Civil Engineering Department - CIVE 711

CIV E 711 - COMPUTER-AIDED PROJECT ORGANIZATION & MANAGEMENT

Instructor: Dr. Tarek Hegazy, CPH 2369CG, Ext.: 32174, E-mail: tarek@uwaterloo.ca

Description:

Application of traditional and Artificial Intelligence-based computerized tools for effectively managing the time, money, and resources of projects. It covers: review of the CPM method, project management software, optimization using Excel Solver, Expert Systems, Neural Networks, OOP programming, Genetic Algorithms, Fuzzy Logic, integrated project management tools, Asset Management Systems, various case studies and hands-on computer workshops. The course involves assignments, computer workshops, a project, and a final examination.

Suggested Texts:

(1) Hegazy 2002, "Computer-Based Construction Project Management," Prentice Hall.
(2) Negnevitsky, M. 2005 "Artificial Intelligence" A guide to intelligent systems, 2nd Ed., Addison Wesley.
(3) Hendrickson, C. and Au, T. "Project management for Construction: Fundamental Concepts for Owners, Engineers, Architects, and Builders," Prentice Hall, 1989.
(4) Ahuja, H.N. "Construction Performance Control by Networks," John Wiley & Sons, 1976.
(5) Clough, R.H. and Sears, E. "Construction Project Management," Second Edition, John Wiley & Sons, Toronto, 1979.

Tentative Content:

Week	Subject
1	• Introduction to Project Management.
2	• Optimization using Excel Solver.
3	• EasyPlan & Microsoft Project Software.
4	• Genetic Algorithms.
5	• AI & Expert Systems.
6	• Neural Networks.
7	• Fuzzy Logic.
8	• Hybrid AI tools.
9	• Asset Management.
10	• Planning of repetitive projects.
11	• Project control techniques & Earned-
12	• Class presentations.

References on Project Management:

- Books on Project Management and Construction Management;
- Trade magazines (e.g., ENR);
- International journals such as:
    - Construction Engineering and Management (ASCE);
    - Computing in Civil Engineering (ASCE);
    - Infrastructure Systems (ASCE);
    - Computer-Aided Civil and Infrastructure Engineering;
    - Automation in Construction;
    - Cost Engineering (AACE);
    - Construction Management and Economics;
    - Knowledge-Based Systems;
    - Quality in Maintenance & Engineering; and
    - Computers in Industry.

- Databases such as "current contents" , "compendex" & "CISTI";
- International organizations such as Project Management Institute (PMI) and American Association of Cost Engineers (AACE);
- A lot of computer software programs;
- Internet search;
- News groups; and
- Government publications such as statistics Canada, etc.

Interesting Project Management Proverbs:

- If you fail to plan, you plan to fail.
- There are no good project managers - only lucky ones. The more you plan the luckier you get.
- Fast - cheap - good: you can have any two, not all three.
- Be realistic. If it takes 1 person 1 hour to go to Toronto, it does not take half an hour from 2 people.
- The person who says it will take the longest and cost the most is the most knowledgeable.
- The most valuable and least used WORD in a project manager's vocabulary is "NO".
- The most valuable and least used PHRASE in a project manager's vocabulary is "I don't know".
- If it happens once it's ignorance, if it happens twice it's neglect, if three times it's policy.
- You can get someone to commit to a strict deadline, but you cannot get him into meeting it.
- A badly planned project will take three times than expected - a well planned project only twice.
- The sooner you get behind schedule, the more time you have to make it up.
- A problem shared is a buck passed.
- Of several possible interpretations of a communication, the least convenient is the correct one.
- If everything is going exactly to plan, something somewhere is going massively wrong.
- Project management tools are used most for predicting, not preventing, cost & schedule overruns.
- For a project manager, overruns are as certain as death and taxes.
- Some projects finish on time in spite of project management best practices.
- When the project's paperwork weighs as much as the project itself, the project is complete.
- If you can interpret project status in several different ways, the most painful will be correct.
- A project ain't over until the fat cheque is cashed.
- No project has ever finished on time, within budget, to requirement - yours won't be the first.
- Good control reveals problems early - which means you'll have longer to worry about them.
- If it can go wrong, it will - Murphy's Law.
- Work expands to fill the time available for its completion - Parkinson's Law.
- The common 7 phases of a project are: Wild enthusiasm; Disillusionment; Confusion; Panic; Search for the guilty; Punishment of the innocent; and Promotion of non-participants.

CIVE 711- Current Research Areas

Current Research Trends
Journal of Computing in Civil Engineering, ASCE, 20(1), 2006

Topic
Period 3 (1997–2003)
Neural networks
Algorithms
Fuzzy sets
Modeling
Imaging techniques
Optimization
Geographic information systems
Decision support systems
Scheduling
Computer software

Topic	Number of papers
Modeling	51
Neural networks	50
Expert systems	46
Design	34
Algorithms	31
Knowledge based systems	28
Simulation	26
Computer software	24
Computer programs	19
Automation	18
Optimization	17
Geographic infomation systems	16
Structural design	16
Decision support systems	15
Scheduling	15
Object oriented languages	13
Fuzzy sets	13
Microcomputers	12
Imaging techniques	10

Titles of research papers as of 2005

ASCE

- Computerized System for Efficient Delivery of Infrastructure Maintenance/Repair Programs
- Work Continuity Constraints in Project Scheduling
- Object-Oriented Scheduling for Repetitive Projects with Soft Logics
- Use of a WBS Matrix to Improve Interface Management in Projects
- Module-Based Construction Schedule Administration for Public Infrastructure Agencies
- Finance-Based Scheduling of Construction Projects Using Integer Programming
- Effective Practice Utilization Using Performance Prediction Software
- Flexible Work Breakdown Structure for Integrated Cost and Schedule Control
- Planning and Scheduling Highway Construction
- Accuracy of Highway Contractor's Schedules
- Fuzzy Logic Approach for Activity Delay Analysis and Schedule Updating
- Critical Path Method with Multiple Calendars
- Quantifying Engineering Project Scope for Productivity Modeling
- Benchmarking of Construction Productivity
- Predicting Industrial Construction Labor Productivity Using Fuzzy Expert Systems
- Time-Cost Optimization of Construction Projects with Generalized Activity Constraints
- Method for Calculating Schedule Delay Considering Lost Productivity
- Delay Analysis Method Using Delay Section
- Impact of Change's Timing on Labor Productivity
- Improved Measured Mile Analysis Technique
- MBF: Modified But-For Method for delay analysis
- Daily windows analysis method

Computing in Civil Engineering

- Web-Vacuum: Web-Based Environment for Automated
- Methodology for the Integration of Project Documents in Model-Based Information Systems
- Framework for Managing Life-Cycle Cost of Smart Infrastructure Systems
- Optimum Bid Markup Calculation Using Neuro fuzzy Systems and Multidimensional Risk Analysis Algorithm
- Modeling and Predicting Biological Performance of Contact Stabilization Process Using Artificial Neural Networks
- Building Project Model Support for Automated Labor Monitoring
- Parallel Computing Framework for Optimizing Construction Planning in Large-Scale Projects

Automation in Construction

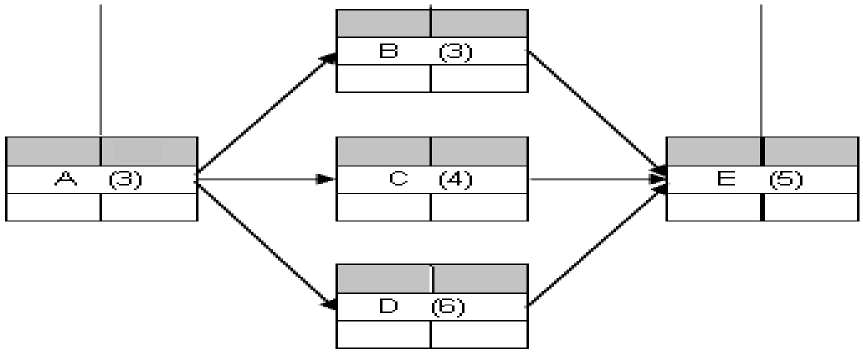
- Applications of electronically facilitated bidding model to preventing construction disputes
- A formalism for utilization of sensor systems and integrated project models for active construction quality control
- Rapid, on-site spatial information acquisition and its use for infrastructure operation and maintenance
- Data modeling issues in simulating the dynamic processes in life cycle analysis of buildings

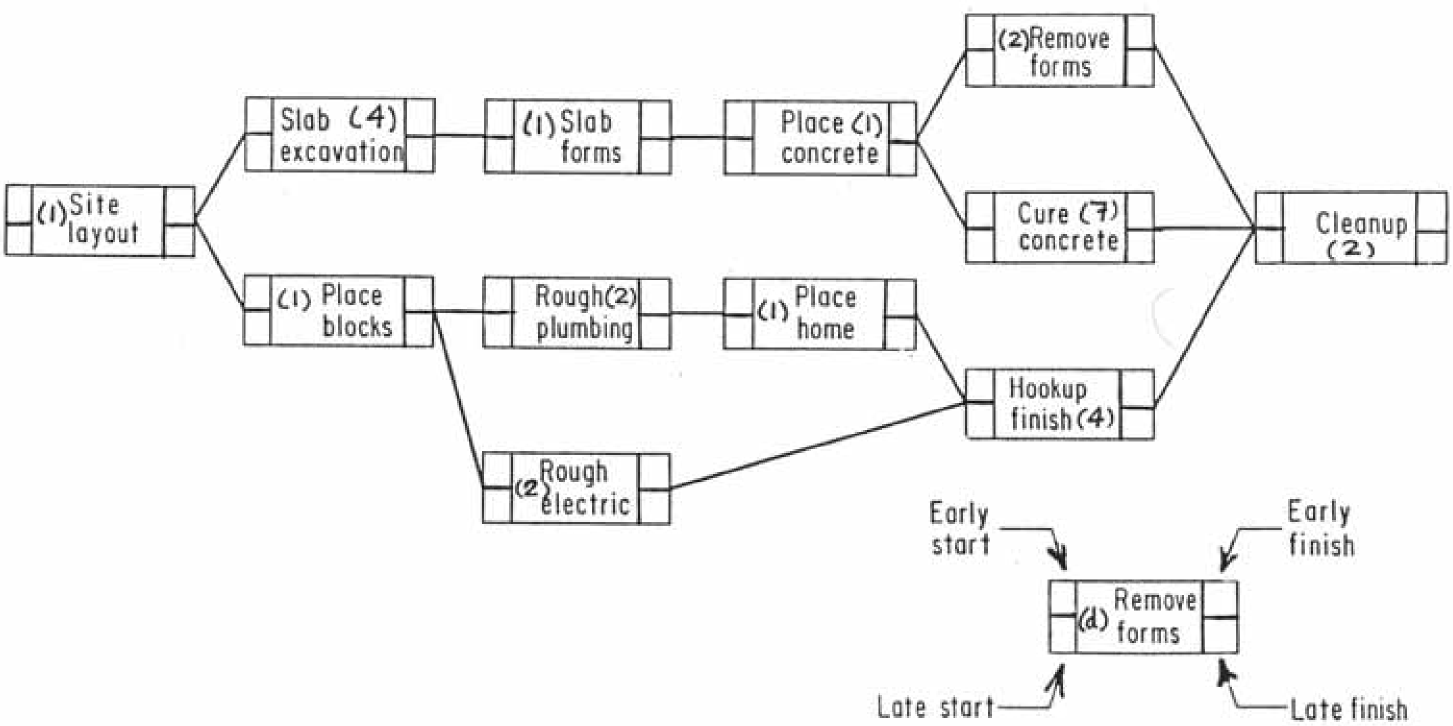
Content-Based Search Engines for construction image databases
Object-oriented framework for durability assessment and life cycle costing of highway bridges
Dynamic planning and control methodology for strategic and operational construction project management
Planning gang formwork operations for building construction using simulations
Application of integrated GPS and GIS technology for reducing construction waste and improving construction efficiency
Maintenance optimization of infrastructure networks using genetic algorithms
A computer-based scoring method for measuring the environmental performance of construction activities
Model-based dynamic resource management for construction projects
4D dynamic management for construction planning and resource utilization
PPMS: a Web-based construction Project Performance Monitoring System
Automated project performance control of construction projects
A cooperative Internet-facilitated quality management environment for construction
A process-based quality management information system
System development for non-unit based repetitive project scheduling
Simulation-based optimization for dynamic resource allocation
A WICE approach to real-time construction cost estimation

Construction Management and Economics

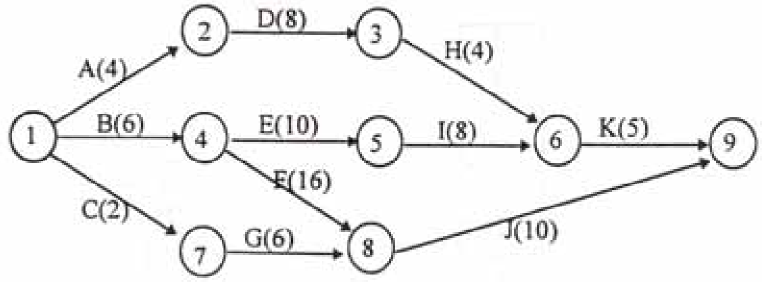
Integrated maintenance management of hospital buildings: a case study
Web-based integrated project control system
Service quality performance of design/build contractors using quality function deployment
Safety and production: an integrated planning and control model
Trends of 4D CAD applications for construction planning
An integrated construction project cost information system using MS Access and MS Project
Production arrangements by US building and non-building contractors: an update
A typology for clients' multi-project environments
Combining various facets of uncertainty in whole-life cost modeling
Grey relation analysis of causes for change orders in highway construction
Development of a model to estimate the benefit-cost ratio performance of housing
Scheduling system with focus on practical concerns in repetitive projects
Managing knowledge: lessons from the oil and gas sector
Simulation of maintenance costs in UK local authority sport centers
Project management decisions with multiple fuzzy goals
Service life prediction of exterior cladding components under standard conditions
Documentation, standardization and improvement of the construction process in house building
Innovative construction technology for affordable mass housing in Tanzania, East Africa
Selection of vertical formwork system by probabilistic neural networks models
Project cost estimation using principal component regression
Using linear model for learning curve effect on highrise floor construction
Accelerating linear projects
Predicting the risk of contractor default in Saudi Arabia utilizing artificial neural network (ANN) and genetic algorithm (GA) techniques
Forecasting the residual service life of NHS hospital buildings: a stochastic approach
Using the principal component analysis method as a tool in contractor pre qualification
The JIT materials management system in developing countries
Use of information and communication technologies by small and medium sized enterprises (SMEs) in building construction
Identifying management research priorities
A linear discrete scheduling model for the resource constrained project scheduling problem
Justification time management in the ready mixed concrete industries of Chongqing, China and Singapore
A model for automated monitoring of road construction
Improvement tools in the UK construction industry
Time series forecasts of the construction labour market in Hong Kong: the Box Jenkins approach

Basics of Scheduling





Another Example



- Calculate ES, LF, & TF for all activities. What are the critical ones ?
- Draw an Early Bar Chart for the project.
- What is the effect of delaying activity H by two days on the total project duration ?

