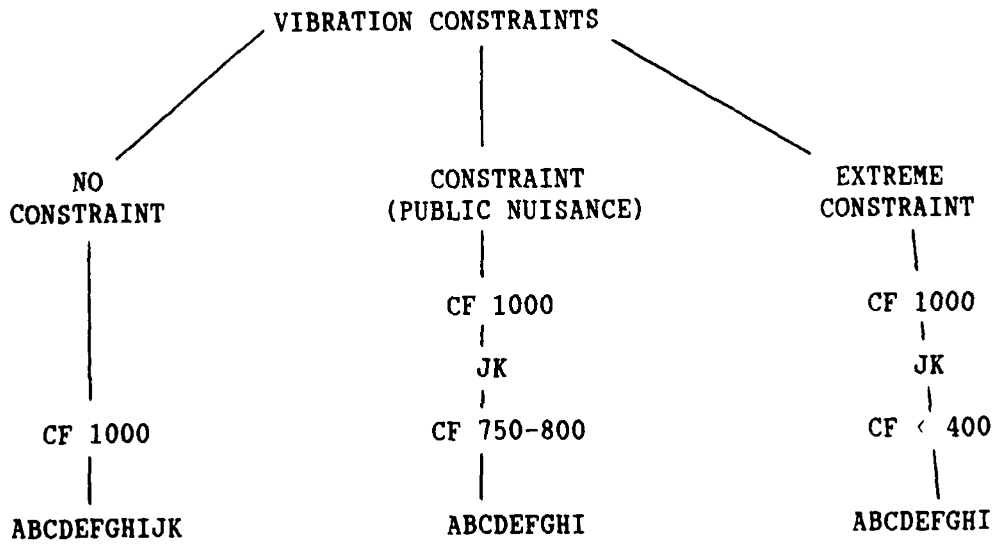


Figure 15. Vibration constraint chart.

CHAPTER IV

EXAMPLES OF PILE SELECTION USING PS1

Introduction

In this chapter, a few example problems are solved using PS1 to show its versatility in pile selection. The ability of the designer to weight pile selection criteria to meet local conditions will be demonstrated. These will then be discussed. The entire input and output for the following problems can be seen in appendices K, L, and M. In Appendix K, all the files of input and output of the process of solving problem 1 can be seen.

It is noted that because of the limits of computer memory, in the expert knowledge data files, most pile types with a confidence level under 500/1000, were discarded. This may somewhat reduce all recommended pile choice confidence levels. Regardless, the recommended top pile choices would not have significantly changed, but the actual confidence levels would be "slightly" higher. It is noted that such an effect is very minor. Therefore the confidence levels are not a measure of the believed ability of the pile to perform in a certain situation, but rather a means to distinguish between recommended pile types.

Problems 1 and 1A

The soil profile shows a fourty-five foot layer of soft clay overlaying a deep layer of stiff clay. There are no boulders in the area and it is felt that negative skin friction is unlikely to occur. The design load per pile is estimated to be 70 tons. The length of the pile is estimated to be 70 feet. There are no noise constraints, but a vibration constraint is present. There is a known constraint on the availability of material (AOM) for the casings of cast-in-place piles driven without a mandrel. There also appears to be constraints on the AOM of composite piles and pressure injected concrete pile equipment. The local contractor is an expert in all types of piling construction except for pressure injected piling where he/she has only some experience. For this problem, all criteria have equal weight of extremely important (1000/1000).

Using these values, the top three PS1 outputs predict steel piles, with a confidence of 93, followed by prestressed concrete piles, with a confidence of 92 and cast-in-place concrete piles, with a confidence of 90 as the top three possibilities. Detailed inputs and outputs are listed in Appendix K.

Using the same site conditions as in Problem 1 except that the weighting of criteria is changed. The vibration effects are weighted as extremely important (1000/1000), local contractor experience and soil conditions are weighted as important (800/1000), noise effects and availability of material are weighted as not very important (500/1000), and length and design loads are weighted as relatively unimportant (200/1000).

Using these new values, the top three PS1 outputs predict auger placed concrete pile, with a confidence of 90, steel piles, with a confidence of 90, and prestressed concrete piles, with a confidence of 88 as the top three possibilities. Detailed inputs and outputs are listed in Appendix K.

The site conditions for problems 1 and 1A are conditions for which many pile types are suitable. This leads to a number of closely grouped pile choices. Still the user has three top choices to work with.

The ability for the user to weight the site conditions to local conditions, increases PS1's capabilities. By changing the weighting factors in problem 1A, auger placed concrete piles have become a top choice, where as in problem 1, it was the sixth choice. In problem 1A, cast-in-place concrete is no longer a top choice.

Problems 2 and 2A

The soil profile shows a deep layer of soft clay. There are boulders in the area and it's strongly believed that negative skin friction is likely. The design load per pile is estimated to be 100 tons. The length of the pile is estimated to be 120 feet. There are noise constraints and extreme vibration constraints. There appears to be a constraint on the AOM of precast concrete and prestressed concrete piles. The local contractor is an expert in all types of piling construction. For this problem, all criteria have equal weight of extremely important (1000/1000).

Using these values, the top three PS1 outputs predict bored piles, with a confidence of 77, prestressed concrete piles, with a confidence of 73, and cast-in-place piles, with a confidence of 68 as the top three possibilities. Detailed inputs and outputs are listed in Appendix L.

Using the same site conditions as in Problem 2 except that the weighting of criteria is changed. The design load is weighted as extremely important (1000/1000), length and soil conditions are weighted as important (800/1000), noise effects and local contractor experience are weighted as not very important (500/1000), and vibration effects and availability of material are weighted as relatively unimportant (200/1000).

Using these values, the top three PS1 outputs predict prestressed concrete piles, with a confidence of 82, steel piles, with a confidence of 72, and cast-in-place piles, with a confidence of 71 as the top three possibilities. Detailed inputs and outputs are listed in Appendix L.

Due to the different constraints, the pile choices are spread out more than in problems 1 and 1A. Changing the weighting factors in problem 2A, knocks bored piles out of the top three and moves prestressed concrete piles into the top choice along with steel piles as the second choice.

Problems 3 and 3A

The soil profile shows a twenty-five foot layer of soft sand overlaying a deep layer of stiff sand. There are no boulders in the area and it is felt that negative skin friction is unlikely. The design load per pile is estimated to be 25 tons. The length of the pile is estimated to be 50 feet. There is no noise constraint and a vibration constraint is present. Precast concrete piles are not available and steel piles are available. The local contractor is an expert in most types of piling construction except for auger placed concrete piling where he/she has some experience, while for composite wood piling and pressure injected piling, he/she has no experience. For this problem, all criteria have equal weight of extremely important (1000/1000).

Using these values, the top three PS1 outputs predict timber piles, with a confidence of 95, cast-in-place pile with mandrel, with a confidence of 92, and auger placed concrete pile, with a confidence of 84 as the top three possibilities. Detailed inputs and outputs are listed in Appendix M.

Using the same site conditions as in Problem 3 except that the weighting of criteria is changed. The vibration effects are weighted as extremely important (1000/1000), availability of material and local contractor experience are weighted as important (800/1000), length and noise effects are weighted as not very important (500/1000), and design load and soil conditions are weighted as relatively unimportant (200/1000).

Using these values, the top three PS1 outputs predict auger placed concrete piles, with a confidence of 94, timber piles, with a confidence of 94, and bored piles with a confidence of 93 as the top three possibilities. Detailed inputs and outputs are listed in Appendix M.

The weighting factors in problem 3A tended to tighten the spread of pile choices and dropped cast-in-place out of the top three, but kept it a very close fourth. Auger placed and timber piles remain strong choices, even though the contractor has limited experience in auger placed piles, while bored piles moves into the top three.

CHAPTER V

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The major portion of this thesis was dedicated to collating scattered pile information and providing a data base for PS1. A comprehensive review of the literature has been made. This search showed that difference of opinions among the experts is common and that there are numerous uncertainties involved in soil engineering. It can take many years for an engineer to establish a strong feeling of proficiency.

Of all criteria, soil parameters play the greatest role in determining the type of pile required. Because of this, soil exploration was a key process in pile design. Various bearing capacity methods were discussed and recommendations provided. Potential soil-pile interaction problems were discussed along with their effects on the numerous pile choices. General information, strengths and weaknesses of each pile type were discussed in detail.

There might be three of four "right" types of pile for a given set of circumstances. Probably one of them is optimum considering loading, soil, costs, space, and environment. It could be difficult choosing those optimum pile choices for pile design and comparison unless the chooser has deep experience to rely on. PS1 was developed to assist the engineer and reduce the iterative process of finding that optimum pile. PS1 is an expert pile selection system using fuzzy logic. With a general knowledge input of the site conditions, PS1 can provide optimum pile choices for further design analysis and cost comparisons. This is the goal of the geotechnical engineer, to find that optimum solution. Fuzzy logic was used because it best modeled the uncertainties and overlapping abilities of various pile types. The expert knowledge data bases were presented and discussed.

The versatility of PS1 was shown through six pile problem sets. The ability of the user to weight criteria to meet his site conditions proved quite useful and showed comparitively substantial changes in pile selection output.

Conclusions

The conclusions reached in this study are:

1. Because of the numerous uncertainties and an industry based largely on engineering experience, a need for an expert system to assist in pile selection is evident.

2. Fuzzy set logic provided a better model of the real world for the PS1 expert system than traditional set logic.

3. PS1 is versatile and useful in selecting initial pile types to be used in further design analysis and cost comparisons. The ability to weight the pile selection criteria by the user, further refines the pile selection model to meet varying local requirements.

4. Because the result of PS1 is only as good as the built-in expert knowledge data base, the current version of PS1 is considered as a prototype expert system. Calibration of PS1 by experts in pile foundations, are required to bring PS1 to the status of a "true" expert system.

Recommendations for Further Research

Improvements to PS1 are recommended which could make it marketable. First, FLOPS needs to be expanded, if possible, so that more than 500 rules can be generated. To encompass all aspects of pile selection, it is estimated that FLOPS be altered to generate at least one thousand rules. This will allow for membership values in the expert data files to range from zero to one thousand, instead of either zero or five hundred to one thousand as currently exists. This will further refine PS1 and more accurately model the pile selection process. Second, further research analyses on the effects of pile selection in the areas of artesian pressure, uplifting forces, wider ranges of soil classifications and attributes, time constraints, and space constraints should be analyzed. Then additional expert knowledge data files can be created for use in PS1. External programs to estimate the pile length required and perform cost comparisons of pile types would round out PS1 into a complete package.