Activity	Duration	ES	LF	TF=LF-ES-d	Critical
A	4				
B	6				
C	2				
D	8				
E	10				
F	16				
G	6				
H	4				
I	8				
J	10				
K	5				

ACTIVITY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
A																																
B																																
C																																
D																																
E																																
F																																
G																																
H																																
I																																
J																																
K																																

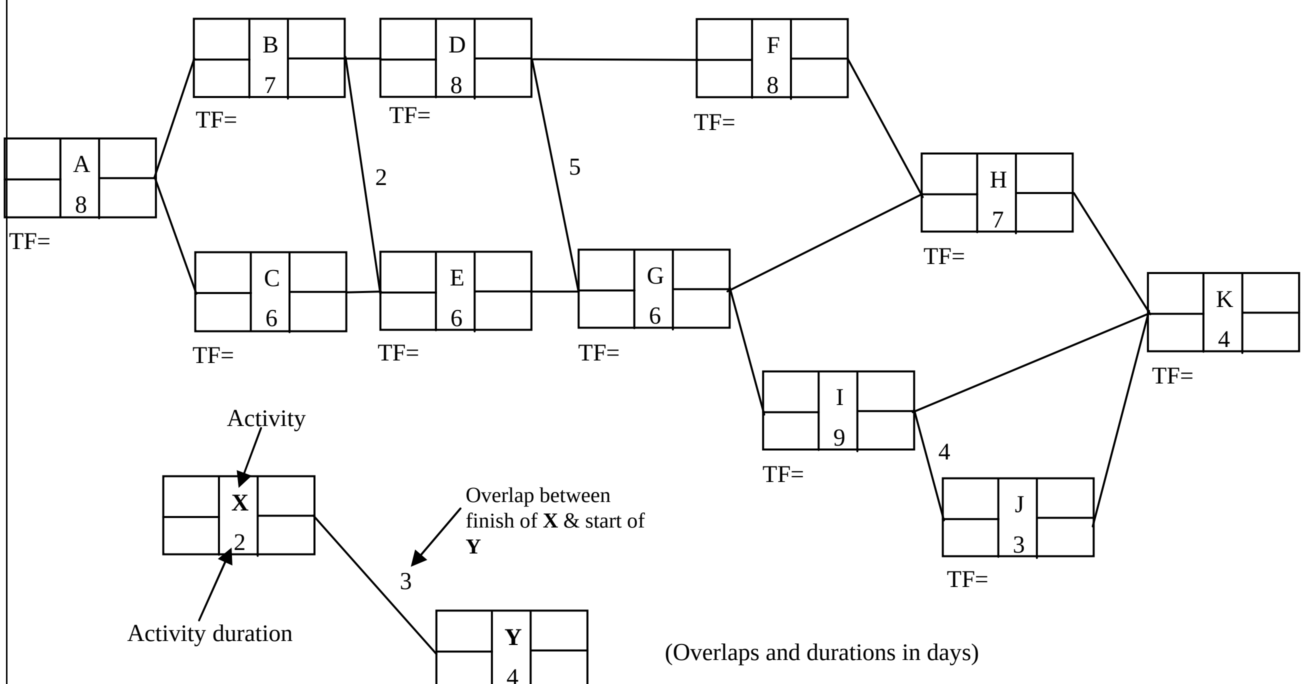
CIVE 711 – Scheduling Assignment

1. For the following project:

   A. Manually draw the logic network and identify the critical path.
   B. Manually draw a “late” bar chart.
   C. What is the effect of delaying activity G by 6 days?
   D. Enter the project into EasyPlan, print the schedule, the network, and the cash flow chart.

Activity	Predecessors	Duration (days)	Cost (x$1000)
A	----	4	5
B	A	6	3
C	A	4	4
D	A	9	2
E	B	3	4
F	D	8	5
G	B	10	2
H	C, E	2	2
I	F	4	4
J	G, H, I	2	3

2. Manually calculate the schedule for the following network. Enter the data into Microsoft Project and print the resulting schedule.



3. In EasyPlan, use the “Web Tutorial” and load Pr7 (as discussed in the project management article). Solve the exercise and print your score and the optimum schedule. Note that the part of dealing with actual progress data is not part of the assignment.

Introduction to Artificial Intelligence

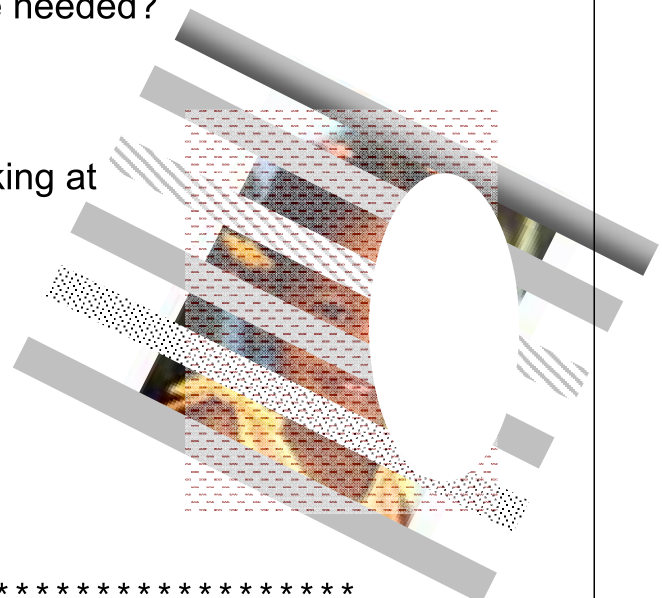
-Take a minute to calculate the following manually:

$45.32\ (98.2 \times 123.59)^{12} / 27.2 = ?$

If not done, give it to a cheap $1 calculator. Who is smart now?

- Consider the example of O.J. Simpson's Trial and the conflicting evidence. How did the jury and judge make a decision? Can the judge explain the logic? Was any calculation involved? How much time needed?

- Can you guess the age of this person by looking at the picture for only 2 seconds?



Was any calculation involved?
Can you explain your logic?
Can your computer do that?

****

Can you read this ?

can uoy blveiee taht I cluod auclaly uesdnatnrd waht I was redanieg. The phaonmneal pweor of the hmuan mnid, aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoatnt thing is taht the frist and lsat ltteer be in the rghit pclae. The rset can be a taotl mses and you can sitll raed it wouthit a porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe. Amzanig huh? Yaeh and I awlyas tghuhot slpeling was ipmorantt!

****

Who has more processing power: A supercomputer or the brain of a fly?
Who is more intelligent?

How to add intelligence to computers?

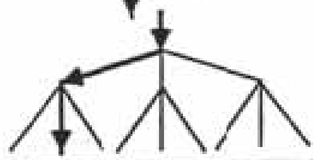
PROBLEM SOLVING TECHNIQUES

COMPUTATION

6598.4
X
123.98

Human mind is not geared to computations

REASONING

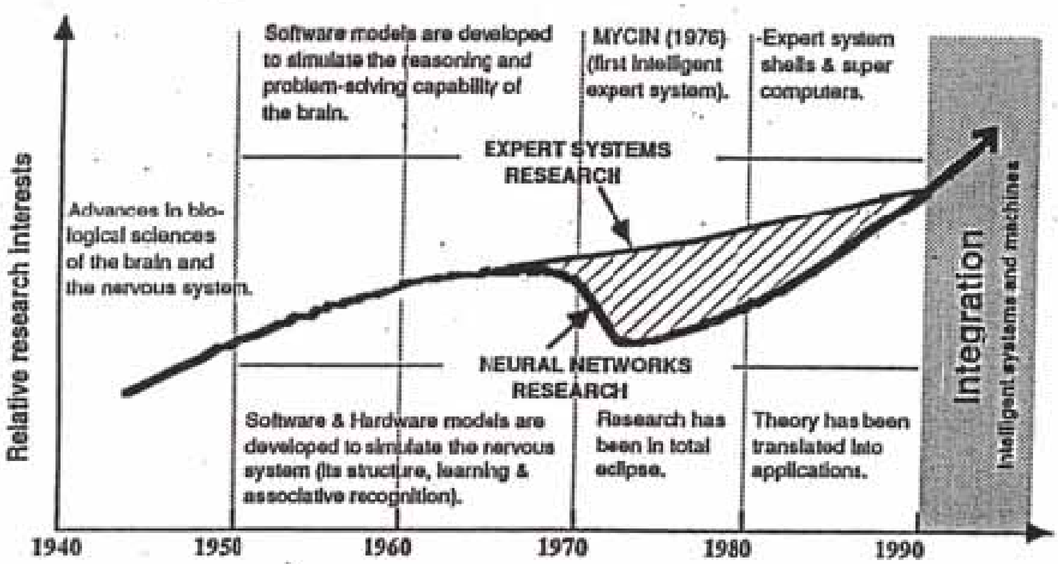


Human mind can reason effectively provided that time and knowledge are available.

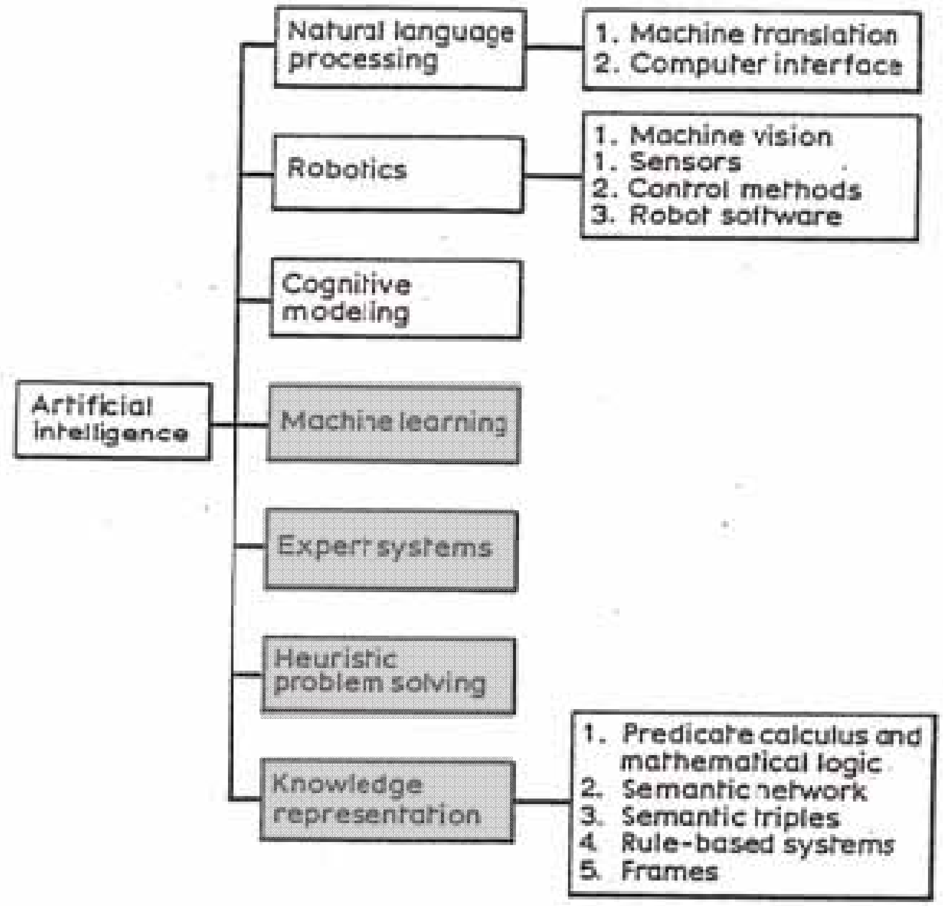
RECOGNITION

Human mind can recognize very effectively with no time or effort, and even based on partial cues.

RECOGNITION PROVIDES FEASIBLE ALTERNATIVES, FROM WHICH, REASONING AND COMPUTATIONS CAN SELECT THE BEST ONE.



Historical evolution of Neural Networks and Expert Systems.



Research in artificial intelligence (AI).

Solving Optimization Problems:

Linear Problems

In linear problems, all the outputs are simple linear functions of the inputs, as in y=mx+b. When problems only use simple arithmetic operations such as addition, subtraction, and Excel functions such as TREND() and FORCAST() it indicates there are purely linear relationships between the variables.

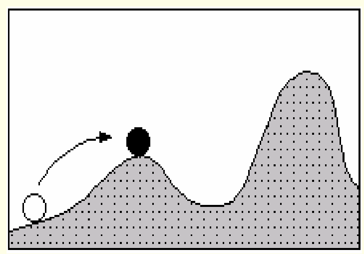
Linear problems have been fairly easy to solve since the advent of computers and the invention by George Dantzig of the Simplex Method. A simple linear problem can be solved most quickly and accurately with a linear programming utility. The Solver utility included with Excel becomes a linear programming tool when you set the "Assume Linear Model" checkbox.Solver then uses a linear programming routine to quickly find the perfect solution. If your problem can be expressed in purely linear terms, you should use linear programming. Unfortunately, most real-world problems cannot be described linearly.

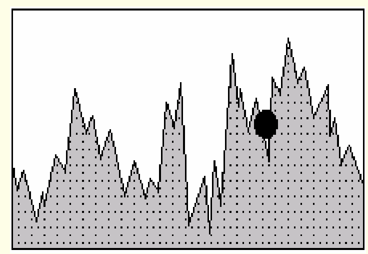
Non-linear Problems

If the cost to manufacture and ship out 5,000 widgets was $5,000, would it cost $1 to manufacture and ship 1 widget? Probably not. The assembly line in the widget factory would still consume energy, the paperwork would still need to be filled out and processed through the various departments, the materials would still be bought in bulk, the trucks would require the same amount of gas to deliver the widgets, and the truck driver would still get paid a full day's salary no matter how full the load was. Most real-world problems do not involve variables with simple linear relationships. These problems involve multiplication, division, exponents, and built-in Excel functions such as SQRT() and GROWTH(). Whenever the variables share a disproportional relationship to one another, the problem becomes non-linear.

If we simply need to find the minimum level of reactants that will give us the highest rate of reaction, we can just start anywhere on the graph and climb along the curve until we reach the top. This method of finding an answer is called hill climbing.

Hill climbing will always find the best answer if a) the function being explored is smooth, and b) the initial variable values place you on the side of the highest mountain. If either condition is not met, hill climbing can end up in a local solution, rather than the global solution.





Highly non-linear problems, the kind often seen in practice, have many possible solutions across a complicated landscape. If a problem has many variables, and/or if the formulas involved are very noisy or curvy, the best answer will probably not be found with hill climbing, even after trying hundreds of times with different starting points. Most likely, a sub-optimal, and extremely local solution will be found (see figure below).

Evolutionary Systems?

Compromise between
Local versus **Global** search strategies.

Combinatorial problems

There is a large class of problems that are very different from the numerical problems examined so far. Problems where the outputs involve changing the order of existing input variables, or grouping subsets of the inputs are called combinatorial problems. These problems are usually very hard to solve, because they often require exponential time; that is, the amount of time needed to solve a problem with 4 variables might be 4 x 3 x 2 x 1, and doubling the number of variables to 8 raises the solving time to 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1, or a factor of 1,680. The number of variables doubles, but the number of possible solutions that must be checked increases 1,680 times. For example, choosing the starting lineup for a baseball team is a combinatorial problem. For 9 players, you can choose one out of the 9 as the first batter. You can then choose one out of the remaining 8 as the second batter, one of the remaining 7 will be the third, and so on. There are thus 9x8x7x6x5x4x3x2x1 (9 factorial) ways to choose a lineup of 9 players. This is about ***362,880*** different orderings. Now if you double the number of players, there are 18 factorial possible lineups, or ***6,402,373,705,000,000*** possible lineups!

Optimization Using Excel Solver

Solve the following two questions using Excel Solver

Question 1: A concrete manufacturer is concerned about how many units of two types of concrete elements should be produced during the next time period to maximize profit. Each concrete element of type I generates a profit of $60, while each element of type II generates a profit of $40. 2 and 3 units of raw materials are needed to produce one concrete element of type I and II, respectively. Also, 4 and 2 units of time are required to produce one concrete element of type I and II, respectively.

If 100 units of raw materials and 120 units of time are available, how many units of each type of concrete element should be produced to maximize profit and satisfy all constraints? Use Excel solver for the solution.

Question 2: A building contractor produces two types of houses: detached and semidetached. The customer is offered several choices of architectural design and layout for each type. The proportion of each type of design sold in the past is shown in the following table. The profit on a detached house and a semidetached house is $1,000 and $800 respectively.

Design	Detached	Semidetached
Type A	0.1	0.33
Type B	0.4	0.67
Type C	0.5	-----

The builder has the capacity to build 400 houses per year. However, an estate of housing will not be allowed to contain more than 75% of the total housing as detached. Furthermore, because of the limited supply of bricks available for type B designs, a 200-house limit with this design is imposed. Use Excel to develop a model of this problem and then use SOLVER to determine how many detached and semidetached houses should be constructed in order to maximize profits. State the optimum profit.

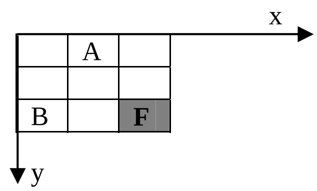
Example on Genetic Algorithms

Problem: A square construction site is divided into 9 grid units. We need to use GAs to determine the best location of two temporary facilities **A** and **B**, so that:

- Facility A is as close as possible to facility B.
- Facility A is as close as possible to the fixed facility F.
- Facility B is as far as possible to the fixed facility F.

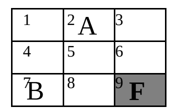
Step 1: **Problem Representation (how to define a facility location)**

Option 1
Using coordinates



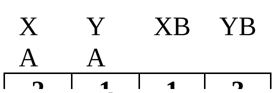
A has X = 2 and Y = 1
B has X = 1 and Y = 3

Option 2
Using location index



A is in Location index 2
B is in Location index 7

Step 2: **Chromosome Structure**
The variables in our problem are the locations of facilities A & B. Then, the chromosome structure for each of the two options in Step 1 are as follows. Note that the genes of a chromosome are the variables.



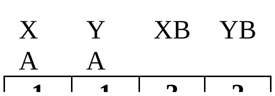
4 Genes
(Values range from 1 to 3)

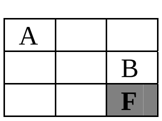
Index A	Index B
2	**7**

2 Genes
(Values range from 1 to 8)

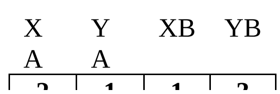
Step 3: **Generate Population (50 to 100 is reasonable diversity & processing time)**
(note: for this exercise, let's consider option 1 representation and a population of 3)

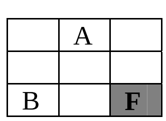
P1



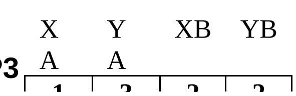


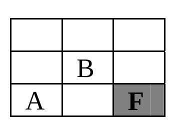
P2





P3





Step 4: Evaluate the Population

P1

A		
		B
		F

P2

	A	
B		**F**

P3

	B	
A		**F**

Objective function = Minimize site score = Minimum of $\sum d \,.\, W$

$$\textbf{Score} = d_{AB} \,.\, W_{AB} + d_{AF} \,.\, W_{AF} + d_{BF} \,.\, W_{BF}$$

Let's consider the closeness weights (W) as follows (from past notes):

$W_{AB} = 100$ (positive means A & B close to each other)
$W_{AF} = 100$ (A & F close to each other)
$W_{BF} = -100$ (negative means B & F far from each other)

Let's also consider the distance (d) between two facilities as the number of horizontal and vertical blocks between them.

P1 Score = $d_{AB} \,.\, W_{AB} + d_{AF} \,.\, W_{AF} + d_{BF} \,.\, W_{BF}$ = 3 . 100 + 4 . 100 + 1 . -100 = **600**
P2 Score = $d_{AB} \,.\, W_{AB} + d_{AF} \,.\, W_{AF} + d_{BF} \,.\, W_{BF}$ = 3 . 100 + 3 . 100 + 2 . -100 = **400**
P3 Score = $d_{AB} \,.\, W_{AB} + d_{AF} \,.\, W_{AF} + d_{BF} \,.\, W_{BF}$ = 2 . 100 + 2 .100 + 2 . -100 = **200**

Step 5: Calculate the Merits of Population Members

Merit of **P1** = (600+400+200) / 600 = **2**
Merit of **P2** = (600+400+200) / 400 = **3**
Merit of **P1** = (600+400+200) / 200 = **6** Notice the sum of merits = 11

Notice that smaller score gives higher merit because we are interested in minimization. In case of maximization, we use the inverse of the merit calculation.

Step 6: Calculate the Relative Merits of Population Members

RM of **P1** = merit * 100 / Sum of merits = 2 . 100 / 11 = **18**
RM of **P2** = merit * 100 / Sum of merits = 3 . 100 / 11 = **27**
RM of **P3** = merit * 100 / Sum of merits = 6 . 100 / 11 = **55**

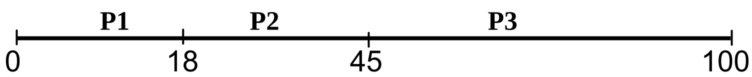
Step 7: Randomly Select Operator (Crossover or Mutation)

Crossover rate = 96% (marriage is the main avenue for evolution)
Mutation rate = 4% (genius people are very rare)

To select which operator to use in current cycle, we generate a random number (from 0 to 100). If the value is between 0 to 96, then crossover, otherwise, mutation.

Step 8: Use the Selected Operator (Assume Crossover)

8.a) Randomly select two parents according to their relative merits of Step 6

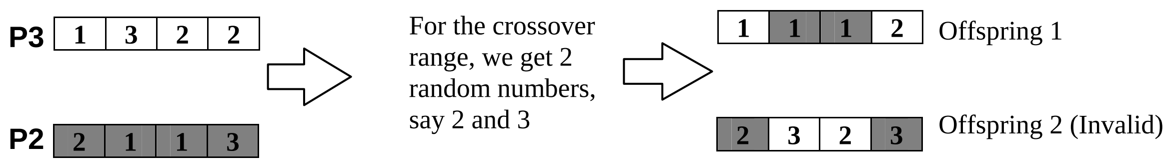


Relative merits on a cumulative horizontal scale.

For first parent, we generate a random number (0 to 100). According to its value, we pick the parent. For example, assume value is 76, then **P3** is selected.

For the 2nd parent, get a random number (0 to 100). Assume 39, Then **P2** is picked.

8.b) Let's apply crossover to generate an offspring



<u>Step 9:</u> Evaluate the Offspring

Offspring 1

A		
B		
		F

Notice that Offspring 2 is invalid because both facilities A & B are at same coordinates (x = 2 and Y = 3) and this is not allowed

Offspring Score = $\mathbf{d_{AB}.W_{AB}} + \mathbf{d_{AF}.W_{AF}} + \mathbf{d_{BF}.W_{BF}} = 1.100 + 4.100 + 3. -100 =$ **200**

<u>Step 10:</u> Compare the Offspring with the Population (Evolve the Population)

Since the offspring score = **200** is better than the worst population member (P1 has a score of 600), then the offspring survives and P1 dies (will be replaced by the offspring).

Accordingly, **P1** becomes:

1	**1**	**1**	**2**

At the end of this step, we **<u>GOTO STEP 4</u>** , repeating the process thousands of times until the best solution is determined. One of the top solutions is as follows:

Score = 0

		B
		A
		F

Comparison among five evolutionary-based optimization algorithms

by
Emad Elbeltagi; Tarek Hegazy; and Donald Grierson

ABSTRACT: Evolutionary algorithms are stochastic search methods that mimic the natural biological evolution and/or the social behavior of species. Such algorithms have been developed to arrive at near-optimum solutions to large-scale optimization problems, for which traditional mathematical techniques may fail. This paper compares the formulation and results of five recent evolutionary-based algorithms: genetic algorithms, memetic algorithms, particle swarm, ant colony systems, and shuffled frog leaping. A brief description of each algorithm is presented along with a pseudocode to facilitate the implementation and use of such algorithms by researchers and practitioners. Benchmark comparisons among the algorithms are presented for both continuous and discrete optimization problems, in terms of processing time, convergence speed, and quality of the results. Based on this comparative analysis, the performance of evolutionary algorithms is discussed along with some guidelines for determining the best operators for each algorithm. The study presents sophisticated ideas in a simplified form that should be ~~is~~ beneficial to both practitioners and researchers involved in solving optimization problems.

1. Introduction

The difficulties associated with using mathematical optimization on large scale engineering problems have contributed to the development of alternative solutions. Linear programming and dynamic programming techniques, for example, often fail (or reach local optimum) in solving NP-hard problems with large number of variables and non-linear objective functions [1]. To overcome these problems, researchers have proposed evolutionary-based algorithms for searching near-optimum solutions to problems.

Evolutionary Algorithms are stochastic search methods that mimic the metaphor of natural biological evolution and/or the social behavior of species. Examples include how ants find the shortest route to a source of food and how birds find their destination during migration. The behaviour of such species is guided by learning, adaptation, and evolution [1]. To mimic the efficient behaviour of these species, various researchers have developed computational systems that seek fast and robust solutions to complex optimization problems. The first evolutionary-based technique introduced in the literature was the Genetic Algorithms, [2]. Genetic Algorithms (GAs) were developed based on the Darwinian principle of the "survival of the fittest" and the natural process of evolution through reproduction. Based on its demonstrated ability to reach near-optimum solutions to large problems, the GAs technique has been used in many applications in science and engineering [e.g.,3,4,5]. Despite their benefits, GAs may require long processing time for a near-optimum solution to evolve. Also, not all problems lend themselves well to a solution with GAs [6].

In an attempt to reduce processing time and improve the quality of solutions, particularly to avoid being trapped in local optima, other Evolutionary Algorithms (EAs) have been introduced during the past 10 years. In addition to various GA improvements, recent developments in EAs include four other techniques inspired by different natural processes: memetic algorithms [7], particle swarm optimization [8], ant colony systems [9], and shuffled frog leaping [10]. A schematic diagram of the natural processes that the five algorithms mimic is shown in Fig. 1.

In this paper, the five EAs presented in Fig. 1 are reviewed and a pseudocode for each algorithm is presented to facilitate its implementation. Performance comparison among the five algorithms is then presented. Guidelines are then presented for determining the proper parameters to use with each algorithm.

2. Five evolutionary algorithms

In general, EAs share a common approach for their application to a given problem The problem first requires some representation to suit each method. Then, the evolutionary search algorithm is applied iteratively to arrive at a near-optimum solution. A brief description of the five algorithms is presented in the following subsections.

2.1. Genetic algorithms

Genetic algorithms (GAs) are inspired by biological systems' improved fitness through evolution [2]. A solution to a given problem is represented in the form of a string, called "chromosome", consisting of a set of elements, called "genes", that hold a set of values for the optimization variables [11].

GAs work with a random population of solutions (chromosomes). The fitness of each chromosome is determined by evaluating it against an objective function. To simulate the natural "survival of the fittest" process, best chromosomes exchange information (through crossover or mutation) to produce offspring chromosomes. The offspring solutions are then evaluated and used to evolve the population if they provide better solutions than weak population members. Usually, the process is continued for a large number of generations to obtain a best-fit (near-optimum) solution. More details on the mechanism of GAs can be found in Goldberg [11] and Al-Tabtabai and Alex [3].

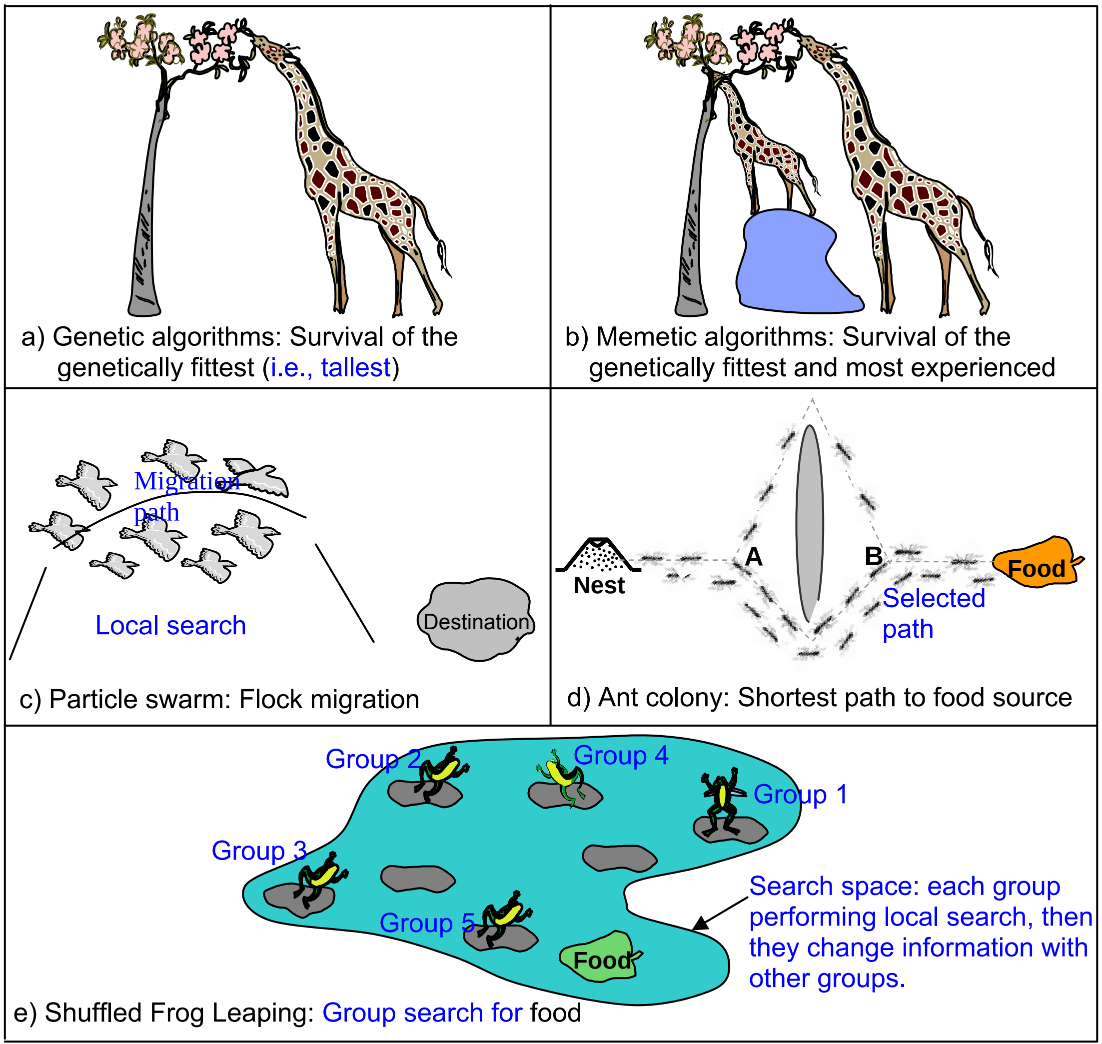


Fig. 1. Schematic diagram of natural evolutionary systems

A pseudocode for the GAs algorithm is shown in Appendix I. Four main parameters affect the performance of GAs: population size, number of generations, crossover rate, and mutation rate. Larger population size (i.e., hundreds of chromosomes) and large number of generations (thousands) increase the likelihood of obtaining a global optimum solution, but substantially increase processing time.

Crossover among parent chromosomes is a common natural process [12] and traditionally is given a rate that ranges from 0.6 to 1.0. In crossover, the exchange of parents' information produces an offspring, as shown in Fig. 2. As opposed to crossover, mutation is a rare process that resembles a sudden change to an offspring. This can be done by randomly selecting one chromosome from the population and then arbitrarily changing some of its information. The benefit of mutation is that it randomly introduces new genetic material to the evolutionary process, perhaps thereby avoiding stagnation around local minima. A small mutation rate less than 0.1 is usually used [11].

The GA used in this study is steady state (an offspring replaces the worst chromosome only if is better than it) and real coded (the variables are represented in real numbers). The main parameters used in the GA procedure are population size, number of generations, crossover rate and mutation rate.