APPENDICES

Appendix A

PS1.c Program List

```
;*****************************************************************
: program PS1    Pile Selector                                   *
:   written      Juang & Ulshafer                                *
:   date         3-20-89                                         *
:   version      1.0                                             *
:   references   Siler and Tucker, "FLOPS" manual                *
;*****************************************************************

cls;
write 7 '\n\n\n' ;
write 7 '                  WELCOME TO PS1                        \n\n';
write 7 'Program PS1 is an expert system written with FLOPS, an      \n';
write 7 'expert system shell, for selection of an appropriate pile   \n';
write 7 'type.  The program uses a blackboard system for input data, \n';
write 7 'intermediate data and final output.  Production rules are   \n';
write 7 'written in FLOPS with automated generation scheme.  External\n';
write 7 'routines written in C are linked to the FLOPS program for   \n';
write 7 'faster computation. \n\n';

debug 0 ;
write 7 'Ready to compile the program  PS1 \n' ;
write 7 '\nCompiling . . .\n' ;

: DECLARATIONS

:control for user-supplied data
literalize start
    use_old_file atm
    store_data   atm
    check_data   atm
    modify_data  atm ;

:user- supplied (problem-specific) data
literalize xdata
   design_load     flt       pile_length        flt     soil_type   atm
   stiffness       atm       negative_friction  atm     boulder     atm
   noise_effect    atm       vibration_effect   atm
   avail_PC        atm       exp_PC             atm
   avail_PSC       atm       exp_PSC            atm
   avail_CIPM      atm       exp_CIPM           atm
   avail_CIP       atm       exp_CIP            atm
   avail_STL       atm       exp_STL            atm
   avail_TM        atm       exp_TM             atm
   avail_CPW       atm       exp_CPW            atm
   avail_CPS       atm       exp_CPS            atm
   avail_PIC       atm       exp_PIC            atm
   avail_APC       atm       exp_APC            atm
   avail_BP        atm       exp_BP             atm ;
```

```
:knowledge base data
literalize DL                          :design load knowledge base (DLKB)
   lower1       flt                      :lower bound of design load
   upper1       flt                      :upper bound
   fsmember     atm                      :member of fuzzt set "pile-type"
   confidence   flt ;                    :confidence of a pile being in

:  the class of fsmember
:knowledge base data
literalize LTH                         :pile length KB (LTHKB)
   lower2       flt
   upper2       flt
   fsmember     atm
   confidence   flt ;

literalize NEX                           :NEKB
   parameter1   atm                      :noise effects (NE)
   fsmember     atm
   confidence   flt ;

literalize VEX                           :VEKB
   parameter2   atm                      :vibration effects (VE)
   fsmember     atm
   confidence   flt ;

:knowledge base data
literalize AOMX                          :AOMKB
   parameter3   atm                      :availability of material (AOM)
   confidence   flt ;

literalize LEX                           :LEKB
   parameter4   atm                      :local exp (LE)
   confidence   flt ;

:knowledge base data
literalize SOIL                          :soil condition knowledge base (SCKB)
   factor1      atm                      : soil type
   factor2      atm                      : stiffness
   factor3      atm                      : negative_friction
   factor4      atm                      : boulders
   fsmember     atm
   confidence   flt ;
```

```
:pile-type selection according to each criterion
literalize pile_type
   criterion1   fzset                  :selection based on design load

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP )

   criterion2   fzset                  :selection based on pile length

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP )

   criterion3   fzset                  :selection based on soil parameters

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP )

   criterion4   fzset                  :selection based on noise effect

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP )

   criterion5   fzset                  :selection based on vibration effect

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP )

   criterion6   fzset                  :selection based on availability

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP )

   criterion7   fzset                  :selection based on experience

( PC PSC CIPM CIP STL TM CPW CPS PIC APC BP ) ;

:final selection
literalize final ;

:block firing control
literalize control
    block int ;

:**************************BLOCKS OF RULES*********************************

debug 0;

:RULES
:block 0 -- to generate blocks  #2, #3, #4, #5, #6, #7, and #8 rules

:rule r0 -- to generate block 2 rules
rule 1000 ( DL  ^lower1 = <LB>  ^upper1 = <UB>  ^fsmember = <FSM>
```

```
^confidence = <CONF> )
      -->
      rule <CONF> 2 ( xdata  ^design_load > <LB>  ^design_load <= <UB> )

    ( pile_type ^criterion1.<FSM> = 0)

    -->

    modify 2  ^criterion1.<FSM> ;

:rule r1 -- to generate block 3 rules
rule 1000 ( LTH  ^lower2 = <LB>  ^upper2 = <UB>  ^fsmember = <FSM>

 ^confidence = <CONF> )
      -->
      rule <CONF> 3 ( xdata  ^pile_length > <LB>  ^pile_length <= <UB> )

    ( pile_type ^criterion2.<FSM> = 0)

    -->

    modify 2  ^criterion2.<FSM> ;

:rule r2 (to generate block #4 rules)
rule 1000 ( SOIL  ^factor1 = <TYPE>  ^factor2 = <ST>

  ^factor3 = <NF>    ^factor4 = <BD>

  ^fsmember = <FSM>  ^confidence = <CONF> )
      -->
      rule <CONF> 4 ( xdata ^soil_type "<TYPE>"  ^stiffness "<ST>"

    ^negative_friction "<NF>"  ^boulder "<BD>" )

    ( pile_type  ^criterion3.<FSM> = 0 )

    -->

    modify 2  ^criterion3.<FSM> ;

:rule r3 -- to generate block #5 rules
rule 1000 ( NEX  ^parameter1 = <NE>  ^fsmember = <FSM>

 ^confidence = <CONF> )
      -->
      rule <CONF> 5 ( xdata  ^noise_effect "<NE>" )

    ( pile_type  ^criterion4.<FSM> = 0 )

    -->

    modify 2  ^criterion4.<FSM> ;
```

```
:rule r4 -- to generate block #6 rules
rule 1000 ( VEX  ^parameter2 = <VE>  ^fsmember = <FSM>
               ^confidence = <CONF> )
      -->
      rule <CONF> 6 ( xdata `vibration_effect "<VE>" )
                    (pile_type  ^criterion5.<FSM> = 0 )
           -->
           modify 2  ^criterion5.<FSM> ;

:rule r5- r15   -- to generate block #7 rules
:rule r5
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_PC "<AOM>" )
                    ( pile_type  ^criterion6.PC = 0 )
           -->
           modify 2  ^criterion6.PC ;

:rule r6
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_PSC "<AOM>" )
                    ( pile_type  ^criterion6.PSC = 0 )
           -->
           modify 2  ^criterion6.PSC ;

:rule r7
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_CIPM "<AOM>" )
                    ( pile_type  ^criterion6.CIPM = 0 )
           -->
           modify 2  ^criterion6.CIPM ;

:rule r8
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_CIP "<AOM>" )
                    ( pile_type  ^criterion6.CIP = 0 )
           -->
           modify 2  ^criterion6.CIP ;

:rule r9
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_STL "<AOM>" )
                    ( pile_type  ^criterion6.STL = 0 )
           -->
           modify 2  ^criterion6.STL ;
```

```
:rule r10
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_TM "<AOM>" )
                    ( pile_type  ^criterion6.TM = 0 )
           -->
           modify 2  ^criterion6.TM ;

:rule r11
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_CPW "<AOM>" )
                    ( pile_type  ^criterion6.CPW = 0 )
           -->
           modify 2  ^criterion6.CPW ;

:rule r12
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_CPS "<AOM>" )
                    ( pile_type  ^criterion6.CPS = 0 )
           -->
           modify 2  ^criterion6.CPS ;

:rule r13
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_PIC "<AOM>" )
                    ( pile_type  ^criterion6.PIC = 0 )
           -->
           modify 2  ^criterion6.PIC ;

:rule r14
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_APC "<AOM>" )
                    ( pile_type  ^criterion6.APC = 0 )
           -->
           modify 2  ^criterion6.APC ;

:rule r15
rule 1000 ( AOMX  ^parameter3 = <AOM>  ^confidence = <CONF> )
      -->
      rule <CONF> 7 ( xdata  ^avail_BP "<AOM>" )
                    ( pile_type  ^criterion6.BP = 0 )
           -->
           modify 2  ^criterion6.BP ;
```

```
:rule r16- r26   -- to generate block #8 rules
:rule r16
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_PC = "<LE>" )
                    (pile_type  ^criterion7.PC = 0 )
           -->
           modify 2  ^criterion7.PC ;

:rule r17
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_PSC = "<LE>" )
                    (pile_type  ^criterion7.PSC = 0 )
           -->
           modify 2  ^criterion7.PSC ;

:rule r18
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_CIPM = "<LE>" )
                    (pile_type  ^criterion7.CIPM = 0 )
           -->
           modify 2  ^criterion7.CIPM ;

:rule r19
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_CIP = "<LE>" )
                    (pile_type  ^criterion7.CIP = 0 )
           -->
           modify 2  ^criterion7.CIP ;

:rule r20
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_STL = "<LE>" )
                    (pile_type  ^criterion7.STL = 0 )
           -->
           modify 2  ^criterion7.STL ;

:rule r21
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_TM = "<LE>" )
                    (pile_type  ^criterion7.TM = 0 )
           -->
           modify 2  ^criterion7.TM ;
```

```
:rule r22
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_CPW = "<LE>" )
                    (pile_type  ^criterion7.CPW = 0 )
           -->
           modify 2  ^criterion7.CPW ;

:rule r23
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_CPS = "<LE>" )
                    (pile_type  ^criterion7.CPS = 0 )
           -->
           modify 2  ^criterion7.CPS ;

:rule r24
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_PIC = "<LE>" )
                    (pile_type  ^criterion7.PIC = 0 )
           -->
           modify 2  ^criterion7.PIC ;

:rule r25
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_APC = "<LE>" )
                    (pile_type  ^criterion7.APC = 0 )
           -->
           modify 2  ^criterion7.APC ;

:rule r26
rule 1000 ( LEX  ^parameter4 = <LE>   ^confidence = <CONF> )
      -->
      rule <CONF> 8 ( xdata ^exp_BP = "<LE>" )
                    (pile_type  ^criterion7.BP = 0 )
           -->
           modify 2  ^criterion7.BP ;
```

```
:-------------------------------------------------------------------
:block 1  for gathering problem-specific data from user
:rule r27

rule 1000 1 (start)
      -->
      write '\n*******************************************************\n',
      write '   Begin to gather problem-specific data from user.\n',
      write '******************************************************\n',
      call get_data ,
      transfer xdata from user.dat ,
      write '\nUser-supplied data has been loaded to FLOPS.\n',
      write '************** END OF USER INPUT ****************\n',
      write '\nReady to run all generated rules on users data. \n',
      call pause ,
      write '\nExecuting . . . \n\n',
      make pile_type ;


:---------------------------------------------------------------------------
:blocks 2-8 -->to run rules generated for each of the seven CRITERIA
:                considered in the selection of pile type.
:---------------------------------------------------------------------------


:block 9 (rule r28) is to determine final selection
:rule r28

rule 1000 9 (final)
      -->
      call compute ;


:---------------------------------------------------------------------------


:block 99 -- always enabled
:rule r29 -- to control block firing

rule 1000 99 ( control ^block <N>  ^block <= 9)
      -->
      fire block <N> del ,
      modify 1 ^block ( <N> + 1 ) ,
      fire block ( <N> + 1 ) on ;
```

```
:**************************END OF BLOCKS AND RULES***************************

: INPUT DATA, INITIATION, AND ACTIONS

write 7 '\nAll original rules compiled, ready to generate new rules. \n';
write 7 '\nReading expert data file DLKB.c (Design Load) . . .\n';
open DLKB.c ;
write 7 '\nReading expert data file LTHKB.c (Length) . . .\n';
open LTHKB.c ;
write 7 '\nReading expert data file SCKB.c (Soil Conditions) . . .\n';
open SCKB.c ;
write 7 '\nReading expert data file NEKB.c (Noise Effects) . . .\n';
open NEKB.c ;
write 7 '\nReading expert data file VEKB.c (Vibration Effects) . . .\n';
open VEKB.c ;
write 7 '\nReading expert data file AOMKB.c (Avail of Material) . . .\n';
open AOMKB.c ;
write 7 '\nReading expert data file LEKB.c (Local Experience) . . .\n';
open LEKB.c ;

:turn all rules off
fire all off ;

:enable block firing control
make control ^block 0 ;

:run block 0 to generate rules
write 7 '\nNow FLOPS is ready to construct new rules. \n';
write 7 '\nConstructing new rules . . . \n';
write 7 '(Please wait for a moment.)\n';
fire block 0 on ;
fire block 99 on ;
debug 0 ;
run 1 ;                     :run block 0 to generate 7 blocks of rules
                            :and run block 99 to turn off block 0 and
                            :turn on block 1
:prule;                     :FOR DEBUGGING ONLY (PRINT ALL RULES)

write 7 '\nNew rules have been generated. \n';
write 7 '\nReady to gather problem-specific data from user \n\n';
call pause ;

:turn off generated rules except block 1 and 99
fire all off ;
fire block 1 on ;
fire block 99 on ;
```

```
:to gather problem-specific data from the user
:by firing block-1 rules (r27). Block #1 also make an instance
:of "pile_type" so that block#2 thru #8 will be firable.


make start;
run 1 ;        :to run block #1

:Next to run blocks #2 through #8
debug 1;
run 1;
run 1;
run 1;
run 1;
run 1;
run 1;
run 1;

transfer pile_type to pile_type.dat ;

make final;
run 1;                                  :to run last block (#9)

write 7 '\nAt the DOS prompt enter  TYPE OUTPUT.PS1 for more\n';
write 7 'information, if desired.\n';
stop ;
```