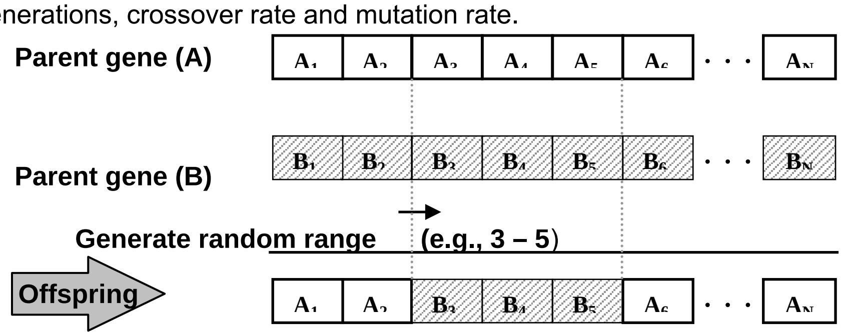


Fig. 2. Crossover operation to generate offspring

2.2. Memetic algorithms

Memetic algorithms (MAs) are inspired by Dawkins' notion of a *meme* [13]. MAs are similar to GAs but the elements that form a chromosome are called memes, not genes. The unique aspect of the MAs algorithm is that all chromosomes and offsprings are allowed to gain some experience, through a local search, before being involved in the evolutionary process [14]. As such, the term MAs is used to describe GAs that heavily use local search [15]. A pseudocode for a MA procedure is given in Appendix II.

Similar to the GAs, an initial population is created at random. Afterwards, a local search is performed on each population member to improve its experience and thus obtain a population of local optimum solutions. Then, crossover and mutation operators are applied, similar to GAs, to produce offsprings. These offsprings are then subjected to the local search so that local optimality is always maintained.

Merz and Freisleben [14] proposed one approach to perform local search through a pair-wise interchange heuristic (Fig. 3). In this method, the local search neighborhood is defined as the set of all solutions that can be reached from the current solution by swapping two elements (memes) in the chromosome. For a chromosome of length *n*, the neighborhood size for the local search *i*:

$$N = \tfrac{1}{2} \, . \, n \, . \, (n - 1) \qquad (1)$$

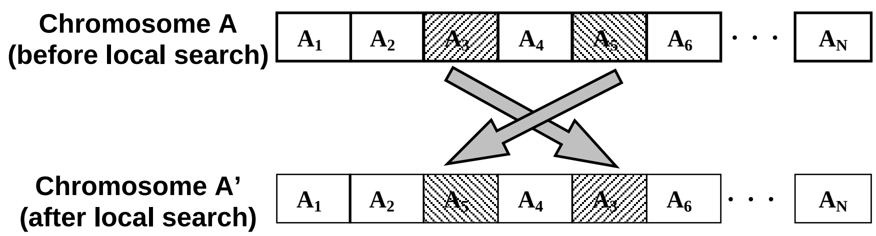


Fig. 3. Applying local search using pair-wise interchange

The number of swaps and consequently the size of the neighborhood grow quadratically with the chromosome length (problem variables). In order to reduce processing time, Merz and Freisleben [14] suggested stopping the pair-wise interchange after performing the first swap that enhances the objective function of the current chromosome. The local search algorithm, however, can be designed to suit the problem nature. For example, another local search can be conducted by adding or subtracting an incremental value from every gene and testing the chromosome's performance. The change is kept if the chromosome's performance improves; otherwise, the change is ignored. A pseudocode of this modified local search is given in Appendix III. As discussed, the parameters involved in MAs are the same four parameters used in GAs: population size, number of generations, crossover rate, and mutation rate in addition to a local search.

2.3. Particle swarm optimization

Particle swarm optimization (PSO) was developed by Kennedy and Eberhart [8]. The PSO is inspired by the social behavior of a flock of migrating birds trying to reach an unknown destination. In PSO, each solution is a "bird" in the flock and is referred to as a "particle". A particle is analogous to a chromosome (population member) in GAs. As opposed to GAs, the evolutionary process in the PSO doesn't create new birds from parent ones. Rather, the birds in the population only evolve their social behavior and accordingly their movement towards a destination [16].

Physically, this mimics a flock of birds that communicate together as they fly. Each bird looks in a specific direction, and then when communicating together, they identify the bird that is in the best location. Accordingly, each bird speeds towards the best bird using a velocity that depends on its current position. Each bird, then, investigates the search space from its new local position, and the process repeats until the flock reaches a desired destination. It is important to note that the process involves both social interaction and intelligence so that birds learn from their own experience (local search) and also from the experience of others around them (global search).

The pseudocode for the PSO is shown in Appendix IV. The process is initialized with a group of random particles (solutions), *N*. The *i*th particle is represented by its position as a point in a *S*-dimensional space, where *S* is the number of variables. Throughout the process, each particle *i* monitors three values: its current position (X_i); the best position it reached in previous cycles (P_i); and its flying velocity (V_i). These three values are represented as follows:

$$\left.\begin{array}{ll} \textit{Current position} & X_i = (x_{i1}, x_{i2}, \ldots, x_{iS}) \\ \textit{Best previous position} & P_i = (p_{i1}, p_{i2}, \ldots, p_{iS}) \\ \textit{Flying velocity} & V_i = (v_{i1}, v_{i2}, \ldots, v_{iS}) \end{array}\right\} \qquad (2)$$

In each time interval (cycle), the position (P_g) of the best particle (g) is calculated as the best fitness of all particles. Accordingly, each particle updates its velocity V_i to catch up with the best particle g, as follows [16]:

$$New\ V_i = \omega \ .\ current\ V_i + c_1\ .\ rand() \times (P_i - X_i) + c_2\ .\ Rand() \times (P_g - X_i) \qquad (3)$$

As such, using the new velocity V_i, the particle's updated position becomes:

$$New\ position\ X_i = current\ position\ X_i + New\ V_i\ ; \qquad V_{max} \geq V_i \geq -V_{max} \qquad (4)$$

where c_1 and c_2 are two positive constants named learning factors (usually $c_1 = c_2 = 2$); *rand()* and *Rand()* are two random functions in the range [0, 1], V_{max} is an upper limit on the maximum change of particle velocity [8], and ω is an inertia weight employed as an improvement proposed by Shi and Eberhart [16] to control the impact of the previous history of velocities on the current velocity. The operator ω plays the role of balancing the global search and the local search; and was proposed to decrease linearly with time from a value of 1.4 to 0.5 [16]. As such, global search starts with a large weight and then decreases with time to favor local search over global search [17].

It is noted that the second term in Eq. 3 represents "*cognition*", or the private thinking of the particle when comparing its current position to its own best. The third term in Eq. 3, on the other hand, represents the "*social*" collaboration among the particles, which compares a particle's current position to that of the best particle [18]. Also, to control the change of particles' velocities, upper and lower bounds for velocity change is limited to a user-specified value of V_{max}. Once the new position of a particle is calculated using Eq. 4, the particle, then, flies towards it [16]. As such, the main parameters used in the PSO technique are: the population size (number of birds); number of generation cycles; the maximum change of a particle velocity V_{max}; and ω.

2.4. Ant colony optimization

Similar to PSO, Ant colony optimization (ACO) Algorithms evolve not in their genetics but in their social behavior. ACO was developed by Dorigo et al. [9] based on the fact that ants are able to find the shortest route between their nest and a source of food. This is done using pheromone trails, which ants deposit whenever they travel, as a form of indirect communication.

As shown in Fig. 1-d, when ants leave their nest to search for a food source, they randomly rotate around an obstacle, and initially the pheromone deposits will be the same for the right and left directions. When the ants in the shorter direction find a food source, they carry the food and start returning back, following their pheromone trails, and still depositing more pheromone. As indicated in Fig. 1-d, an ant will most likely choose the shortest path when returning back to the nest with food as this path will have the most deposited pheromone. For the same reason, new ants that later starts out from the nest to find food will also choose the shortest path. Over time, this positive feedback (autocatalytic) process prompts all ants to choose the shorter path [19].

Implementing the ACO for a certain problem requires a representation of S variables for each ant, with each variable i has a set of n_i options with their values *lij,* and their associated pheromone concentrations $\{\tau_{ij}\}$; where i = 1, 2,….S, and j = 1, 2,…n_i. As such, an ant is consisted of S values that describe the path chosen by the ant as shown in Fig. 4 [20]. A pseudocode for the ACO is shown in Appendix V. Other researchers use a variation of this general algorithm, incorporating a local search to improve the solution [21].

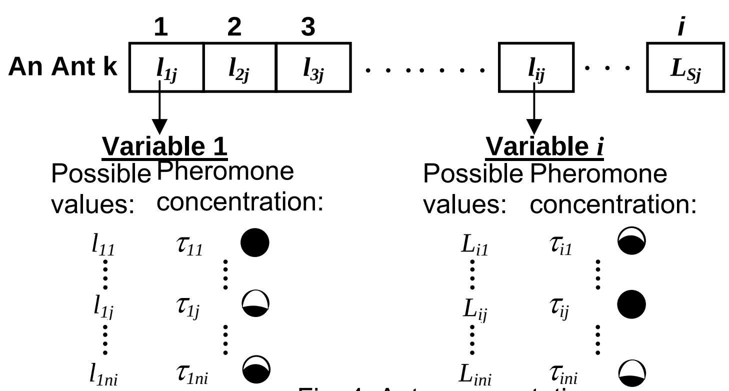


Fig. 4. Ant representation

In the ACO, The process starts by generating m random ants (solutions). An ant k (k=1, 2, …., m) represents a solution string, with a selected value for each variable. Each ant is then evaluated according to an objective function. Accordingly, pheromone concentration associated with each possible route (variable value) is changed in a way to reinforce good solutions, as follows [9]:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \Delta\tau_{ij} \quad ;\ t = 1, 2, \ldots, T \quad (5)$$

where T is the number of iterations (generation cycles); $\tau_{ij}(t)$ is the revised concentration of pheromone associated with option l_{ij} at iteration t; $\tau_{ij}(t-1)$ is the concentration of pheromone at the previous iteration ($t-1$); $\Delta\tau_{ij}$ = change in pheromone concentration; and ρ = pheromone evaporation rate (0 to 1). The reason for allowing pheromone evaporation is to avoid too strong influence of the old pheromone to avoid premature solution stagnation [22]. In Eq. 5, the change in pheromone concentration $\Delta\tau_{ij}$ is calculated as [9]:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \begin{cases} R / fitness_k & \text{if option } l_{ij} \text{ is chosen by ant } k \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where R is a constant called the pheromone reward factor; and $fitness_k$ is the value of the objective function (solution performance) calculated for ant k. It is noted that the amount of pheromone gets higher as the solution improves. Therefore, for minimization problems, Eq. 6 shows the pheromone change as proportional to the inverse of the fitness. In maximization problems, on the other hand, the fitness value itself can be directly used.

Once the pheromone is updated after an iteration, the next iteration starts by changing the ants' paths (i.e., associated variable values) in a manner that respects pheromone concentration and also some heuristic preference. As such, an ant k at iteration t will change the value for each variable according to the following probability [9]:

$$P_{ij}(k,t) = \frac{[\tau_{ij}(t)]^{\alpha} \times [\eta_{ij}]^{\beta}}{\sum_{l_{ij}} [\tau_{ij}(t)]^{\alpha} \times [\eta_{ij}]^{\beta}} \quad (7)$$

where $P_{ij}(k,t)$ = probability that option l_{ij} is chosen by ant k for variable i at iteration t; $\tau_{ij}(t)$ = pheromone concentration associated with option l_{ij} at iteration t; η_{ij} = heuristic factor for preferring among available options and is an indicator of how good it is for ant k to select option l_{ij} (this heuristic factor is generated by some problem characteristics and its value is fixed for each option l_{ij}); and α and β are exponent parameters that control the relative importance of pheromone concentration versus the heuristic factor [20]. Both α and β can take values greater than zero and can be determined by trial and error. Based on the previous discussion, the main parameters involved in ACO are: number of ants m; number of iterations t; exponents α and β; pheromone evaporation rate ρ; and pheromone reward factor R.

2.5. Shuffled frog leaping algorithm

The shuffled frog leaping (SFL) algorithm, in essence, combines the benefits of the genetic-based memetic algorithms and the social behavior-based particle swarm optimization algorithms. In the SFL, the population consists of a set of frogs (solutions) that is partitioned into subsets referred to as memeplexes. The different memeplexes are considered as different cultures of frogs, each performing a local search. Within each memeplex, the individual frogs hold ideas, that can be influenced by the ideas of other frogs, and evolve through a process of memetic evolution. After a defined number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process [23]. The local search and the shuffling processes continue until defined convergence criteria are satisfied [10].

As described in the pseudocode of Appendix VI, an initial population of "*P*" frogs is created randomly. For *S*-dimensional problems (*S* variables), a frog *i* is represented as $X_i = (x_{i1}, x_{i2}, \ldots\ldots, x_{iS})$. Afterwards, the frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into *m* memeplexes, each containing *n* frogs (i.e., $P = m \times n$). In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the mth memeplex, and frog m+1 goes back to the first memeplex, etc.

Within each memeplex, the frogs with the best and the worst fitnesses are identified as X_b and X_w, respectively. Also, the frog with the global best fitness is identified as X_g. Then, a process similar to PSO is applied to improve only the frog with the worst fitness (not all frogs) in each cycle. Accordingly, the position of the frog with the worst fitness is adjusted as follows:

Change in frog position $(D_i) = rand() \cdot (X_b - X_w)$ (8)

New position X_w = *current position* $X_w + D_i$; $D_{max} \geq D_i \geq -D_{max}$ (9)

where *rand()* is a random number between 0 and 1; and D_{max} is the maximum allowed change in a frog's position. If this process produces a better solution, it replaces the worst frog. Otherwise, the calculations in Eqs. 8 and 9 are repeated but with respect to the global best frog (i.e., X_g replaces X_b). If no improvement becomes possible in this case, then a new solution is randomly generated to replace that frog. The calculations then continue for a specific number of iterations [10]. Accordingly, the main parameters of SFL are: number of frogs *P*; number of memeplexes; number of generation for each memeplex before shuffling; number of shuffling iterations; and maximum step size.

3. Comparison among evolutionary algorithms' results

All the EAs described earlier have been coded using the Visual Basic programming language and all experiments took place on a 1.8 GHz AMD Laptop machine. The performance of the five evolutionary algorithms is compared using two benchmark problems for continuous optimization and a third problem for discrete optimization. A description of these test problems is given in the following.

3.1. Continuous optimization

Two well-known continuous optimization problems are used to test four of the EAs: *F8* (*Griewank's*) function and the *F10* function. Details of these functions are as follows:
F8 (Griewank's function): The objective function to be optimized is a scalable, non-linear, and non-separable function that may take any number of variables (x_i *s*), i.e.,

$$f(x_{i|i=1,N}) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} (\cos(x_i / \sqrt{i})) \qquad (10)$$

The summation term of the *F8* function (Eq. 10) includes a parabolic shape while the cosine function in the product term creates waves over the parabolic surface. These waves create local optima over the solution space [24]. The *F8* function can be scaled to any number of variables *N*. The values of each variable are constrained to a range (-512 to 511). The global optimum (minimum) solution for this function is known to be zero when all *N* variables equal zero.

F10 Function: This function is non-linear, non-separable, and involves two variables *x* and *y*., i.e.,

To scale this function (Eq. 11) to any number of variables, an extended *EF10* function is created using the following relation, [24],

$$f10(x, y) = (x^2 + y^2)^{0.25}[\sin^2(50(x^2 + y^2)^{0.1}) + 1] \qquad (11)$$

Accordingly, the extended *F10* function is:

$$EF(x_{i|i=N}) = \sum_{j=1}^{N} \sum_{i=1}^{N} F(x_i, x_j) \qquad (12)$$

$$EF10(x_{i|i=1,N}) = \sum_{i=1}^{N-1} (x_i^2 + x_{i+1}^2)^{0.25}[\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1] \qquad (13)$$

Similar to the *F8* function, the global optimum solution for this function is known to be zero when all *N* variables equal zero, for the variable values ranging from -100 to 100.

3.2. Discrete optimization

In this section, a time-cost trade-off (TCT) construction management problem is used to compare among the five EAs with respect to their ability to solve discrete optimization problems. The problem relates to an 18-activity construction project that was described in [25]. The activities, their predecessors, and durations are presented in Table 1 along with five optional methods of construction that vary from cheap and slow (option 5) to fast and expensive (option1). The 18 activities were input to a project management software (Microsoft Project) with activity durations being set to those of option 5 (least costs and longest durations among the five options). The total direct cost of the project in this case is $99,740 (sum of all activities' costs for option 5) with the project duration being 169 days (respecting the precedence relations in Table 1). The indirect cost of $500/day was then added to obtain a total project cost of $184,240.

With the initial schedule exceeding a desired deadline of 110-days, it is required to search for the optimum set of construction options that meet the deadline at minimum total cost. In this problem, the decision variables are the different methods of construction possible for each activity (i.e., five discrete options, 1 to 5, with associated durations and costs). The objective function is to minimize the total project cost (direct and indirect) and is formulated as follows:

$$Min \quad (T.I + \sum_{i=1}^{n} C_{ij}) \qquad (14)$$

where *n* = number of activities; C_{ij} = direct cost of activity *i* using its method of construction *j; T* = total project duration; and *I* = daily indirect cost. To facilitate the optimization using the different EAs, macro programs of the 5 EAs were written using the VBA language that comes with the Microsoft Project software. The data in Table 1 were stored in one of the tables associated with the software. When any one of the EA routines is activated, the evolutionary process selects one of the five construction options to set the activities' durations and costs. Accordingly, the project's total cost (objective function) and duration changes. The evolutionary process then continues to attempt to optimize the objective function.

Table 1: Test problem for discrete optimization

Activity No.	Depends On	Option 1		Option 2		Option 3		Option 4		Option 5	
		Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)	Duration (days)	Cost ($)
1	-	14	2 400	15	2 150	16	1 900	21	1 500	24	1 200
2	-	15	3 000	18	2 400	20	1 800	23	1 500	25	1 000
3	-	15	4 500	22	4 000	33	3 200	—	—	—	—
4	-	12	45 000	16	35 000	20	30 000	—	—	—	—
5	1	22	20 000	24	17 500	28	15 000	30	10 000	—	—
6	1	14	40 000	18	32 000	24	18 000	—	—	—	—
7	5	9	30 000	15	24 000	18	22 000	—	—	—	—
8	6	14	220	15	215	16	200	21	208	24	120
9	6	15	300	18	240	20	180	23	150	25	100
10	2, 6	15	450	22	400	33	320	—	—	—	—
11	7, 8	12	450	16	350	20	300	—	—	—	—
12	5, 9, 10	22	2 000	24	1 750	28	1 500	30	1 000	—	—
13	3	14	4 000	18	3 200	24	1 800	—	—	—	—
14	4, 10	9	3 000	15	2 400	18	2 200	—	—	—	—
15	12	12	4 500	16	3 500	—	—	—	—	—	—
16	13, 14	20	3 000	22	2 000	24	1 750	28	1 500	30	1 000
17	11, 14, 15	14	4 000	18	3 200	24	1 800	—	—	—	—
18	16, 17	9	3 000	15	2 400	18	2 200	—	—	—	—

3.3. Parameter settings for evolutionary algorithms

As discussed earlier, each algorithm has its own parameters that affect its performance in terms of solution quality and processing time. To obtain the most suitable parameter values that suit the test problems, a large number of experiments were conducted. For each algorithm, an initial setting of the parameters was established using values previously reported in the literature[Emad, list the source references here]. Then, the parameter values were changed one by one and the results were monitored in terms of the solution quality and speed. The final parameter values for the five EAs are:

Genetic Algorithms: The crossover probability (C_P) and the mutation probability (M_P) were set to 0.8 and 0.08, respectively. The population size was set at 200 and 500 offsprings. The evolutionary process was kept running until no improvements were made in the objective function for 10 consecutive generation cycles (i.e., 500 * 10 offsprings or the objective function reached its known target value, whichever comes first.

Memetic Algorithms: MAs are similar to GAs but apply local search on chromosomes and offsprings. The standard pair-wise interchange search does not suit the continuous functions *F8* and *F10*, and the local search procedure in Appendix III is used instead. For the discrete problem, on the other hand, the pair-wise interchange was used. The same values of C_P = 0.8 and M_P = 0.08 that were used for the GAs are applied to the MAs. After experimenting with various values, a population size of 100 chromosomes was used for the MAs.

Particle Swarm Optimization: Upon experimentation, the suitable numbers of particles and generations were found to be 40 and 10000, respectively. Also, the maximum velocity was set as 20 for the continuous problems and 2 for the discrete problem. The inertia weight factor ω was also set as a time-variant linear function decreasing with the increase of number of generations where, at any generation *i,*

$$\omega = 0.4 + 0.8 * (\text{number of generations} - i) / (\text{number of generations} - 1) \qquad (15)$$

such that ω =1.2 and 0.4 at the first and last generation, respectively.

Ant Colony Optimization: As the ACO algorithm is suited to discrete problems alone, no experiments were done using it for the *F8* and *F10* test functions. However, the TCT discrete problem was used for experimentation with the ACO. After extensive experimentation, 30 ants and 100 iterations were found suitable. Also, the other parameters were set as follows: α = 0.5; β = 2.5; □ (pheromone evaporation rate) = 0.4; and *R* (reward factor depends on problem nature) = 10.

Shuffled Frog Leaping: Different settings were experimented with to determine suitable values for parameters to solve the test problems using the SFL algorithm. A population of 200 frogs, 20 memeplexes, and 10 iterations per memeplex were found suitable to obtain good solutions.

3.4. Results and discussions

The results found from solving the three test problems using the five evolutionary algorithms, which represents a fairly wide class of problems, are summarized in Tables 2 and 3, and Fig. 5 (the Y axis of Fig. 5 is a log scale to show long computer run times). It is noted that the processing time for solving the *EF10* function was similar to that of the *F8* function and follows the same trend as shown in Fig. 5.

Twenty trial runs were performed for each problem. The performance of the different algorithms was compared using three criteria: 1) the percentage of success, as represented by the number of trials required for the objective function to reach its known target value; 2) the average value of the solution obtained in all trials; and 3) the processing time to reach the optimum target value. The processing time, and not the number of generation cycles, was used to measure the speed of each EA because the number of generations in each evolutionary cycle is different from one algorithm to another. In all experiments, the solution stopped when one of two following criteria was satisfied: 1) the *F8* and *EF10* objective functions reached a target value of 0.05 or less (i.e., to within an acceptable tolerance of the known optimum value of zero), or 110 days for the TCT problem; or 2) the objective function value did not improve in ten consecutive generations. To experiment with different problem sizes, the *F8* test function in Eq. (10) was solved using 10, 20, 50, and 100 variables, while the *EF10* test function in Eq. (13) was solved using 10, 20, and 50 variables (it becomes too complex for larger numbers of variables).

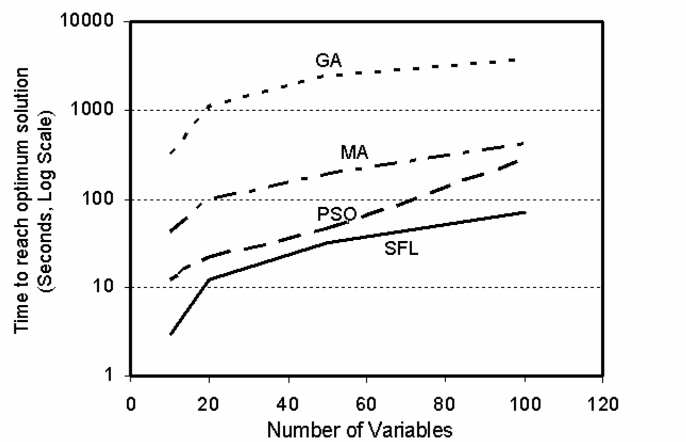


Fig. 5. Processing time to reach the optimum for *F8* function

Table 2 - Results of the continuous optimization problems

Comparison criteria	Algorithm	Number of variables						
		F8				*EF10*		
		10	20	50	100	10	20	50
% Success	GAs (Evolver)	50	30	10	0	20	0	0
	MAs	90	100	100	100	100	70	0
	PSO	30	80	100	100	100	80	60
	ACO	-	-	-	-	-	-	-
	SFL	50	70	90	100	80	20	0
Mean solution	GAs (Evolver)	0.060	0.097	0.161	0.432	0.455	1.128	5.951
	MAs	0.014	0.013	0.011	0.009	0.014	0.068	0.552
	PSO	0.093	0.081	0.011	0.011	0.009	0.075	2.895
	ACO	-	-	-	-	-	-	-
	SFL	0.080	0.063	0.049	0.019	0.058	2.252	6.469

Table 3 - Results of the discrete optimization problem

Algorithm	Minimum Project Duration (days)	Average Project Duration (days)	Minimum Cost ($)	Average Cost ($)	% Success Rate	Processing Time (second)
GAs	113	120	162 270	164 772	0	16
MAs	110	114	161 270	162 495	20	21
PSO	110	112	161 270	161 940	60	15
ACO	110	122	161 270	166 675	20	10
SFL	112	123	162 020	166 045	0	15

Surprisingly, the GA performed more poorly than all the other four algorithms. In fact, it was found to perform more poorly than even that reported in Whitley et al. [24] and Raphael and Smith [26] when using the CHC and Genitor genetic algorithms, while it performed better than the ESGAT genetic algorithm version. A commercial GA package, Evolver [27], was used to verify the results. Evolver is an add-in program to Microsoft Excel, where the objective function, variables (adjustable cells), and the constraints are readily specified by highlighting the corresponding spreadsheet cells. Evolver performed almost the same way as the VB code with slight improvement. The results of using Evolver are reported in Table 2. The difference in Evolver's results compared to those of the other EA algorithms may in part be because Evolver uses binary rather than real coding.

As shown in Table 2 for the *F8* function, the GA was able to reach the target for 50% of the trials with 10 variables, and the number of successes decreased as the number of variables increased. Despite its inability to reach the optimum value of zero with the larger number of 100 variables, the GA was able to achieve a solution close to the optimum (0.432 for the *F8* function with 100 variables). Also, it is noticed from Fig 5 that as the number of variables increased, the processing time to reach the target also increased (from 5min:12sec with 10 variables to 40min:27sec with 50 variables). As shown in Table 2 for the *EF10* test function, the GA was only able to achieve 20% success using 10 variables, and that the solution quality decreased as the number of variables increased (e.g., the objective function = 5.951 using 50 variables). Using the GA to solve the TCT problem, the minimum solution obtained was 113 days with a minimum total cost of $162,270 and the success rate for reaching the optimum solution was zero, as shown in Table 3.

Upon applying the MA, the results improved significantly compared to those obtained using the GA, in terms of both the success rate (Table 2) and the processing time (Fig. 5). Solving the *F8* function using 100 variables, for example, the success rate was 100% with a processing time of 7min:08 sec. Even for the trials with less success rate, as shown in Table 2, the solutions were very close to the optimum. That is to say, the local search of the MA improved upon the performance of the GA. When applying the MA to the TCT problem, it was able to reach the optimum project duration of 110 days and a total cost of $161,270, with a 20% success rate and an average cost that improved upon that of the GA (Table 3). It is to be noted that the local search module presented in Appendix III was applied for the *F8* and *EF8* functions, while the pair-wise interchange local search module was applied to the TCT problem.

The PSO algorithm outperformed the GA and the MA in solving the *EF10* function in terms of the success rate (Table 2), the processing time (Fig. 5), while it was less successful than the MA in solving the *F8* function. Also, the PSO algorithm outperformed all other algorithms when used to solve the TCT problem, with a success rate of 60% and average total cost of $161,940, as shown in Table 3.

The ACO algorithm was applied only to the TCT discrete optimization problem. While it was able to achieve the same success rate as the GA (20%), the average total cost of the 20 runs was greater than that of all other algorithms (Table 3). This is due to the scattered nature of the obtained results (minimum duration of 110 days, and maximum duration of 139 days) caused by premature convergence that happened in some runs. To avoid premature convergence, the pair-wise inter-change local search module was applied and the results obtained were greatly improved with a success rate of 100%, but the average processing time increased from 10 to 48 seconds.
When solving the *F8* and *EF10* test functions using the SFL algorithm, it was found that the success rate (Table 2) was better than the GA and similar to that for PSO. However, it performed less well when used to solve the *EF10* function. As shown in Fig. 5, the SFL processing times were the least among all algorithms. Interestingly, it is noticed from Table 2 that as the number of variables increased for the *F8* function, the success rates for SFL, MA and PSO all increased. This is because the *F8* function becomes smoother as its dimensions increase [2]. As opposed to this trend, the success rate decreased for the GA as the number of variables increased. The same trend for the GA was also reported in [24] and [26] when used to solve the *F8* function. Also, using the SFL algorithm to solve the TCT problem, the minimum duration obtained was 112 days with minimum total cost of $162,020 (Table 3). While the success rate for the SFL was zero, its performance was better than the GA.

It is interesting to observe that the behavior of each optimization algorithm in all test problems (continuous and discrete) was consistent. In particular, the PSO algorithm generally outperformed all other algorithms in solving all the test problems in terms of solution quality (except for the F8 function with 10 and 50 variables). Accordingly, it can be concluded that the PSO is a promising optimization tool, in part due to the effect of the inertia weight factor ω. In fact, to take advantage of the fast speed of the SFL algorithm, the authors suggest using a weight factor in Eq. (3) for SFL that is similar to that used for PSO (some preliminary experiments conducted by the authors in this regard have shown good results).

4. Conclusions

In this paper, five evolutionary-based search methods were presented. These include: genetic algorithm (GA), memetic algorithm (MA), particle swarm optimization (PSO), ant colony optimization (ACO), and shuffled frog leaping (SFL). A brief description of each method is presented along with a pseudocode to facilitate their implementation. Visual Basic programs were written to implement each algorithm. Two benchmark continuous optimization test problems were solved using all but the ACO algorithm, and the comparative results were presented. Also presented were the comparative results found when a discrete optimization test problem was solved using all five algorithms. The PSO method was generally found to perform better than other algorithms in terms of success rate and solution quality, while being second best in terms of processing time.

Appendix I. Pseudocode for a GA Procedure

```
Begin;
    Generate random population of P solutions (chromosomes);
    For each individual i є P: calculate fitness (i);
        For i = 1 to number of generations;
                Randomly select an operation (crossover or mutation);
                If crossover;
                        Select two parents at random ia and ib;
                        Generate on offspring ic = crossover (ia and ib);
                Else If mutation;
                        Select one chromosome i at random;
                        Generate an offspring ic = mutate (i);
                End if;
                Calculate the fitness of the offspring ic;
                If ic is better than the worst chromosome then replace the worst chromosome by ic;
        Next i;
    Check if termination = true;
End;
```

Appendix II. Pseudocode for a MA Procedure

```
Begin;
    Generate random population of P solutions (chromosomes);
    For each individual i є P: calculate fitness (i);
    For each individual i є P: do local-search (i);
        For i = 1 to number of generations;
                Randomly select an operation (crossover or mutation);
                If crossover;
                        Select two parents at random ia and ib;
                        Generate on offspring ic = crossover (ia and ib);
                        ic = local-search (ic);
                Else If mutation;
                        Select one chromosome i at random;
                        Generate an offspring ic = mutate (i);
                        ic = local-search (ic);
                End if;
                Calculate the fitness of the offspring;
                If ic is better than the worst chromosome then replace the worst chromosome by ic;
        Next i;
    Check if termination = true;
End;
```

Appendix III. Pseudocode for the Memetic Local Search

```
Begin;
    Select an incremental value d= a * Rand ( ), where a is a constant that suits the variable values;
    For a given chromosome i є P: calculate fitness (i);
    For j = 1 to number of variables in chromosome i;
        Value (j) = value (j) + d;
        If chromosome fitness not improved then value (j) = value (j)-d;
        If chromosome fitness not improved then retain the original value (j);
```

Next j;
End;

Appendix IV. Pseudocode for a PSO Procedure

```
Begin;
    Generate random population of N solutions (particles);
    For each individual i є N: calculate fitness (i);
    Initialize the value of the weight factor, ω;
        For each particle;
                Set pBest as the best position of particle i;
                If fitness (i) is better than pBest;
                pBest (i) = fitness (i);
        End;
        Set gBest as the best fitness of all particles;
        For each particle;
                Calculate particle velocity according to Eq. 3;
                Update particle position according to Eq. 4;
        Update the value of the weight factor, ω;
    Check if termination = true;
End;
```

Appendix V. Pseudocode for an ACO Procedure

```
Begin;
    Initialize the pheromone trails and parameters;
        Generate population of m solutions (ants);
        For each individual ant k є m: calculate fitness(k);
        For each ant determine its best position;
        Determine the best global ant;
        Update the pheromone trail;
    Check if termination = true;
End;
```

Appendix VI. Pseudocode for a SFL Procedure

```
Begin;
    Generate random population of P solutions (frogs);
    For each individual i є P: calculate fitness (i);
    Sort the population P in descending order of their fitness;
        Divide P into m memeplexes;
        For each memeplex;
                Determine the best and worst frogs;
                Improve the worst frog position using Eqs. 4 or 5;
                Repeat for a specific number of iterations;
        End;
        Combine the evolved memeplexes;
        Sort the population P in descending order of their fitness;
    Check if termination = true;
End;
```

References

[1] Lovbjerg M. Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality. Masters Thesis, Aarhus Universitet, Denmark; 2002.
[2] Holland J. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI; 1975.
[3] Al-Tabtabai H, Alex PA. Using genetic algorithms to solve optimization problems in construction. Engineering, Construction, and Architectural Management 1999;6(2):121–32.