Appendix B

Program Get_Data.c

```
/*******************************************************************/
/* Program get_data.c -- called by FLOPS program pile.c            */
/*                    -- to get data from user                     */
/*******************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define MESSAGE1 "Estimated or required design load per pile (tons) =  "
#define MESSAGE2 "Estimated or required pile length (ft) =  "
#define MESSAGE3 "Must know if it's cohesive (clay) or cohesionless (sand)."

/* key in '\' and <Enter> for continuation of string in #define statement */

#define MESSAGE4 \
"Need to know how pile develops resistance: \
friction, end bearing, or both."

#define MESSAGE5 \
"Lowering of water or new fill on ground surface may cause development \
of negative skin friction."

#define MESSAGE6 \
"Presence of boulders affects pile driving and \
selection of pile type."

#define MESSAGE7 \
"For some areas noise arisen from pile installation \
may be unacceptable."

#define MESSAGE8 \
"For some areas vibration arisen form pile installation \
may be unacceptable."

#define MESSAGE9 \
"Need to know if a particular type of pile material \
is available."

#define MESSAGE10 \
"Need to know if local experience on selected type \
of pile is secured."
```

```
#define YES 'y'
#define NO 'n'

/* Function declarations */
int   get_a_response ();
void  data ();
void  action1 (char *message);
void  action2 (int N, int *choice, char *message);

/* Other declarations */
char  parameter[BUFSIZ], stringL[BUFSIZ], dummy[80];
char  *answer, *slevel, *result;
FILE  *fp_in, *fp_out ;

main ()
{
   int ch;

   printf ("\nUse previously-created data file (\"user.dat\") ? (y/n)\n");
   ch = get_a_response();
   if ( ch == NO)
      data ();

X1:   printf ("\nWant to verify data ? (y/n)\n");
      ch = get_a_response();
      if (ch == YES)

 {

 if ((fp_in = fopen("user.dat", "r")) == NULL)

    printf ("Can't open the file \"user.dat\".\n");

 else

    while ((ch = getc(fp_in)) != EOF)

       putc (ch, stdout);


 printf ("\n\nWant to modify data ? (y/n)\n");

 ch = get_a_response();

 if (ch == YES)

    {

    data ();
```

```
    goto X1;

    }

 }
      printf ("\nData updated and stored !\n");
}

/* Function definition */
int get_a_response ()
{
      int ch;

X2:   ch = getchar ();
      gets (dummy);
      ch = tolower (ch);   /* change to lower case letter if any */
      if (ch != YES  && ch != NO)

 {

 printf ("\nMust enter  y or n. Try again.  ");

 goto X2;

 }
      return (ch);
}

/* Function definition */
void data ()
{
   int choice;
   int index;
   char *pile_type [] = {


   "PRECAST CONCRETE PILE",


   "PRESTRESSED CONCRETE PILE",


   "CAST-IN-PLACE PILE WITH MANDREL",
```

```
  "CAST-IN-PLACE PILE WITHOUT MANDREL",

  "STEEL PILE",

  "TIMBER PILE",

  "COMPOSITE PILE (WOOD-CONCRETE)",

  "COMPOSITE PILE (STEEL-CONCRETE)",

  "PRESSURE INJECTED CONCRETE PILE",

  "AUGER INJECTED CONCRETE PILE",

  "BORED PILE"

 };

  if ((fp_out = fopen("user.dat", "w")) == NULL)
     printf ("Can't open the file \"user.dat\".\n");
  else
    {
     rewind (fp_out);
     fputs ("\"xdata\",", fp_out);   /* "xdata" is requied by FLOPS */
     printf ("\n\nANSWER ALL QUESTIONS ASKED ...\n");

     printf ("\nIs data on DESIGN LOAD PER PILE\n");
     prir*' ("known or can be estimated? (y/n)  ");

action1 (MESSAGE1);

     printf ("\nIs data on PILE LENGTH known or can be estimated? (y/n)\n");

action1 (MESSAGE2);
```

```
     printf ("\nWhich one of the descriptions is right on SUBSURFACE SOIL\n");

     printf ("TYPE? (Enter 1, 2, or 3):\n");
     printf ("  1) mostly cohesive, 2) mostly cohesionless, 3) why?\n");

action2 (3, &choice, MESSAGE3);

     switch (choice)

{

case 1 :

     fputs ("\"COHESIVE\",", fp_out);

     fputs (result, fp_out);

     break;

case 2 :

     fputs ("\"COHESIONLESS\",", fp_out);

     fputs (result, fp_out);

     break;

}

     printf ("\nWhich one of the descriptions is right on SUBSURFACE\n");
     printf ("SOIL CONDITION ? (Enter 1, 2, 3, or 4):\n");
     printf ("  1) very deep soft layer, 2) soft layer underlain by ");
     printf ("stiff layer\n  3) soft layer underlain by hard stratum ");
     printf ("4) why?\n");

action2 (4, &choice, MESSAGE4);

     switch (choice)

{

case 1 :

     fputs ("\"soft\",", fp_out);

     fputs (result, fp_out);

     break;

case 2 :
```

```
fputs ("\"stiff\",", fp_out);

fputs (result, fp_out);

break;

case 3 :

fputs ("\"hard\",", fp_out);

fputs (result, fp_out);

break;

}

printf ("\nWhich one of the descriptions is right on DEVELOPMENT\n");
printf ("OF NEGATIVE SKIN FRICTION ? (Enter 1, 2, or 3):\n");
printf ("  1) likely, 2) unlikely, 3) why?\n");

action2 (3, &choice, MESSAGE5);

switch (choice)

{

case 1 :

fputs ("\"likely\",", fp_out);

fputs (result, fp_out);

break;

case 2 :

fputs ("\"unlikely\",", fp_out);

fputs (result, fp_out);

break;

}

printf ("\nWhich one of the descriptions is right on PRESENCE ");
printf ("OF BOULDERS IN THE \nRANGE OF PILE EMBEDMENT? ");
printf ("(Enter 1, 2, or 3):\n");
printf ("  1) present, 2) not present, 3) why? \n");
```

```
action2 (3, &choice, MESSAGE6);

     switch (choice)

{

case 1 :

     fputs ("\"yes\",", fp_out);

     fputs (result, fp_out);

     break;

case 2 :

     fputs ("\"no\",", fp_out);

     fputs (result, fp_out);

     break;

}

     printf ("\nWhich one of the descriptions is right on TOLERANCE\n");
     printf ("OF NOISE EFFECTS ? (Enter 1, 2, 3, or 4):\n");
     printf ("  1) extreme constraint, 2) constraint, 3) no constraint, ");
     printf ("4) why?\n");

action2 (4, &choice, MESSAGE7);

     switch (choice)

{

case 1 :

     fputs ("\"extreme_constraint\",", fp_out);

     fputs (result, fp_out);

     break;

case 2 :

     fputs ("\"constraint\",", fp_out);

     fputs (result, fp_out);
```

```
        break;

case 3 :

        fputs ("\"no_constraint\",", fp_out);

        fputs (result, fp_out);

}

        printf ("\nWhich one of the descriptions is right on TOLERANCE\n");
        printf ("OF VIBRATION EFFECTS ? (Enter 1, 2, 3, or 4):\n");
        printf ("  1) extreme constraint, 2) constraint, 3) no constraint, ");
        printf ("4) why?\n");

action2 (4, &choice, MESSAGE8);

        switch (choice)

{

case 1 :

        fputs ("\"extreme_constraint\",", fp_out);

        fputs (result, fp_out);

        break;

case 2 :

        fputs ("\"constraint\",", fp_out);

        fputs (result, fp_out);

        break;

case 3 :

        fputs ("\"no_constraint\",", fp_out);

        fputs (result, fp_out);

        break;

}

        for (index= 0; index <= 10; index++)
        {
```

```
     printf ("\nWhich one of the descriptions is right on MATERIAL");
     printf (" AVAILABILITY OF\n%s", pile_type[index] , " ?");
     printf (" (Enter 1, 2, 3, or 4):\n");
     printf ("  1) economically available, 2) available, 3) not ");
     printf ("available, 4) why?\n");

action2 (4, &choice, MESSAGE9);

     switch (choice)

{

case 1 :

     fputs ("\"economical\",", fp_out);

     fputs (result, fp_out);          /* "result" is conf. level */

     break;                           /* in the form of an ASCII */

case 2 :                              /* string */

     fputs ("\"available\",", fp_out);

     fputs (result, fp_out);

     break;

case 3 :

     fputs ("\"not_available\",", fp_out);

     fputs (result, fp_out);

     break;

}

     printf ("\nWhich one of the descriptions is right on LOCAL ");
     printf ("CONSTRUCTION EXPERIENCE OF\n%s", pile_type[index]," ? ");
     printf (" (Enter 1, 2, 3, or 4):\n");
     printf ("  1) expert, 2) some experience, 3) no experience ");
     printf ("4) why?\n");

action2 (4, &choice, MESSAGE10);

     switch (choice)

{
```

```
 case 1 :

      fputs ("\"expert\",", fp_out);

      fputs (result, fp_out);          /* "result" is conf. level */

      break;                           /* in the form of an ASCII */

 case 2 :                              /* string */

      fputs ("\"some_experience\",", fp_out);

      fputs (result, fp_out);

      break;

 case 3 :

      fputs ("\"no_experience\",", fp_out);

      fputs (result, fp_out);

      break;

 }

      } /* end of for-loop */

     } /* end of first if-else structure */

   fclose (fp_out);
}  /* end of function data */


/* Function Definition */
/* action1 -- to obtain quantitative data and to write to output file; */
/*            parameter and its conf. level are stored as ASCII string */

void action1 (char *message)
{
   int ch;

   ch = getchar ();
   gets (dummy);
   ch = tolower (ch);          /* change to lower case letter if any */
   if (ch != YES  &&  ch != NO)
      {
      printf ("\nMust enter  y or n. Try again.  ");
      action1 (message);
      }
```

```
   else if (ch == YES)
      {
      printf ("\n%s", message);
      answer = gets (parameter);
      fputs (answer, fp_out);
      fputs (",1000 ,", fp_out);
      }
   else
      fputs ("0,0, ", fp_out);
   return;
}

/* Function definition */
/* action2 --  to obtain and return "choice", and to assign value to  */
/*             global variable "result" which is an ASCII string of   */
/*             the confidence level  */

void action2 (int N, int *choice, char *message)



      /* Note that "*choice" is the value stored */
{                            /* in the address named "choice"         */
   int level;

   scanf ("%d", choice);      /* Note that "choice" is already an address */
   gets (dummy);
   if (*choice >= 1 && *choice <= N)
   {
      if (*choice != N)
      {
       printf ("\nWhat is the confidence of your answer on the above ");
       printf ("question ?\n");
       printf ("(Enter 1, 2, or 3):\n");
       printf ("   1) absolutely sure, 2) very sure, 3) sure\n");
       scanf ("%d", &level);
       gets (dummy);               /* preventing any unwanted string */



    /* from being assigned to next input */
       if (level == 1)  level = 1000;
       if (level == 2)  level = 900;
       if (level == 3)  level = 750;

       slevel = itoa (level, stringL, 10);  /* convert int. to string */
       result = strcat (stringL, ", ");     /* add to string a comma */
      }
```

```
    else
    {
     printf ("\n%s\n", message);
     printf ("\n****************************\nNow enter your choics: \n");
     action2 (N, choice, MESSAGE10);
    }
  }
  else
  {
   printf ("\nMust enter an integer between 1 and %d. Try again \n", N);
   action2 (N, choice, MESSAGE10);
  }
}
```