[4] Hegazy T. Optimization of construction time-cost trade-off analysis using genetic algorithms. Canadian J of Civil Engineering 1999;26:685-97.
[5] Grierson DE, Khajehpour S. Method for conceptual design applied to office buildings. J of Computing in Civil Engineering 2002;16(2):83-103.
[6] Joglekar A, Tungare M. Genetic algorithms and their use in the design of evolvable hardware. http://www.manastungare.com/articles/genetic/genetic-algorithms.pdf, 2003, accessed on May 20, 2004, 15 pages.
[7] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California; 1989.
[8] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ;1942-48;1995.
[9] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics 1996;26(1):29-41.
[10] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J of Water Resources Planning and Management 2003;129(3):210-25.
[11] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, Mass: Addison-Wesley Publishing Co; 1989.
[12] Caudill M. Evolutionary neural networks. AI Expert 199; March:28-33.
[13] Dawkins R. The selfish gene. Oxford: Oxford University Press; 1976.
[14] Merz P, Freisleben B. A genetic local search approach to the quadratic assignment problem. Proceedings of the 7th International Conference on Genetic Algorithms, (CT. Bäck, Ed.), Morgan Kaufmann; 465-72; 1997.
[15] Moscato P, Norman MG. A memetic approach for the traveling salesman problem – implementation of a computational ecology for combinatorial optimization on message-passing systems. In: M. Valero, E. Onate, M. Jane, J.L. Larriba, B. Suarez, editors. International Conference on parallel computing and transputer application. Amsterdam, Holland; IOS Press; 1992; 177-86,
[16] Shi Y, Eberhart R. A modified particle swarm optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, NJ; IEEE Press; 1998; 69-73.
[17] Eberhart R, Shi Y. Comparison between genetic algorithms and particle swarm optimization. Proceedings of the 7th Annual Conference on Evolutionary Programming, Springer Verlag; 1998; 611-8.
[18] Kennedy J. The particle swarm: social adaptation of knowledge. Proceedings of the IEEE International Conference on Evolutionary Computation (Indianapolis, Indiana), Piscataway, NJ; IEEE Service Center; 1997; 303-8.
[19] Dorigo M, Gambardella, LM. Ant colonies for the traveling salesman problem. Biosystems, Elsevier Science 1997;43(2):73-81.
[20] Maier HR, et al. Ant colony optimization for design of water distribution systems. J of Water Resources Planning and Management 2003;129(3):200-9.
[21] Rajendran C, Ziegler H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of Jobs. European J of Operational Research 2004;155(2):426-38.
[22] Merkle D, Middendorf M, Schmeck H. Ant colony optimization for resource-constrained project scheduling. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000); 2000; 893-900.
[23] Liong S-Y, Atiquzzaman, Md. Optimal design of water distribution network using shuffled complex evolution. J of the Institution of Engineers, Singapore; 2004;44(1): 93-107.
[24] Whitley D, Beveridge R, Graves C, Mathias K. Test driving three 1995 genetic algorithms: new test functions and geometric matching. J of Heuristics 1995;1: 77-104.
[25] Feng C, Liu L, Burns S. Using genetic algorithms to solve construction time–cost trade-off problems. J of computing in civil engineering 1997;11(3):184–9.
[26] Raphael B, Smith IFC. A direct stochastic algorithm for global search. J of applied mathematics and computation 2003;146(2-3):729-58.
[27] Evolver, Evolver Version 4.0.2, Palisade Corporation; 1998.

Knowledge-Based Expert Systems

Why?

Components?

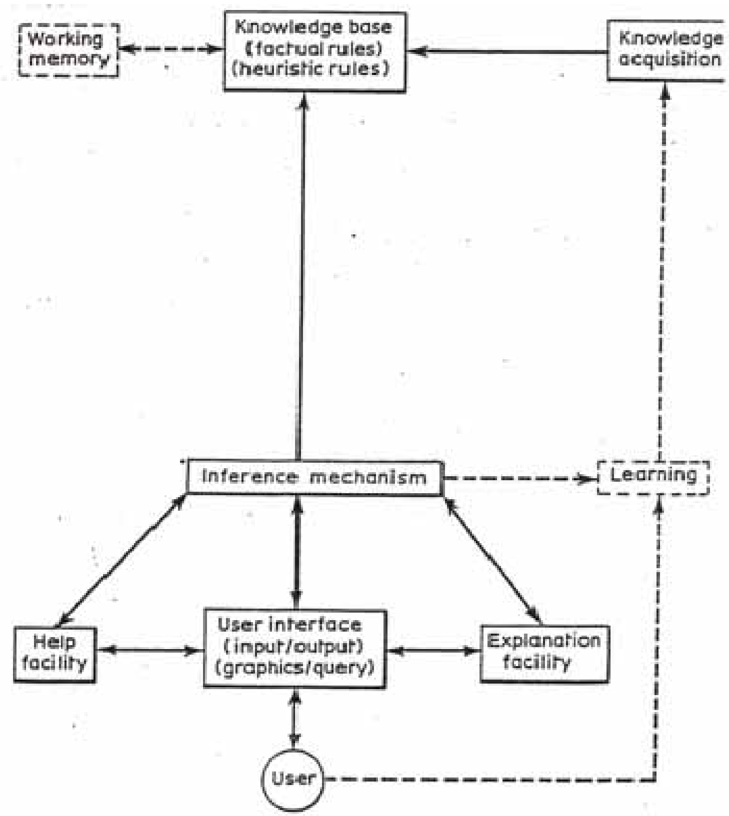
Sources of Knowledge?

Inference Mechanisms?

Points	Characteristic
(2)	1. Task is primarily cognitive, requiring analysis, synthesis, or decision making rather than perception or action
(2)	2. Involves primarily symbolic knowledge and reasoning
(2)	3. Is complex, involving many parameters
(1)	4. Involves chains of reasoning on multiple levels of knowledge
(2)	5. Uses heuristics or rules of thumb and requires judgment or reasoning about subjective factors
(1)	6. Can't be solved using conventional computing methods
(2)	7. Often must be solved with incomplete or inaccurate data
(2)	8. Often requires explanation, justification of results, or reasoning
(1)	9. Is at an intermediate stage of knowledge formalization that uses heuristics and classification rather than search or algorithms
(1)	10. Task knowledge is confined to a narrow domain
(1)	11. Task knowledge is stable
(1)	12. Incremental progress is possible; task can be subdivided
(1)	13. Doesn't require reasoning about time or space
(1)	14. Isn't natural-language intensive
(1)	15. Requires little or no common sense or general-world knowledge
(1)	16. Doesn't require the system to learn from experience
(1)	17. Is similar to one in an existing expert system
(1)	18. Data and case studies are available
(1)	19. System performance can be accurately and easily measured

$\frac{\_\_\_}{25} = \frac{\text{Points earned}}{\text{Points possible}} = \_\_\_$ score

TABLE
Desirable task characteristics (25 points).



Components of an expert system.

How to Build a KBES?

Knowledge Acquisition

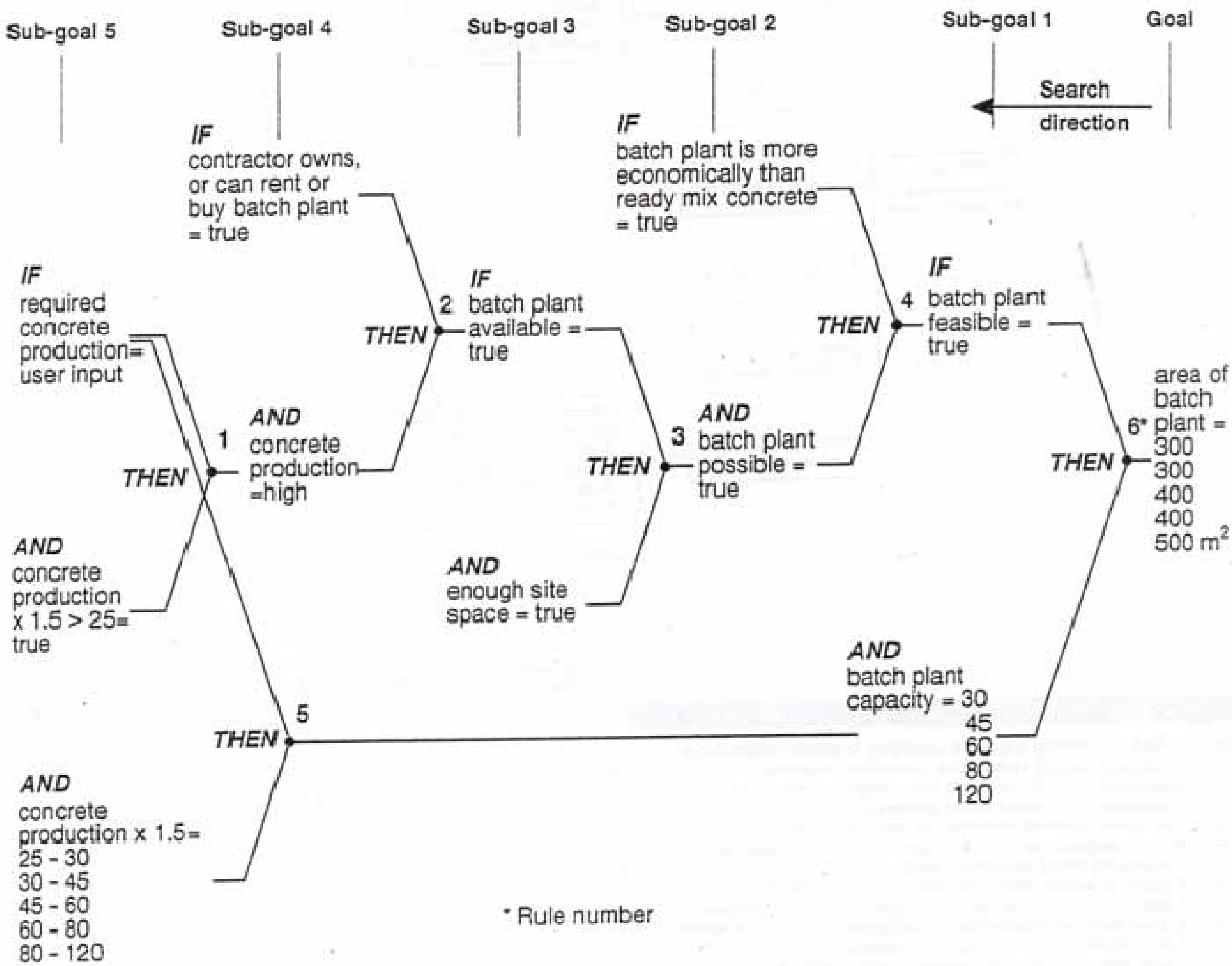
Inference Engine

Consultation

Challenges?

Variations?

Case-Based Systems



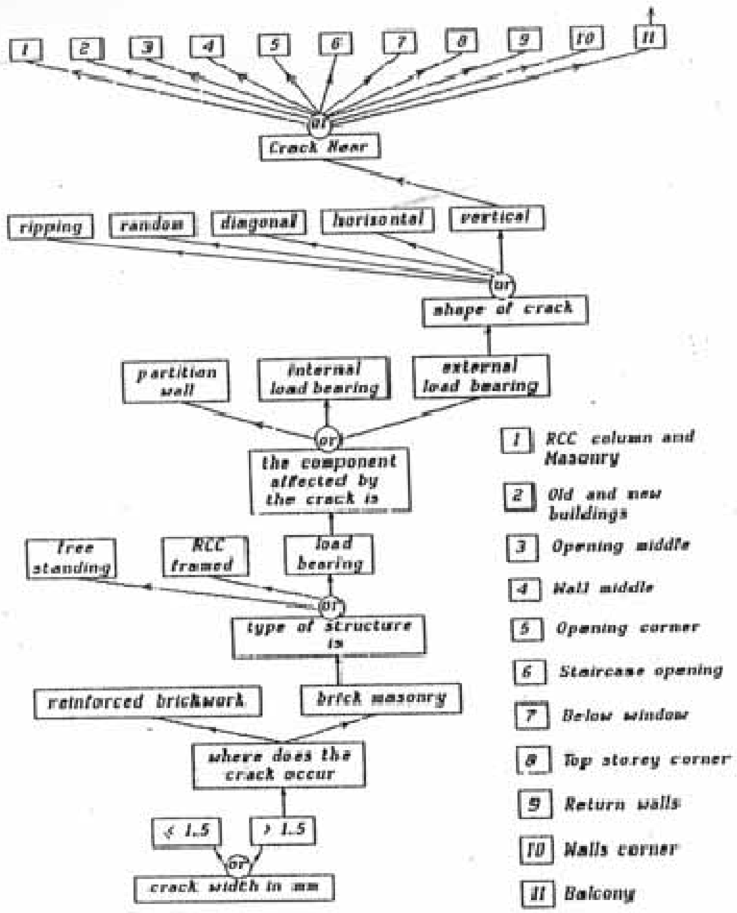
Knowledge Representation and the Backward Chaining Process

1 IF cracks are diagonal AND from dpc to opening AND in one direction THEN cause is expansion

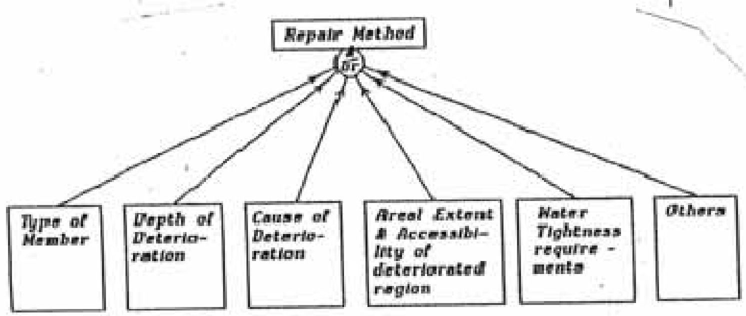
2 IF cracks are vertical AND widest at top AND on opposite sides THEN cause is movement