Appendix C

Program Compute.C

```
/***********************************************************************/
/* program COMPUTE.C - computation for final selection of pile        */
/***********************************************************************/
/* main ()              |  Called by FLOPS program ps1.c               */
/*                      |  No data transfered from ps1.c               */
/*                      |  But will read a file pile_type.dat created  */
/*                      |  By ps1.c                                    */
/*                      |  Then it creates output.ps1                  */
/*                      |  Before return to FLOPS                      */
/***********************************************************************/

#include <stdio.h>
#define LENGTH 5  /* maximum length of any field in a record */


  /* stored in the file pile_type.dat */
#define NUMBER 11 /* number of pile types */

#define MESSAGEX \
"If your answer is NO, the program uses the default setting, which \
applies\nequal weight to all criteria.  Now enter your choice.\n"

/* Function declaration */
void sort_control (char *type[], int final_conf[]);
void sort (char *type[], int final_conf[]);

char *type [] = {


     "PC - precast concrete pile",


     "PSC - prestressed concrete pile",


     "CIPM - cast-in-place with mandrel",


     "CIP - cast-in-place without mandrel',


     "STL - steel pile (H, I, or Pipe)",
```

```
 int ch ;
 int i, j, index, choice, choiceX, choiceY ;
 float sum1, sum2;
 char intarry[LENGTH] ;

 int conf [7][11] ;     /* conf is confidence level for each member  */


/* of a 11-member fuzzy set; 7 fuzzy sets    */


/* resulted from FLOPS execution of ps1.c    */


/* data are stored in pile_type.dat          */
 int final_conf [11]; /* final confidence on selection of each pile */


 /* Open file output.ps1 for writing */

if (( fp_pile = fopen("output.ps1", "w")) == NULL)
   {
   printf ("Can't open the file \"output.ps1\". \n");
   printf ("Program COMPUTE.C will stop.\n");
   goto XX;
   }
else
{       /* to write hardcopy outputs to file output.ps1 herein */

 /* Begin to read in pile_type.dat which results from FLOPS execution.  */
 /* pile_type.dat is a ASCII file with only one record containing 78    */
 /* fields. The first field is a string which is the name of an element */
 /* in FLOPS. The other fields are membership values in terms of conf.  */
 /* level for each member of the fuzzy sets "criterion1", "criterion2", */
 /* up to "criterion7". */

 if (( fp_type = fopen("pile_type.dat", "r")) == NULL)
    {
    printf ("Can't open the file \"pile_type.dat\". \n");
    printf ("Program COMPUTE.C will stop.\n");
    goto XX;
    }
 else
    {
    while ((ch = getc(fp_type)) != ',');  /* skip 1st field (a string) */
```

```
     fprintf (fp_pile, "KEY OUTPUT OF ps1.c FOR PILE SELECTION\n\n");
     fprintf (fp_pile, "I. Confidence for selecting each of the eleven ") ;
     fprintf (fp_pile,"piles\n   based on each of the seven criteria.\n\n");
     for (j=0; j <11; j++)

fprintf (fp_pile, "%s\n", type[j]);

     fprintf (fp_pile, "\n   PC   PSC  CIPM   CIP   STL    TM   CPW    ") ;
     fprintf (fp_pile, "CPS   PIC   APC    BP\n");


for (i = 0; i < 7; i++)

{

   for (j = 0; j < 11; j++)

   {                              /* collect characters in a field */

   index = 0 ;                   /* and store in an array intarry */

   while (((ch = getc(fp_type)) != ',')  &&  (index < LENGTH))

      intarry [index++] = ch ;

   intarry [index] = '\0' ;



 /* convert string to an integer */

   conf [i][j] = atoi (intarry) ;

   }

fprintf (fp_pile, "%5d %5d %5d %5d %5d %5d %5d %5d %5d %5d %5d\n",

 conf[i][0], conf[i][1], conf[i][2], conf[i][3],

 conf[i][4], conf[i][5], conf[i][6], conf[i][7],

 conf[i][8], conf[i][9], conf[i][10] );

}
     }
  fclose (fp_type) ;
```

```
   /* The following section is to collect user supplied weighting data */

   printf ("\n*********************************\n");
   printf ("\n  Need some more data from you!\n");
   printf ("\n*********************************\n");
   printf ("\nThe following are a list of pile selection criteria:\n\n");
     for (i=0; i < 7; i++)

 printf ("   %s\n", criteria[i]);

   printf ("\nDo you want to put weights to these criteria BY YOURSELF?\n");
   printf ("(Enter  1- yes,  2- no,  3- why? ):\n");
X:   scanf ("%d", &choiceX);

   if (choiceX != 1  &&  choiceX != 2  && choiceX != 3) {
      printf ("\nMust enter an integer 1 ,2, or 3. Try again!\n");
      goto  X; }

   if (choiceX == 1)
   {
      for (i=0; i < 7; i++)
      {
       printf ("\nOn a relative basis, assign a weight for the criterion\n");
       printf ("%s", criteria[i]);
       printf ("\nEnter 1- extremely important, 2- important, 3- not very");
       printf (" important\n      4- unimportant\n");
Y:     scanf ("%d", &choiceY);

if (choiceY != 1 && choiceY != 2 && choiceY != 3 && choiceY != 4) {

   printf ("\nMust enter an integer 1, 2, or 3. Try again!\n");

   goto  Y; }

if (choiceY == 1)  weight[i] = 1000 ;

if (choiceY == 2)  weight[i] =  800 ;

if (choiceY == 3)  weight[i] =  500 ;

if (choiceY == 4)  weight[i] =  200 ;
      }
   }

   if (choiceX == 3) {
      printf ("\n%s", MESSAGEX);
      goto X ; }

   printf ("\n\n\n****** NO MORE QUESTIONS ASKED *******\n");
```

```
  /* to write the weighting factors to file pointed by fp_pile */
     fprintf (fp_pile, "\n\nII. Weighting factors for the seven ");
     fprintf (fp_pile, "criteria:\n\n") ;

fprintf (fp_pile, "%8d\n%8d\n%8d\n%8d\n%8d\n%8d\n%8d\n",


 weight[0], weight[1], weight[2], weight[3],


 weight[4], weight[5], weight[6]);

fprintf (fp_pile,"\nThe seven criteria used, in the order of the");

fprintf (fp_pile,"\nabove listed weight are as follows:\n\n");

for (i=0; i < 7; i++)

    fprintf (fp_pile, "%s\n", criteria[i]);


  /* Begin to compute Weighted Confidence */

  for (j=0; j < 11; j++)
  {
    sum1 = 0 ;
    sum2 = 0 ;
    for (i=0; i < 7; i++)
    {
      sum1 += ((conf[i][j]/10) * (weight[i]/10)) ;
      sum2 += (weight[i] /10) ;
    }
    final_conf[j] = sum1 / sum2 ;
  }

    fprintf (fp_pile, "\n\nIII. Final Confidence for selecting a ");
    fprintf (fp_pile, "particular pile: \n");

    fprintf(fp_pile, "\npile:   PC   PSC  CIPM   CIP   STL    TM   CPW   ");
    fprintf(fp_pile, "CPS   PIC   APC    BP\n");
    fprintf(fp_pile, "conf:%5d %5d %5d %5d %5d %5d %5d %5d %5d %5d %5d\n",


 final_conf[0], final_conf[1], final_conf[2], final_conf[3],


 final_conf[4], final_conf[5], final_conf[6], final_conf[7],


 final_conf[8], final_conf[9], final_conf[10] );
```

```
   printf ("\n\n*******************************************");
   printf ("\nEND OF COMPUTATION!  END OF COMPUTATION!\n\n");
   printf ("Select an option to view the recommendation");
   printf (" (Enter 1, 2, or 3): \n");
   printf ("  1) all results, 2) top three choices, or 3) top choice\n");
Z:   scanf ("%d", &choice);

   printf ("\nRECOMMENDATION FROM ps1.c ON PILE SELECTION\n");
   if (choice >= 1  &&  choice <= 3)
   {
     switch (choice)
     {
     case 1 :
      printf ( "\n\npile:   PC   PSC  CIPM   CIP   STL    TM   CPW   ") ;
      printf ( "CPS   PIC   APC    BP\n");
      printf ( "conf:%5d %5d %5d %5d %5d %5d %5d %5d %5d %5d %5d\n",


  final_conf[0], final_conf[1], final_conf[2], final_conf[3],


  final_conf[4], final_conf[5], final_conf[6], final_conf[7],


  final_conf[8], final_conf[9], final_conf[10] );
      printf ("\n\nSymbols used above are as follows:\n\n");
      for (j=0; j <11; j++)

 printf ("%s\n", type[j]);
      break;

     case 2 :
      sort_control ( type, final_conf);
      for (j=0; j < 3; j++)

 printf ( "\n%s   --> confidence = %4d\n", type[j], final_conf[j]);
      break;

     case 3 :
      sort_control ( type, final_conf);
      printf ("\nThe best choice is  %s", type[0]);
      printf ("\nWith a confidence of %4d\n", final_conf[0]);
      fprintf (fp_pile, "\n\nThe best choice is  %s", type[0]);
      fprintf (fp_pile, "\nWith a confidence of %4d\n", final_conf[0]);
      break;
     }
   }

   else
   {
```

```
      printf ("\nMust enter an integer between 1 and 3. Try again!\n");
      goto  Z;
   }

     fprintf (fp_pile, "\n\n***** END OF output.ps1 *****\n");
     fclose (fp_pile) ;
 }

XX:   index= -1;
}


/* Function definition */
void sort_control (char *type[], int final_conf[])
{
   int n, j ;
   int newconf[NUMBER] ;
   char *newtype[NUMBER] ;

   n= 0;
   for (j=0; j < NUMBER; j++)
   {
     sort (type, final_conf);
     newtype [n] = type[0] ;
     newconf [n] = final_conf[0] ;
     final_conf[0] = 0 ;
     n += 1 ;
   }

   for (j=0; j < NUMBER; j++)
   {
     final_conf[j] = newconf[j] ;
     type[j] = newtype[j] ;
   }
}


/* Function definition */
void sort (char *type[], int final_conf[])
{
   int j, temp, count ;
   char *tempX ;
   int max ;

   max = final_conf[0] ;
   for (j=1; j < NUMBER; j++)
   {
    if (max < final_conf[j])
```

```
max = final_conf[j] ;
   }

   j = 0;
   while (max != final_conf[j]) j++ ;
   count = j ;

   tempX = type[count] ;
   type[count] = type[0] ;
   type[0] = tempX ;
   temp = final_conf[count] ;
   final_conf[count] = final_conf[0] ;
   final_conf[0] = temp ;
}
```

Appendix D

List of DLKB Expert Knowledge File

```
:Design loading (tons)
:Key:
:PC = Precast Concrete piles
:PSC = Prestressed Concrete piles
:CIPM = Cast-in-Place with Mandrel
:CIP = Cast-in-Place without Mandrel
:STL = Steel pile (H, I, of open pipe)
:TM = Timber pile
:CPW = Composite Pile Wood/concrete
:CPS = Composite Pile Steel/concrete
:     also Composite Pile Steel Pipe filled w/concrete & steel H-beam
:PIC = Pressure Injected Concrete
:APC = Auger Pressure injected Concrete
:BP = Bored/Cassion Piles


make DL ^lowerl 0 ^upperl 21 ^fsmember "CIPM" ^confidence 600;

make DL ^lowerl 0 ^upperl 21 ^fsmember "TM" ^confidence 1000;

make DL ^lowerl 21 ^upperl 29 ^fsmember "CIPM" ^confidence 800;

make DL ^lowerl 21 ^upperl 29 ^fsmember "TM" ^confidence 1000;

make DL ^lowerl 21 ^upperl 29 ^fsmember "CPW" ^confidence 700;

make DL ^lowerl 29 ^upperl 39 ^fsmember "PC" ^confidence 800;

make DL ^lowerl 29 ^upperl 39 ^fsmember "PSC" ^confidence 800;

make DL ^lowerl 29 ^upperl 39 ^fsmember "CIPM" ^confidence 1000;

make DL ^lowerl 29 ^upperl 39 ^fsmember "CIP" ^confidence 500;

make DL ^lowerl 29 ^upperl 39 ^fsmember "TM" ^confidence 700;

make DL ^lowerl 29 ^upperl 39 ^fsmember "CPW" ^confidence 1000;

make DL ^lowerl 29 ^upperl 39 ^fsmember "CPS" ^confidence 1000;

make DL ^lowerl 29 ^upperl 39 ^fsmember "APC" ^confidence 800;

make DL ^lowerl 39 ^upperl 50 ^fsmember "PC" ^confidence 1000;

make DL ^lowerl 39 ^upperl 50 ^fsmember "PSC" ^confidence 1000;

make DL ^lowerl 39 ^upperl 50 ^fsmember "CIPM" ^confidence 1000;

make DL ^lowerl 39 ^upperl 50 ^fsmember "CIP" ^confidence 800;
```

```
make DL ^lowerl 39 ^upperl 50 ^fsmember "STL" ^confidence 1000;
make DL ^lowerl 39 ^upperl 50 ^fsmember "TM" ^confidence 600;
make DL ^lowerl 39 ^upperl 50 ^fsmember "CPW" ^confidence 700;
make DL ^lowerl 39 ^upperl 50 ^fsmember "CPS" ^confidence 1000;
make DL ^lowerl 39 ^upperl 50 ^fsmember "PIC" ^confidence 900;
make DL ^lowerl 39 ^upperl 50 ^fsmember "APC" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "PC" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "CIPM" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "CIP" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "STL" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "CPS" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "PIC" ^confidence 1000;
make DL ^lowerl 50 ^upperl 59 ^fsmember "APC" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "PC" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "CIPM" ^confidence 800;
make DL ^lowerl 59 ^upperl 70 ^fsmember "CIP" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "STL" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "CPS" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "PIC" ^confidence 1000;
make DL ^lowerl 59 ^upperl 70 ^fsmember "APC" ^confidence 900;
make DL ^lowerl 70 ^upperl 80 ^fsmember "PC" ^confidence 900;
make DL ^lowerl 70 ^upperl 80 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 70 ^upperl 80 ^fsmember "CIPM" ^confidence 600;
```

```
make DL ^lowerl 70 ^upperl 80 ^fsmember "CIP" ^confidence 1000;
make DL ^lowerl 70 ^upperl 80 ^fsmember "STL" ^confidence 1000;
make DL ^lowerl 70 ^upperl 80 ^fsmember "CPS" ^confidence 1000;
make DL ^lowerl 70 ^upperl 80 ^fsmember "PIC" ^confidence 1000;
make DL ^lowerl 70 ^upperl 80 ^fsmember "APC" ^confidence 600;
make DL ^lowerl 70 ^upperl 80 ^fsmember "BP" ^confidence 500;
make DL ^lowerl 80 ^upperl 100 ^fsmember "PC" ^confidence 800;
make DL ^lowerl 80 ^upperl 100 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 80 ^upperl 100 ^fsmember "CIP" ^confidence 800;
make DL ^lowerl 80 ^upperl 100 ^fsmember "STL" ^confidence 1000;
make DL ^lowerl 80 ^upperl 100 ^fsmember "CPS" ^confidence 900;
make DL ^lowerl 80 ^upperl 100 ^fsmember "PIC" ^confidence 1000;
make DL ^lowerl 80 ^upperl 100 ^fsmember "BP" ^confidence 600;
make DL ^lowerl 100 ^upperl 120 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 100 ^upperl 120 ^fsmember "STL" ^confidence 1000;
make DL ^lowerl 100 ^upperl 120 ^fsmember "CPS" ^confidence 800;
make DL ^lowerl 100 ^upperl 120 ^fsmember "PIC" ^confidence 900;
make DL ^lowerl 100 ^upperl 120 ^fsmember "BP" ^confidence 700;
make DL ^lowerl 120 ^upperl 150 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 120 ^upperl 150 ^fsmember "STL" ^confidence 900;
make DL ^lowerl 120 ^upperl 150 ^fsmember "CPS" ^confidence 600;
make DL ^lowerl 120 ^upperl 150 ^fsmember "BP" ^confidence 800;
make DL ^lowerl 150 ^upperl 200 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 150 ^upperl 200 ^fsmember "STL" ^confidence 850;
make DL ^lowerl 150 ^upperl 200 ^fsmember "BP" ^confidence 1000;
```

```
make DL ^lowerl 200 ^upperl 350 ^fsmember "PSC" ^confidence 1000;
make DL ^lowerl 200 ^upperl 350 ^fsmember "BP" ^confidence 1000;
make DL ^lowerl 350 ^upperl 500 ^fsmember "PSC" ^confidence 900;
make DL ^lowerl 350 ^upperl 500 ^fsmember "BP" ^confidence 1000;
make DL ^lowerl 500 ^upperl 1000 ^fsmember "PSC" ^confidence 700;
make DL ^lowerl 500 ^upperl 1000 ^fsmember "CPS" ^confidence 1000;
make DL ^lowerl 500 ^upperl 1000 ^fsmember "BP" ^confidence 1000;
make DL ^lowerl 1000 ^upperl 1500 ^fsmember "CPS" ^confidence 1000;
```