3 IF cracks are vertical AND in centre of panel AND widest at bottom THEN cause is expansion

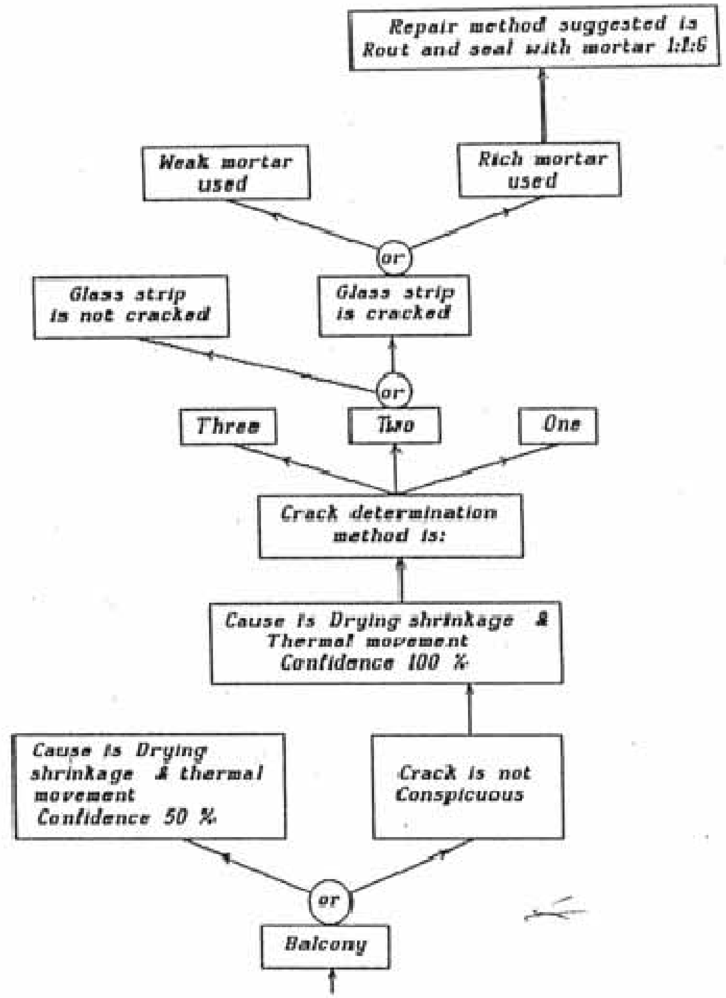
4 IF cracks are vertical AND near a corner AND it is a new building THEN cause is expansion



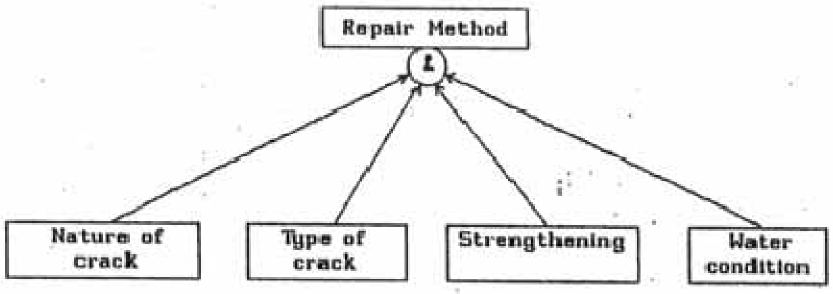
(a) Part of Knowledge Net Showing Path Taken to Reach Goal in Example



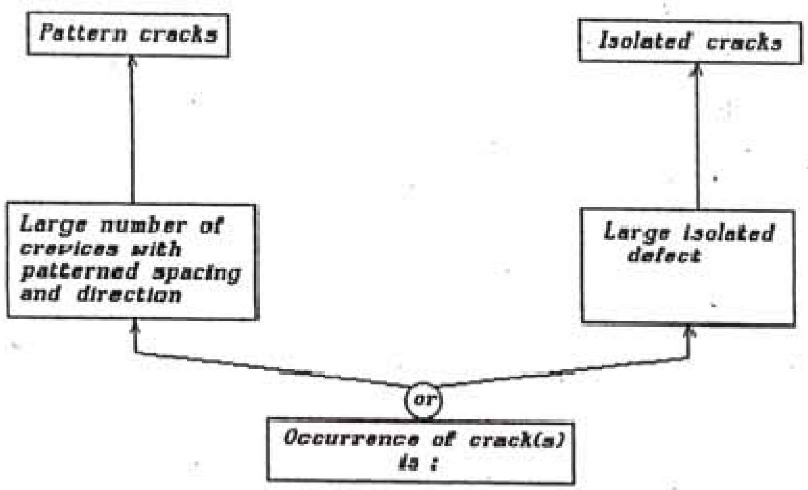
Static Knowledge for Spalling and Disintegration on Concrete



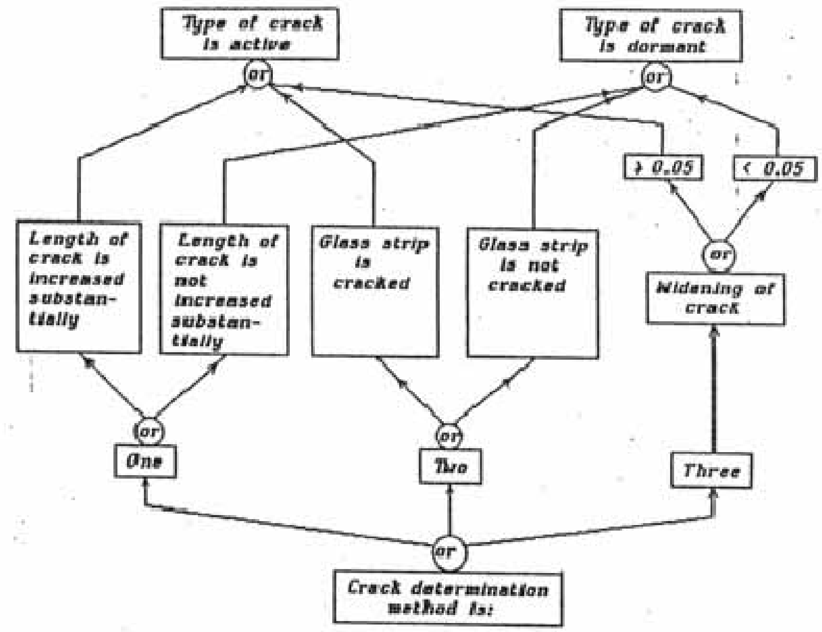
(b) Part of Knowledge Net Showing Path Taken to Reach Goal in Example



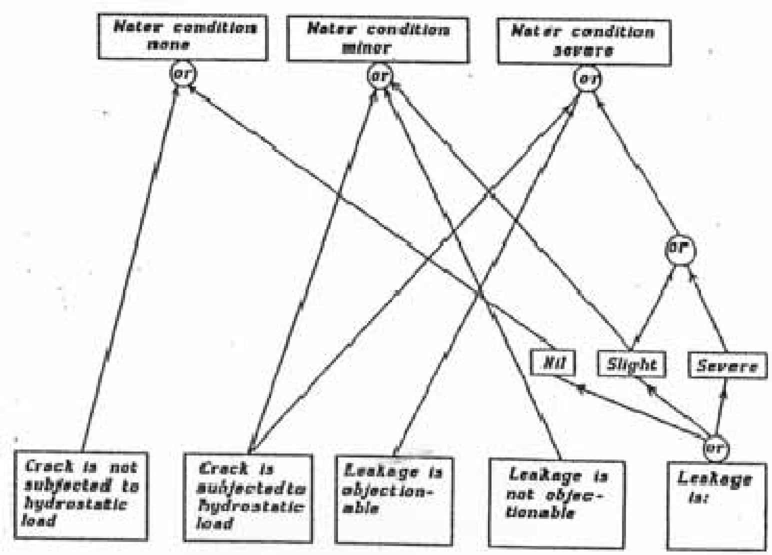
(a) Static Knowledge Net for Cracking in Concrete



(b) Sample Knowledge Net for Suggestion of Repair Method for Cracking In Concrete; Nature of Crack



(c) Sample Knowledge Net for Suggestion of Repair Method for Cracking In Concrete; Type of Crack



(d) Sample Knowledge Net for Suggestion of Repair Method for Cracking In Concrete; Water Condition

Artificial Intelligence Tools

Artificial Neural Networks

Why?

Components & Benefits?

Model Development Cycle? ______________________________

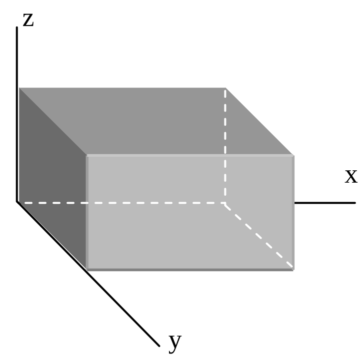
Does training work?

Purpose of Training is: ______________________________

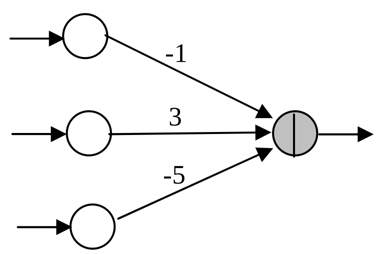
Learning Rule: ______________________________

Example

Design a simple ANN that takes the coordinates of a point and accordingly will be able to tell if the point is on the top or the bottom surface. Train the ANN on a number of cases.



Training rule is: $W_{new} = W_{old} \pm \text{Error}\ (\Delta)\ .\ X$



			Desired	Actual				
0	1	1	1					
1	0	0	0					
0	1	0	0					
0	0	1	1					

Can we write Equation? Comparison with regression?

Challenges?

Design parameters?

Generalization versus over-training?

Variations? Optimization? Sensitivity? Integration?

Computer Implementation?

Three Important References (Please print them):

Adeli, H. (2001) "*Neural Networks in Civil Engineering; 1989-2000,*" Computer-Aided Civil and Infrastructure Engineering, Vol. 16, pp. 126-142

Hegazy, T., Fazio, P., and Moselhi, O., (1994) "*Developing Practical NN Applications Using Backpropagation*," Journal of Microcomputers in Civil Engineering, Vol. 9, No. 2, pp. 145-159.

Hegazy T. and Ayed, A., (1998) "*A Neural Network Model for Parametric Cost Estimation of Highway Projects*," Journal of Construction Engineering and Management, ASCE, Vol. 24, No. 3, pp. 210-218.

EDITORIAL

WHERE AND WHY ARTIFICIAL NEURAL NETWORKS ARE APPLICABLE IN CIVIL ENGINEERING

In our attemps to design, analyze and control the behavior of systems, both man-made and natural, we engineers find that we must first be able to model and predict their complex behaviors. However, the behavior of many of these systems is governed by nonlinear multivariate (and often unknown) interrelationships, exhibits time-variance, and occurs within a "noisy," less-controllable physical environment. For example, consider the problem of detecting damage in a building with thousands of structural members by interpreting data collected from accelerometers and displacement transducers placed at various locations on the structure. This particular problem is known as an inverse mapping problem, in which the state of a system is inferred from the behavior exhibited by the system.

Within the last several years, researchers have begun to investigate the potential of artificial neural networks (ANNs) as a tool for supporting the modeling of engineering systems. An ANN is a computational mechanism able to acquire, represent, and compute a mapping from one multivariate space of information to another, given a set of data representing that mapping. An ANN is quite simply a collection of simple processing units (often simulated on a serial computer in software) that pass around activations that are filtered and modified by the connections between the processing units. What a network computes (i.e., the mapping between one pattern of information and another) is defined by the topology and nature of the interconnections between its processors. Some types of ANNs are able to automatically acquire these mappings, even when certain amounts of measurement error exist in the data.

Many of the problems that engineers must deal with are exactly the types of problems for which ANNs appear to be most applicable. Engineers are interpreters of incomplete, noisy data—such as interpreting sensor information to determine the existence, and location, of damage in a structural system. Engineers are designers and controllers of complex systems for which there is no exact model of behavior and whose expected performance is unknown, requiring that it be estimated—such as a heating, ventilation, and air-conditioning (HVAC) system within a large multifunction building.

This special issue of the journal has been organized to illustrate some of the different types of problems to which ANNs may be applied and to discuss the success, or difficulty, with which these problems have been solved using such an approach. Each paper in this special issue has been selected because it presents a unique perspective of the form and operation of ANNs, presents an overview of the applicability of ANNs to civil engineering, or presents an interesting application of an ANN to a civil-engineering-related problem. The first two papers introduce the topic of neural networks, the next two papers address problems of interpretation and classification, and the last four papers address problems of prediction and estimation.

The first paper, by Flood and Kartam, presents the concepts pertaining to ANNs in a clear and novel way, using an excellent geometric analogy, that improves the reader's understanding of the concepts. In the second paper, Flood and Kartam classify the major types of problems in civil engineering to which they feel neural networks are most applicable and present examples of each type of problem.

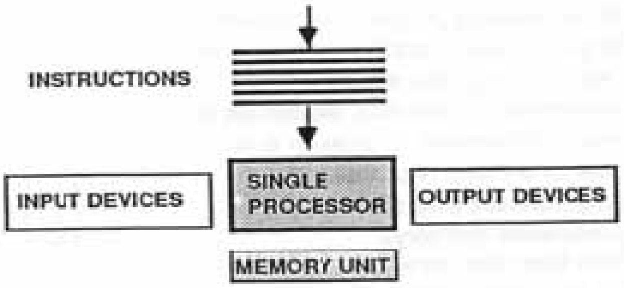
The third paper, by Szewczyk and Hajela, presents a neural-network-based approach to solving the previously mentioned inverse mapping problem of detecting damage in structural systems. The ANN they develop is able to quickly acquire and compute the mapping from patterns of displacement information to damage states (i.e., member stiffness reductions). The fourth paper, by Gagarin, Flood, and Albrecht, describes in detail an application of neural networks to another inverse mapping problem—predicting the attributes of a passing truck from strain response readings taken from the bridge it passes over. The benefit of this approach is that bridge loading statistics can be acquired using more conventional, and more reliable, strain gage technology, and without the need to stop and weigh vehicles.

The fifth paper, by Karunanithi, Grenney, Whitley, and Bovee, presents an application of an ANN as a predictive model. In this case, the model is used to predict the flow of a river. The ANN is trained to take a period of historical river-flow data and to predict the flow beyond immediately beyond that period. The authors compare their ANN-based predictive approach with others currently in use. The sixth paper, by Murtaza and Fisher, presents an application of an ANN to problems of estimating downstream outcomes (behaviors or performance metric values) of early upstream decisions. The authors describe the development of an ANN used to estimate the feasibility of using construction modularization for a given project. The objective of this ANN is different from that of Karunanithi et al.; this ANN must make an estimate of feasibility from early project information and not on data collected during the time immediately preceding the time of prediction. In the seventh paper, Chao and Skibniewski also describe the use of an ANN as an estimator. They describe how to use an ANN to estimate the productivity of various construction activities. The authors present an interesting approach for acquiring the data they need to train the network—they observe bench-scale operations and construction simulations. Using ANNs as a means of capturing and efficiently utilizing simulation data is an interesting and effective application of this technology. The last paper, by Rogers, presents an application of an ANN to the problem of behavior modeling or prediction of behavior. In this paper, Rogers describes the use of an ANN as a surrogate behavior model for a more complicated and time-consuming finite-element-based structural analysis procedure.

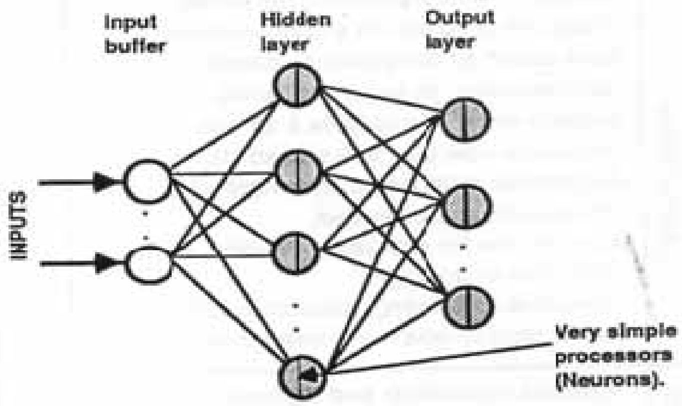
As illustrated by the applications presented in this special issue, neural networks will play a role in addressing the tasks of interpretation, classification, modeling, prediction, and estimation in civil engineering. There are other types of problems that are not illustrated in this special issue, such as control and optimization, in which neural networks have also been shown to be promising. The advantages of using ANNs for these problems are that they are universal approximators of multivariate nonlinear mappings, some are able to acquire this mapping automatically, they are able to compute these mappings when there are small amounts of error in the input patterns, and they are able to compute these mappings very fast because of their inherent parallel nature. The intent of this special issue is to provide our readers with a focused discussion and illustration of appropriate civil engineering problems to which ANNs should be applied and why.