Appendix E

List of SCKB Expert Knowledge File

```
:SOIL:      Cohesive soils
:factor1 = cohesive/cohesionless
:factor2 = stiffness of the bearing soil (soft/stiff/hard)
:factor3 = negative skin friction (likely/unlikely)
:factor4 = boulders present (yes/no)

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PC" ^confidence 900;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "STL" ^confidence 500;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "TM" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "APC" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PC" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 950;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "TM" ^confidence 700;
```

```
make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "APC" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "BP" ^confidence 900;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PC" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "STL" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "APC" ^confidence 600;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "BP" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 850;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 950;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 500;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "APC" ^confidence 600;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 950;
```

```
make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 950;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "APC" ^confidence 600;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "BP" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 750;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 750;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 950;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "BP" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PC" ^confidence 600;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 500;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "TM" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 600;
```

```
make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 600;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PC" ^confidence 750;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 750;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 500;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 950;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "TM" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PC" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "STL" ^confidence 1000;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "BP" ^confidence 900;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 550;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 700;
```

```
make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 700;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesive"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 500;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 650;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 900;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 750;

make SOIL  ^factor1 "Cohesive"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "BP" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PC" ^confidence 900;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "STL" ^confidence 500;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "TM" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 1000;
```

```
make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PIC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "APC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 950;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "TM" ^confidence 900;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 950;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PIC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "APC" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "BP" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PC" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "STL" ^confidence 1000;
```

```
make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "APC" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "no" ^fsmember "BP" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 850;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 950;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 500;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 750;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PIC" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "APC" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 950;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 950;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 650;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PIC" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "APC" ^confidence 600;
```

```
make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "BP" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 750;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 750;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 950;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "unlikely"
           ^factor4 "yes" ^fsmember "BP" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PC" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "TM" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PIC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PC" ^confidence 750;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 750;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIPM" ^confidence 600;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CIP" ^confidence 950;
```

```
make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "TM" ^confidence 900;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPW" ^confidence 650;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 650;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PIC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "PSC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "STL" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "no" ^fsmember "BP" ^confidence 900;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 550;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 650;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "soft" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PIC" ^confidence 500;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 700;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CIP" ^confidence 650;
```

```
make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 500;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "stiff" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PIC" ^confidence 500;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PC" ^confidence 650;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "PSC" ^confidence 650;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "STL" ^confidence 1000;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "CPS" ^confidence 800;

make SOIL  ^factor1 "Cohesionless"  ^factor2 "hard" ^factor3 "likely"
           ^factor4 "yes" ^fsmember "BP" ^confidence 750;
```

Appendix F

List of LTHKB Expert Knowledge File

```
:Length Parameters (feet)
:Key;
:PC = Precast Concrete piles
:PSC = Prestressed Concrete piles
:CIPM = Cast-in-Place with Mandrel
:CIP = Cast-in-Place without Mandrel
:STL = Steel pile (H, I, of open pipe)
:TM = Timber pile
:CPW = Composite Pile Wood/concrete
:CPS = Composite Pile Steel/concrete
:   ALSO Composite Pile Steel Pipe filled w/concrete & steel H-beam
:PIC = Pressure Injected Concrete
:APC = Auger Pressure injected Concrete
:BP = Bored/Cassion Piles

make LTH ^lower2 10 ^upper2 29 ^fsmember "TM" ^confidence 600;

make LTH ^lower2 10 ^upper2 29 ^fsmember "CIPM" ^confidence 1000;

make LTH ^lower2 10 ^upper2 29 ^fsmember "CIP" ^confidence 600;

make LTH ^lower2 10 ^upper2 29 ^fsmember "PIC" ^confidence 1000;

make LTH ^lower2 10 ^upper2 29 ^fsmember "BP" ^confidence 700;

make LTH ^lower2 29 ^upper2 39 ^fsmember "PC" ^confidence 700;

make LTH ^lower2 29 ^upper2 39 ^fsmember "CIPM" ^confidence 1000;

make LTH ^lower2 29 ^upper2 39 ^fsmember "CIP" ^confidence 1000;

make LTH ^lower2 29 ^upper2 39 ^fsmember "STL" ^confidence 600;

make LTH ^lower2 29 ^upper2 39 ^fsmember "TM" ^confidence 1000;

make LTH ^lower2 29 ^upper2 39 ^fsmember "PIC" ^confidence 1000;

make LTH ^lower2 29 ^upper2 39 ^fsmember "APC" ^confidence 800;

make LTH ^lower2 29 ^upper2 39 ^fsmember "BP" ^confidence 900;

make LTH ^lower2 39 ^upper2 50 ^fsmember "PC" ^confidence 1000;

make LTH ^lower2 39 ^upper2 50 ^fsmember "PSC" ^confidence 500;

make LTH ^lower2 39 ^upper2 50 ^fsmember "CIPM" ^confidence 1000;

make LTH ^lower2 39 ^upper2 50 ^fsmember "CIP" ^confidence 1000;

make LTH ^lower2 39 ^upper2 50 ^fsmember "STL" ^confidence 1000;
```

```
make LTH ^lower2 39 ^upper2 50 ^fsmember "TM" ^confidence 1000;
make LTH ^lower2 39 ^upper2 50 ^fsmember "PIC" ^confidence 1000;
make LTH ^lower2 39 ^upper2 50 ^fsmember "APC" ^confidence 1000;
make LTH ^lower2 39 ^upper2 50 ^fsmember "BP" ^confidence 1000;
make LTH ^lower2 50 ^upper2 59 ^fsmember "PC" ^confidence 800;
make LTH ^lower2 50 ^upper2 59 ^fsmember "PSC" ^confidence 700;
make LTH ^lower2 50 ^upper2 59 ^fsmember "CIPM" ^confidence 1000;
make LTH ^lower2 50 ^upper2 59 ^fsmember "CIP" ^confidence 1000;
make LTH ^lower2 50 ^upper2 59 ^fsmember "STL" ^confidence 1000;
make LTH ^lower2 50 ^upper2 59 ^fsmember "TM" ^confidence 1000;
make LTH ^lower2 50 ^upper2 59 ^fsmember "CPW" ^confidence 700;
make LTH ^lower2 50 ^upper2 59 ^fsmember "CPS" ^confidence 700;
make LTH ^lower2 50 ^upper2 59 ^fsmember "PIC" ^confidence 900;
make LTH ^lower2 50 ^upper2 59 ^fsmember "APC" ^confidence 1000;
make LTH ^lower2 50 ^upper2 59 ^fsmember "BP" ^confidence 1000;
make LTH ^lower2 59 ^upper2 80 ^fsmember "PC" ^confidence 600;
make LTH ^lower2 59 ^upper2 80 ^fsmember "PSC" ^confidence 1000;
make LTH ^lower2 59 ^upper2 80 ^fsmember "CIPM" ^confidence 800;
make LTH ^lower2 59 ^upper2 80 ^fsmember "CIP" ^confidence 1000;
make LTH ^lower2 59 ^upper2 80 ^fsmember "STL" ^confidence 1000;
make LTH ^lower2 59 ^upper2 80 ^fsmember "TM" ^confidence 600;
make LTH ^lower2 59 ^upper2 80 ^fsmember "CPW" ^confidence 900;
make LTH ^lower2 59 ^upper2 80 ^fsmember "CPS" ^confidence 900;
make LTH ^lower2 59 ^upper2 80 ^fsmember "PIC" ^confidence 600;
make LTH ^lower2 59 ^upper2 80 ^fsmember "APC" ^confidence 600;
```

```
make LTH ^lower2 59 ^upper2 80 ^fsmember "BP" ^confidence 1000;
make LTH ^lower2 80 ^upper2 100 ^fsmember "PSC" ^confidence 1000;
make LTH ^lower2 80 ^upper2 100 ^fsmember "CIPM" ^confidence 500;
make LTH ^lower2 80 ^upper2 100 ^fsmember "CIP" ^confidence 600;
make LTH ^lower2 80 ^upper2 100 ^fsmember "STL" ^confidence 1000;
make LTH ^lower2 80 ^upper2 100 ^fsmember "CPW" ^confidence 900;
make LTH ^lower2 80 ^upper2 100 ^fsmember "CPS" ^confidence 900;
make LTH ^lower2 80 ^upper2 100 ^fsmember "BP" ^confidence 1000;
make LTH ^lower2 100 ^upper2 120 ^fsmember "PSC" ^confidence 900;
make LTH ^lower2 100 ^upper2 120 ^fsmember "CIP" ^confidence 500;
make LTH ^lower2 100 ^upper2 120 ^fsmember "STL" ^confidence 1000;
make LTH ^lower2 100 ^upper2 120 ^fsmember "CPW" ^confidence 750;
make LTH ^lower2 100 ^upper2 120 ^fsmember "CPS" ^confidence 750;
make LTH ^lower2 100 ^upper2 120 ^fsmember "BP" ^confidence 800;
make LTH ^lower2 120 ^upper2 200 ^fsmember "PSC" ^confidence 750;
make LTH ^lower2 120 ^upper2 200 ^fsmember "STL" ^confidence 850;
make LTH ^lower2 120 ^upper2 200 ^fsmember "CPW" ^confidence 600;
make LTH ^lower2 120 ^upper2 200 ^fsmember "CPS" ^confidence 650;
make LTH ^lower2 200 ^upper2 399 ^fsmember "STL" ^confidence 700;
make LTH ^lower2 200 ^upper2 399 ^fsmember "CPS" ^confidence 600;
```

Appendix G

List of AOMKB Expert Knowledge File

```
:Availability of Materials
:Economically available = 1000
:Available, but not economical = 800
:Not available = 0

make AOMX ^parameter3 "economical"  ^confidence 1000;

make AOMX ^parameter3 "available"  ^confidence 800;
```

Appendix H

List of LEKB Expert Knowledge File

```
:Local Contractor Experience
:Expert experience = 1000
:Some experience = 800
:No experience = 300

make LEX ^parameter4 "expert"  ^confidence 1000;

make LEX ^parameter4 "some_experience"  ^confidence 800;

make LEX ^parameter4 "no_experience"  ^confidence 300;
```

Appendix I

List of NEKB Expert Knowledge File

```
:Extreme constraint = 0 for driven piles & 1000 for bored piles
:Constraint = 750-800 for driven piles & 1000 for bored piles
:No constraint = 1000 for both driven & bored piles

make NEX ^parameterl "constraint" ^fsmember "PC" ^confidence 775;

make NEX ^parameterl "no_constraint" ^fsmember "PC" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "PSC" ^confidence 775;

make NEX ^parameterl "no_constraint" ^fsmember "PSC" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "CIPM" ^confidence 780;

make NEX ^parameterl "no_constraint" ^fsmember "CIPM" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "CIP" ^confidence 800;

make NEX ^parameterl "no_constraint" ^fsmember "CIP" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "STL" ^confidence 800;

make NEX ^parameterl "no_constraint" ^fsmember "STL" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "TM" ^confidence 800;

make NEX ^parameterl "no_constraint" ^fsmember "TM" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "CPW" ^confidence 780;

make NEX ^parameterl "no_constraint" ^fsmember "CPW" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "CPS" ^confidence 780;

make NEX ^parameterl "no_constraint" ^fsmember "CPS" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "PIC" ^confidence 800;

make NEX ^parameterl "no_constraint" ^fsmember "PIC" ^confidence 1000;

make NEX ^parameterl "extreme_constraint" ^fsmember "APC" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "APC" ^confidence 1000;

make NEX ^parameterl "no_constraint" ^fsmember "APC" ^confidence 1000;

make NEX ^parameterl "extreme_constraint" ^fsmember "BP" ^confidence 1000;

make NEX ^parameterl "constraint" ^fsmember "BP" ^confidence 1000;

make NEX ^parameterl "no_constraint" ^fsmember "BP" ^confidence 1000;
```

Appendix J

List of VEKB Expert Knowledge File

```
:Extreme constraint = 0 for driven piles & 1000 for bored piles
:constraint = 750-800 for driven diles & 1000 for bored piles
:No constraint = 1000 for both driven & bored piles (to save computer memory,
:all "no constraint" selections have been deleted since the relative final
:output is not affected.)

make VEX ^parameter2 "constraint" ^fsmember "PC" ^confidence 750;

make VEX ^parameter2 "constraint" ^fsmember "PSC" ^confidence 750;

make VEX ^parameter2 "constraint" ^fsmember "CIPM" ^confidence 750;

make VEX ^parameter2 "constraint" ^fsmember "CIP" ^confidence 750;

make VEX ^parameter2 "constraint" ^fsmember "STL" ^confidence 800;

make VEX ^parameter2 "constraint" ^fsmember "TM" ^confidence 800;

make VEX ^parameter2 "constraint" ^fsmember "CPW" ^confidence 775;

make VEX ^parameter2 "constraint" ^fsmember "CPS" ^confidence 775;

make VEX ^parameter2 "constraint" ^fsmember "PIC" ^confidence 800;

make VEX ^parameter2 "extreme_constraint" ^fsmember "APC" ^confidence 1000;

make VEX ^parameter2 "constraint" ^fsmember "APC" ^confidence 1000;

make VEX ^parameter2 "extreme_constraint" ^fsmember "BP" ^confidence 1000;

make VEX ^parameter2 "constraint" ^fsmember "BP" ^confidence 1000;
```

Appendix K

Data for Problems 1 and 1A

User.dat

```
"xdata",70,1000 ,70,1000 ,"cohesive",1000, "stiff",1000, "unlikely",750,
"no",1000, "no_constraint",1000, "constraint",900, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "available",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "available",900,
"expert",1000, "available",900, "expert",1000, "economical",1000,
"some_experience",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000,
```

PILE TYPE DATA FOR PROBLEM 1

"pile_type",1000,1000,800,1000,1000,0,0,1000,1000,900,0,600,1000,800,1000,1000,6
00,900,900,600,600,1000,750,750,700,750,750,700,750,750,0,750,750,1000,1000,1000
,1000,1000,1000,1000,1000,1000,1000,1000,750,750,750,750,800,800,775,775,800,900
,900,1000,1000,1000,800,1000,1000,800,800,1000,1000,1000,1000,1000,1000,1000,100
0,1000,1000,1000,800,1000,1000

KEY OUTPUT OF ps1.c FOR PILE SELECTION OF PROBLEM 1

I. Confidence for selecting each of the eleven piles based on each of the seven criteria.

PC - precast concrete pile
PSC - prestressed concrete pile
CIPM - cast-in-place with mandrel
CIP - cast-in-place without mandrel
STL - steel pile (H, I, or Pipe)
TM - timber pile
CPW - composite pile (wood + concrete)
CPS - composite pile (steel + concrete)
PIC - pressure injected concrete pile
APC - auger pressured pile
BP - bored pile

PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
1000	1000	800	1000	1000	0	0	1000	1000	900	0
600	1000	800	1000	1000	600	900	900	600	600	1000
750	750	700	750	750	700	750	750	0	750	750
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
750	750	750	750	800	800	775	775	800	900	900
1000	1000	1000	800	1000	1000	800	800	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	800	1000	1000

II. Weighting factors for the seven criteria:

1000
1000
1000
1000
1000
1000
1000

The seven criteria used, in the order of the above listed weight are as follows:

DL - design load
LTH - approximate pile embedment requirement
SOIL - subsurface soil conditions
NE - noise effects
VE - vibration effects
AOM - availability of pile material
LE - local experience

III. Final Confidence for selecting a particular pile:

pile:	PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
conf:	87	92	86	90	93	72	74	88	74	87	80

***** END OF output.ps1 *****

KEY OUTPUT OF ps1.c FOR PROBLEM 1A PILE SELECTION

I. Confidence for selecting each of the eleven piles
based on each of the seven criteria.

PC - precast concrete pile
PSC - prestressed concrete pile
CIPM - cast-in-place with mandrel
CIP - cast-in-place without mandrel
STL - steel pile (H, I, or Pipe)
TM - timber pile
CPW - composite pile (wood + concrete)
CPS - composite pile (steel + concrete)
PIC - pressure injected concrete pile
APC - auger pressured pile
BP - bored pile

PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
1000	1000	800	1000	1000	0	0	1000	1000	900	0
600	1000	800	1000	1000	600	900	900	600	600	1000
750	750	700	750	750	700	750	750	0	750	750
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
750	750	750	750	800	800	775	775	800	900	900
1000	1000	1000	800	1000	1000	800	800	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	800	1000	1000

II. Weighting factors for the seven criteria:

200
200
800
500
1000
500
800

The seven criteria used, in the order of the
above listed weight are as follows:

DL - design load
LTH - approximate pile embedment requirement
SOIL - subsurface soil conditions
NE - noise effects
VE - vibration effects
AOM - availability of pile material
LE - local experience

III. Final Confidence for selecting a particular pile:

pile:	PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
conf:	86	88	85	86	90	82	81	86	69	90	87

***** END OF output.ps1 *****

Appendix L

Data For Problems 2 and 2A

User.dat

```
"xdata",100,1000 ,120,1000 ,"cohesive",1000, "soft",1000, "likely",900,
"yes",1000, "constraint",1000, "extreme_constraint",1000, "available",1000,
"expert",1000, "available",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000,
```

KEY OUTPUT OF ps1.c FOR PROBLEM 2 PILE SELECTION

I. Confidence for selecting each of the eleven piles based on each of the seven criteria.

PC - precast concrete pile
PSC - prestressed concrete pile
CIPM - cast-in-place with mandrel
CIP - cast-in-place without mandrel
STL - steel pile (H, I, or Pipe)
TM - timber pile
CPW - composite pile (wood + concrete)
CPS - composite pile (steel + concrete)
PIC - pressure injected concrete pile
APC - auger pressured pile
BP - bored pile

PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
800	1000	0	800	1000	0	0	900	1000	0	600
0	900	0	500	1000	0	750	750	0	0	800
550	650	0	700	0	0	0	0	0	0	0
775	775	780	800	800	800	780	780	800	1000	1000
0	0	0	0	0	0	0	0	0	1000	1000
800	800	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

II. Weighting factors for the seven criteria:

1000
1000
1000
1000
1000
1000
1000

The seven criteria used, in the order of the above listed weight are as follows:

DL - design load
LTH - approximate pile embedment requirement
SOIL - subsurface soil conditions
NE - noise effects
VE - vibration effects
AOM - availability of pile material
LE - local experience

III. Final Confidence for selecting a particular pile:

pile:	PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
conf:	56	73	39	68	68	40	50	63	54	57	77

***** END OF output.ps1 *****

KEY OUTPUT OF ps1.c FOR PROBLEM 2A PILE SELECTION

I. Confidence for selecting each of the eleven piles
based on each of the seven criteria.

PC - precast concrete pile
PSC - prestressed concrete pile
CIPM - cast-in-place with mandrel
CIP - cast-in-place without mandrel
STL - steel pile (H, I, or Pipe)
TM - timber pile
CPW - composite pile (wood + concrete)
CPS - composite pile (steel + concrete)
PIC - pressure injected concrete pile
APC - auger pressured pile
BP - bored pile

PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
800	1000	0	800	1000	0	0	900	1000	0	600
0	900	0	500	1000	0	750	750	0	0	800
550	650	0	700	0	0	0	0	0	0	0
775	775	780	800	800	800	780	780	800	1000	1000
0	0	0	0	0	0	0	0	0	1000	1000
800	800	1000	1000	1000	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

II. Weighting factors for the seven criteria:

1000
800
800
500
200
200
500

The seven criteria used, in the order of the
above listed weight are as follows:

DL - design load
LTH - approximate pile embedment requirement
SOIL - subsurface soil conditions
NE - noise effects
VE - vibration effects
AOM - availability of pile material
LE - local experience

III. Final Confidence for selecting a particular pile:

pile:	PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
conf:	57	82	27	71	72	27	42	64	52	35	66

***** END OF output.ps1 *****

Appendix M

Data for Problems 3 and 3A

User.dat

```
"xdata",25,1000 ,50,1000 ,"cohesionless",1000, "stiff",1000, "unlikely",900,
"no",1000, "no_constraint",1000, "constraint",1000, "not_available",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "available",900,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000, "economical",1000, "expert",1000, "economical",1000,
"expert",1000,
```

KEY OUTPUT OF psl.c FOR PROBLEM 3 PILE SELECTION

I. Confidence for selecting each of the eleven piles
based on each of the seven criteria.

PC - precast concrete pile
PSC - prestressed concrete pile
CIPM - cast-in-place with mandrel
CIP - cast-in-place without mandrel
STL - steel pile (H, I, or Pipe)
TM - timber pile
CPW - composite pile (wood + concrete)
CPS - composite pile (steel + concrete)
PIC - pressure injected concrete pile
APC - auger pressured pile
BP - bored pile

PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
0	0	800	0	0	1000	700	0	0	0	0
1000	500	1000	1000	1000	1000	0	0	1000	1000	1000
900	900	900	900	800	900	900	900	900	900	700
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
750	750	750	750	800	800	775	775	800	1000	1000
0	1000	1000	1000	800	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

II. Weighting factors for the seven criteria:

1000
1000
1000
1000
1000
1000
1000

The seven criteria used, in the order of the
above listed weight are as follows:

DL - design load
LTH - approximate pile embedment requirement
SOIL - subsurface soil conditions
NE - noise effects
VE - vibration effects
AOM - availability of pile material
LE - local experience

III. Final Confidence for selecting a particular pile:

pile:	PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
conf:	66	73	92	80	77	95	76	66	81	84	81

***** END OF output.psl *****

KEY OUTPUT OF ps1.c FOR PROBLEM 3A PILE SELECTION

I. Confidence for selecting each of the eleven piles
based on each of the seven criteria.

PC - precast concrete pile
PSC - prestressed concrete pile
CIPM - cast-in-place with mandrel
CIP - cast-in-place without mandrel
STL - steel pile (H, I, or Pipe)
TM - timber pile
CPW - composite pile (wood + concrete)
CPS - composite pile (steel + concrete)
PIC - pressure injected concrete pile
APC - auger pressured pile
BP - bored pile

PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
0	0	800	0	0	1000	700	0	0	0	0
1000	500	1000	1000	1000	1000	0	0	1000	1000	1000
900	900	900	900	800	900	900	900	900	900	700
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
750	750	750	750	800	800	775	775	800	1000	1000
0	1000	1000	1000	800	1000	1000	1000	1000	1000	1000
1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

II. Weighting factors for the seven criteria:

200
500
200
500
1000
800
800

The seven criteria used, in the order of the
above listed weight are as follows:

DL - design load
LTH - approximate pile embedment requirement
SOIL - subsurface soil conditions
NE - noise effects
VE - vibration effects
AOM - availability of pile material
LE - local experience

III. Final Confidence for selecting a particular pile:

pile:	PC	PSC	CIPM	CIP	STL	TM	CPW	CPS	PIC	APC	BP
conf:	68	82	92	88	85	94	79	76	89	94	93

***** END OF output.ps1 *****

REFERENCES CITED

1. Vanikar, S. N. Manual on Design and Construction of Driven Pile Foundations, (U. S. Department of Transportation, Federal Highway Administration, April 1986).

2. Whitaker, T. The Design of Piled Foundations, (Pergamon Press, New York, 1976).

3. Das, B. M. Principles of Foundation Engineering, (PWS Engineering, Boston, 1984), Chap. 8.

4. Sowers, G. B. and Sowers, G. F. Introductory Soil Mechanics and Foundations, (The Macmillan Co., Taiwan, 1970), 3rd Ed., Chap. 6.

5. Canadian Foundation Engineering Manual, The Canadian Geotechnical Society, (BiTech Publishers Ltd., Vancouver, B.C., 1985), Chap. 19, 20, 21.

6. Foundations and Earth Structures, Design Manual 7.2, U. S. Navy, 1982, Chap. 5.

7. Bjerrum, J. L. "Reduction of negative skin friction on steel piles to rock," Proceedings, 7th Int. Conf. Soil Mech., Mexico, 1969.

8. Fellenius, B. H. "Negative skin friction on long piles driven in clay," No. 18, Royal Swedish Academy of Engineering Sciences, Stockholm, 1971.

9. Bowles, J. E. Foundation Analysis and Design, (McGraw-Hill Co., New York, 1988), 4th Ed., Chap. 16, 17.

10. Poulos and Davis Pile Foundation Analysis and Design, 1980.

11. O'Neill, M. W. Pile Group Prediction Symposium: Summary, Federal Highway Administration, 1987.

12. Coyle, H. M. and Castello, R. "New Design Correlations for Piles in Sand." Journal of Geotech. Engrg. Div., ASCE. 107 No. 7, Jul 1981, pp. 965-986.

13. Cheney, R. S. and Chassie, R. G. Soils and Foundations Workshop Manual, U. S. Department of Transportation, Federal Highway Administration (HHO-33), 1982.

14. "Deep Foundations," CE 833, Clemson University, Jan - May 1988.

15. Schmertmann, J. H. "Guidelines for Cone Penetration Tes: Perform -ance and Design," FHWA-TS-78-209, U. S. Dept. ofTransportation, 1978.

16. Nordlund, R. L. "Bearing Capacity of Piles in Cohesionless Soils," Journal of the Soil Mechanics and Foundation Div., ASCE, 89, No. 3, May 1963, pp. 1-35.

17. Thurman, A. G. Discussion of "Bearing Capacity of Piles in Cohesionless Soils," by R. L. Nordlund, Journal of Soil Mechanics and Foundation Div., ASCE, 90, No. 1, Jan. 1964, pp. 127-129.

18. Wey, J. L. "Predicting Load Carrying Capacity of Single Pile in Sand," Thesis presented to Univ. of Clemson, S. C., in partial fulfillment of the requirements for the degree of Master of Science Civil Engineering, 1987.

19. Meyerhof, G. G. "Bearing Capacity and Settlement of Pile Foundations," Journal of the Geotech. Engineering Div., ASCE, Vol. 102, No. GT3, March 1976.

20. Briaud, J. L. and Tucker, L. M. "Piles in Sand: A Method Including Residual Stresses," Journal of Geotech. Engineering, ASCE, Vol. 111, No. 11, Nov. 1984.

21. Recommended Procedure for Planning, Design, and Constructing Fixed Offshore Platforms, American Petroleum Institite, API RP2A, 15th Ed., 1984.

22. Bustamante, M. and Gianeselli, L. "Prevision de la Capacite Portante des Pieux par la Methode Penetrometrique" (version 83), Compte Rendu de Recherche F.A.E.R. 1.05.02.2, Lavoratoire Central des Ponts et Chaussees, 1983.

23. DeRuiter, J. and Feringen, F. "Pile Foundations for Large North Sea Structures," Marine Geotechnology, Vol. 3, No. 3, 1979.

24. "Michigan Pile Test Program Test Results are Released," Eng News-Record, ENR, pp. 26-28, 33-34, May 20, 1965.

25. Hiley, A. "Pile-Driving Calculations with Notes on Driving Forecs and Ground Resistances," The Structural Engineer, London, Vol. 8, pp. 246-259, 278-288.

26. Wave Equation Analysis Program, 1986.

27. Vesic, A. S. "Design of Pile Foundation," NCHRP Synthesis of Highway Practice, No. 42, Transportation Research Board, 1977.

28. Sidi, I. D. "Probabilistic Prediction of Pile Capacities," Thesis presented to the University of Illinois-Urbana, in partial fulfillment of the requirement for the degree of Doctor of Philosophy, May 1986.

29. Chellis, R. D. Pile Foundation, (MacGraw-Hill Co., New York, 1961), 2nd Ed.

30. Kezdi, A. "Pile Foundations," In H. F. Winterkorn and K. Y. Fang, Eds. Foundation Engineering Handbook, Van Nostrand Reinhold, New York, 1975).

31. Kuthy et al. "Lateral Load Capacity of Vertical Pile Groups," No. NYSDOT-ERD-77-RR47, Apr. 1977.

32. Meyer and Reese "Analysis of Single Piles Under Lateral Loading," No. FHWA/TX-79/38+244-1, 1979.

33. Siler, W. S. and Tucker, D. A Fuzzy Logic Production System, (Kemp-Carraway Heart Institute, 1986)

34. Forgy, C.L., OPS5 User's Manual, Technical Report CMU-CS-81-135, Carnegie-Mellon University, 1981.

35. Zadeh, L.A., "Fuzzy Sets", Inf. Control, Vol. 8, pp.338-353, 1965.

36. Zadeh, L.A., "Outline of A New Approach to The Analysis of Complex Systems and Decision Process", IEEE Trans. on Systems, Man and Cybernetics SMCX-3, pp. 28-44, 1973.

37. Blockley, D.I., "The Role of Fuzzy Sets in Civil Engineering", Fuzzy Sets and Systems, No. 2, pp. 267-278, 1979.

38. Brown, C., "A Fuzzy Safety Measure", J. Engrg. Mech., ASCE, Vol. 105, No. EM5, pp. 855-872, 1979.

39. Brown, C. and Yao, J.T.P., "Fuzzy Sets in Structural Engineering", J. Structural Engrg., ASCE, Vol. 109, No. 5, pp. 1211-1225, 1983.

40. Chameau, J., Altschaeffl, A., Michael, H.L. and Yao, J.T.P., "Potential Applications of Fuzzy Sets in Civil Engineering", Int. J. Man-Mach. Stud., Vol. 19, pp. 9-18, 1983

41. Dong, W. and Wong, F.S., "Fuzzy Weighted Averages and Implementation of the Extension Principle", Fuzzy Sets and Systems, Vol. 21, pp. 183-199, 1987.

42. Hinkle, A.J. and Yao, J.T.P., "Linguistic Assessment of Welded Structures with Fatigue Damage", J. Approximate Reasoning, Vol.2, pp. 47-63, 1988.

43. Juang, C.H. and Elton, D.J., "Fuzzy Logic for Estimation of Earthquake Intensity Based on Building Damage Records", Civil Engineering Systems, Vol. 3, pp. 187-191, 1986.

44. Juang, C.H., Burati, J.L., and Kalidindi, S.N., "A Fuzzy System for Bid Proposal Evaluation Using Microcomputers", Civil Engineering Systems, Vol. 4, pp. 124-130, 1987.

45. Juang, C.H., "Development of A Decision Support System Using Fuzzy Sets", J. Microcomputers in Civil Engineering, July, 1988.

46. Juang, C.H., Fuzzy Set Theory and Its Applications in Civil Engineering, A NSC Workshop Manual on Civil Engineering Applications of Fuzzy Sets, July, National Cheng Kung University, Taiwan, 1988.

47. Mullarkey, P.W. and Fenves, S.J., "Fuzzy Logic in a Geotechnical Knowledge-Based System: CONE", Proc. NSF Workshop on Civil Engineering Applications of Fuzzy Sets, Purdue Universuty, Oct., pp. 126-169, 1985.

48. Yao, J.T.P., "Damage Assessment of Existing Structures", J. Engrg. Mech., ASCE, Vol. 106, No. EM4, pp. 785-799, 1980.

49. Yao, J.T.P., Safety and Reliability of Existing Structures, (Pitman Advanced Publishing, Boston, MASS., 1985).

50. Microsoft C Compiler, Version 5.1, Microsoft Corporation, USA, 1988.

51. Fuller, F. M. Engineering of Pile Installations, (McGraw-Hill Co., New York, 1983)

52. Spangler, M. G. and Handy, R. L. Soil Engineering, (Harper and Row, Publishers, New York, 1982) 4th Ed.

53. Hunt, H. W. Piletips...Design and Installation of Pile Foundations, Associated Pile and Fitting Corp., 1974.

54. "Timber Piles and Construction Timbers," ASCE Manual of Practice No. 17, 1959.

55. West, A. S. Piling Practice, 1972.

56. Greer, D. M. and Gardner, W. S. Construction of Drilled Pier Foundations, (John Wiley and Sons, New York, 1986).