James H. Garrett Jr.
Associate Professor of Civil Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890

NEURAL NETWORK CHARACTERISTICS



DIGITAL COMPUTER
(serial architecture)



NEURAL NETWORK
(parallel architecture)

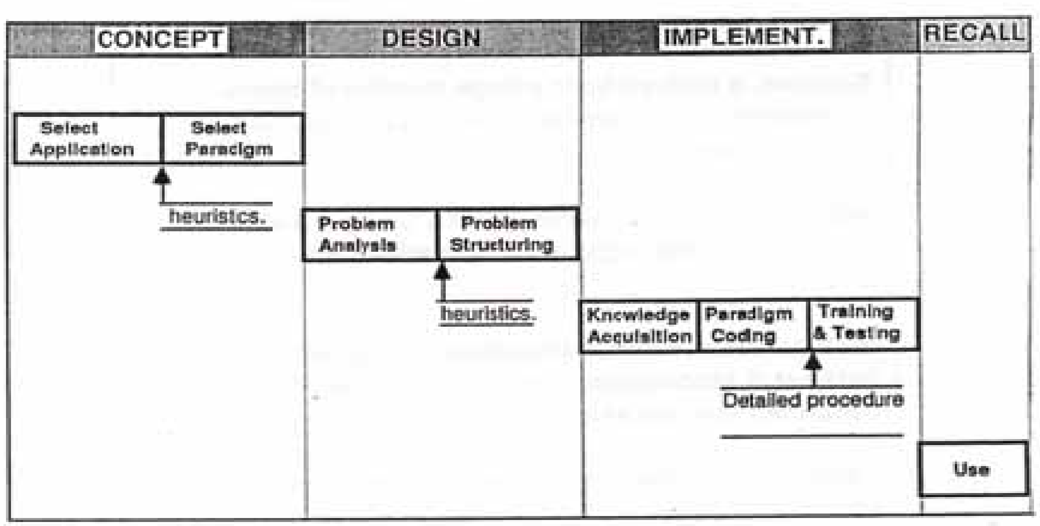
ADVANTAGES OF N.N. :

DUE TO STRUCTURE:

- Speedy Processing.
- Fault Tolerant.
- Distributed Memory.
- Efficient Performance. - Hardware

DUE TO OPERATION:

- Learn by Example.
- Can self-organize.
- Low Requirements.
- Able to Generalize.
- Have Associative Memory.



HUMAN BRAIN

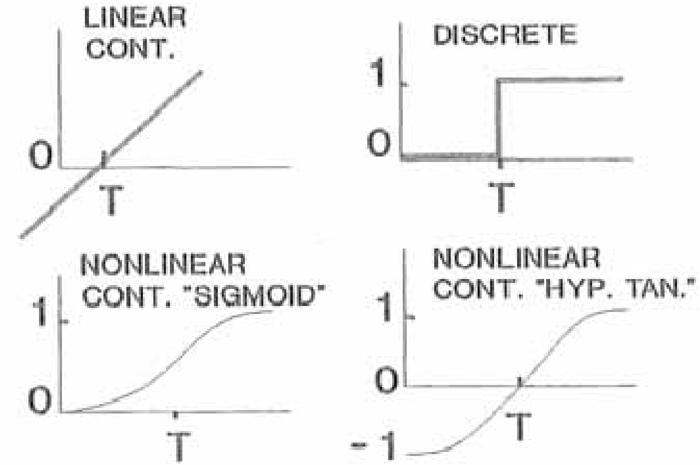
10^{11} P.E.S

10^{15} INTERCONNECTIONS

NEURAL NETWORK PARADIGMS

- PERCEPTRON.
- BACKPROPAGATION.
- COUNTER PROPAGATION.
- HOPFIELD.
- BAM.
- ART.

TRANSFER FUNCTION



BACKPROPAGATION ERROR FUNCTION

$$E = \frac{1}{2P} \sum_{j=1}^{P} \sum_{i=1}^{n} (T_{ij} - O_{ij})^2$$

- E: TOTAL ERROR
- P: TRAINING EXAMPLES
- n: OUTPUT P.E.S
- T_{ij}: DESIRED OUTPUTS
- O_{ij}: ACTUAL OUTPUTS

BACKPROPAGATION TRAINING

DELTA RULE

$$W_{NEW} = W_{OLD} + \eta * \delta * X$$

- FOR EACH P.E.
- η: LEARNING RATE COEFF.
- δ: ERROR (DES.-ACT.)
- X: INPUT VALUE

ADJUST WEIGHTS MINIMIZING THE ERROR FUNCTION

	EXPERT SYSTEMS
Limitations	-Limited to well-understood reasoning intensive problems in a narrow domain. -Low speed as compared to parallel architectures (in large problems). -Experts must be available & able to articulate how they arrive at solutions. -Expensive software development and maintenance are needed. -Not suitable for problems including induction or analogy. -Incapable of learning, generalization, or respond to noisy or unseen data.
Advantages	**-Provide reasoning and explanation capabilities & good user interface. -Facilitate interaction with needed numerical computations. -Preserve heuristic knowledge**

NEURAL NETWORKS	
-Solve analogy problems considering large number of attributes in parallel. -Learn through few examples and generalize the solution, responsds to noisy, incomplete, or unseen data. -Able to perform real time tasks. -Fault tolerant (has distributed and associative memory). -Very fast (has parallel architecture). -No knowledge acquisition problem. -Little or no software required. -Less user sofistication required to use it.	Advantages
-Sometimes provide non-reasonable response although well trained and thoroghly checked. -Do not provide explanation.	Limitations

PROBLEM CHARACTERISTICS

CRITERA FOR NEURAL NETWORK APPLICATIONS	MARKUP PROBLEM
Conventional computer technology is inadequate.	✓
Problem requires qualitative or complex quantitative reasoning.	✓
Problem is routine and knowledge is mainly implicit and can not be modeled in IF..THEN rules.	✓
Solution is derived from a large number of highly-interdependent parameters that have no precise quantification.	✓
Solution is derived quickly, based on "gut feeling" and no setp-wise logic or computations are involved.	✓
Problem area is rich in historical examples but data set is incomplete, contains errors, and describes specific examples.	✓
Development time is short, but sufficient training time is available.	✓

Exercise

Develop an ANN to estimate the cost per square foot for one-story homes. Enter data for 40 houses but use only 30 training cases and 10 test cases. After you enter the data, erase or initialize the weights. Then, use solver to obtain an average error of 10% and add a constraint that each individual case has only 15% error. Test the accuracy of the ANN on the 10 test cases. If you train the 30 cases on an error level of 20%, what is the impact on the error of the test cases?

Creative One-Story Homes: *Designed for Convenience*

DESIGN INFORMATION

Page	Design	Sq. Footage	Bdrm	Bath	Foundation	Framing	Facade	Roof	Width & Depth
24	9738	2,136	3	2½	Crawl	2x4	Siding	Conv.	76'-4" x 64'-4"
25	9764	1,815	3	2	Crawl	2x4	Siding	Conv.	70'-8" x 70'-2"
26	8063	1,789	3	2	Crawl, Slab	2x4	Brick	Conv.	78'-0" x 47'-0"
27	9734	1,977	3	2	Crawl	2x4	Siding	Conv.	69'-8" x 67'-6"
28	3348	2,549	4	2½	Basement	2x6	Siding	Truss	88'-8" x 53'-6"
28	3466	1,800	2	2	Crawl	2x6	Siding	Truss	89'-0" x 46'-2"
29	2947	1,830	3	2	Basement	2x6	Siding	Conv.	75'-0" x 43'-5"
30	3332	2,168	3	2½	Basement	2x6	Siding	Comb.	76'-4" x 46'-0"
30	2603	1,949	3	2½	Basement	2x4	Brick	Truss	74'-10" x 42'-10"
31	2880	2,758	3	2½ + ½	Basement	2x6	Siding	Conv.	81'-4" x 76'-0"
32	3360	4,062	3	2½	Basement	2x4	Brick	Truss	60'-0" x 72'-0"
33	3559	2,916	3	2½	Basement	2x6	Siding	Truss	77'-10" x 73'-10"
33	3560	2,189	3	2	Slab	2x6	Stucco	Truss	56'-0" x 72'-0"
34	9204	1,911	3	2	Basement	2x4	Brick	Conv.	56'-0" x 58'-0"
34	9201	1,996	2	2	Basement	2x4	Brick	Conv.	64'-0" x 50'-0"
35	9362	2,172	3	3	Basement	2x4	Brick	Conv.	76'-0" x 46'-0"
36	9375	2,456	3	2½	Basement	2x4	Brick	Conv.	66'-0" x 68'-0"
37	7232	2,512	2	2½	Basement	2x4	Brick	Conv.	74'-0" x 67'-8"
38	8013	2,409	3	2½	Crawl, Slab	2x4	Brick	Conv.	85'-8" x 68'-4"
39	9806	2,697	3	2½	Basement	2x4	Brick	Conv.	65'-3" x 67'-3"
40	8009	2,326	4	2	Crawl, Slab	2x4	Brick	Conv.	79'-4" x 50'-5"
41	9831	2,150	3	2½	Basement	2x4	Brick	Conv.	64'-0" x 64'-4"
42	3346	2,032	2	2	Basement	2x6	Stucco	Conv.	64'-9" x 34'-9"
43	9885	2,295	3	2	Basement	2x4	Brick	Conv.	69'-0" x 49'-6"
43	9808	2,902	3	2½	Basement	2x4	Brick	Conv.	71'-3" x 66'-3"
44	9894	1,650	3	2½	Basement	2x4	Brick	Conv.	55'-6" x 57'-6"
44	9807	2,785	3	3	Basement	2x4	Stucco	Conv.	72'-0" x 73'-0"
45	9872	1,815	3	2½	Basement	2x4	Stucco	Conv.	60'-0" x 58'-6"
46	8923	2,361	4	3	Slab	2x4	Brick	Conv.	62'-0" x 67'-10"
47	8006	2,109	3	2	Slab	2x4	Brick	Conv.	57'-8" x 67'-0"
48	3488	1,944	3	2	Basement	2x6	Brick	Truss	72'-8" x 47'-4"
48	9161	1,923	3	2	Slab	2x4	Brick	Conv.	62'-0" x 57'-4"
49	9088	1,994	3	2	Slab	2x4	Brick	Conv.	65'-8" x 56'-6"
50	2851	2,739	3	2½	Basement	2x4	Brick	Conv.	91'-8" x 52'-0"
50	2779	3,225	3	2½	Basement	2x4	Brick	Conv.	92'-8" x 46'-8"
51	2962	2,112	2	2½	Basement	2x6	Stucco	Comb.	63'-4" x 54'-10"
52	2915	2,758	3	2½ + ½	Basement	2x6	Siding	Conv.	81'-4" x 78'-0"
52	9250	2,133	3	2½	Basement	2x4	Brick	Conv.	74'-4" x 58'-0"
53	2879	3,440	4	2½ + ½	Basement	2x6	Siding	Conv.	105'-0" x 52'-8"
54	6622	2,190	3	2	Slab	2x6	Siding	Truss	58'-0" x 54'-0"
55	6607	2,200	3	2½	Slab	2x6	Brick	Truss	63'-0" x 79'-0"
55	6634	3,477	3	2½	Slab	Block	Stucco	Comb.	95'-0" x 88'-8"
56	2922	3,505	3	2½	Slab	2x6	Stucco	Comb.	110'-7" x 66'-11"
56	3431	1,907	3	2½	Slab	2x6	Stucco	Conv.	61'-6" x 67'-4"
57	3405	3,144	4	3	Slab	2x6	Stucco	Conv.	139'-10" x 63'-8"

COSTING INFORMATION (per square foot)

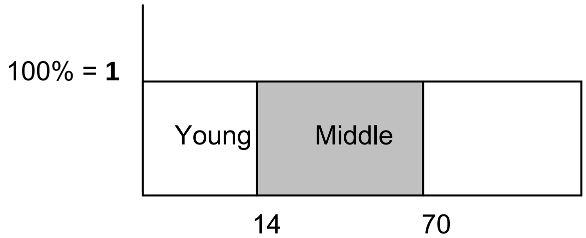
Site Work	Foundation	Framing	Ext. Walls	Roofing	Interiors	Specialties	Mechanical	Electrical	Overhead	Cost per Sq.Foot	Total Cost
1.08	11.99	24.80	26.42	5.46	16.54	8.40	8.19	5.33	7.58	$115.79	$247,327
1.15	12.67	28.21	25.58	5.74	16.27	9.79	8.38	6.10	7.97	$121.86	$221,176
0.40	7.91	13.07	16.49	5.11	13.70	8.40	8.21	4.17	5.42	$82.88	$148,272
1.11	14.46	28.59	23.49	5.77	17.15	9.88	8.70	6.19	8.07	$123.41	$243,982
1.01	10.70	14.81	15.91	2.91	13.08	8.52	7.71	3.72	5.49	$83.86	$213,759
1.16	72.63	30.09	16.48	2.02	15.83	8.51	7.38	4.90	11.13	$170.13	$306,234
1.15	11.90	14.90	16.07	4.11	13.88	9.59	8.47	3.58	5.86	$89.51	$163,803
1.07	12.81	16.39	15.93	4.40	13.74	10.53	8.61	3.48	6.09	$93.05	$201,732
1.12	12.68	14.19	18.26	3.15	13.04	8.35	7.21	3.79	5.72	$87.51	$170,557
0.97	7.97	15.35	16.53	3.28	11.90	9.94	7.39	3.89	5.40	$82.62	$240,176
0.65	6.03	10.85	14.12	2.38	11.40	9.97	7.04	2.25	4.53	$69.22	$281,172
0.96	9.63	17.22	21.32	4.54	12.31	10.71	7.36	3.31	6.12	$93.48	$272,588
0.34	8.78	11.15	14.95	3.85	13.00	12.35	8.24	3.48	5.34	$81.48	$178,360
1.13	9.67	16.27	17.00	7.46	14.81	11.19	8.25	4.22	6.30	$96.30	$184,029
1.11	9.16	16.58	18.63	7.84	14.89	43.38	8.13	4.95	8.72	$133.39	$266,246
1.07	9.59	14.93	17.24	7.38	15.34	12.55	8.31	4.42	6.36	$97.19	$211,097
1.02	43.79	17.33	20.58	4.15	13.64	14.44	8.08	4.25	8.91	$136.19	$334,483
1.02	10.72	16.35	23.04	7.06	16.97	13.54	7.41	4.30	7.03	$107.44	$269,889
0.32	4.94	19.39	20.12	5.05	17.14	9.74	8.26	3.87	6.22	$95.05	$228,975
0.99	19.26	20.62	23.30	4.24	17.24	12.25	6.52	3.74	7.58	$115.74	$312,151
0.33	10.89	12.53	15.70	4.15	13.41	10.17	7.47	3.17	5.45	$83.27	$193,686
1.08	20.81	19.24	27.55	4.43	18.58	14.99	8.45	3.49	8.30	$126.92	$272,878
1.10	13.26	18.14	20.52	3.88	14.06	8.56	8.82	3.53	6.43	$98.30	$199,746
1.05	18.79	23.99	30.44	4.19	17.04	12.80	7.51	3.35	8.34	$127.5u	$292,613
0.97	15.58	19.39	28.70	3.97	16.17	11.13	6.86	3.77	7.46	$114.00	$330,828
1.20	21.24	22.95	32.20	5.02	22.20	15.42	10.12	3.98	9.40	$143.73	$237,155
0.98	17.71	15.84	18.97	4.23	22.56	14.35	8.32	4.74	7.54	$115.24	$320,943
1.15	17.30	21.07	23.14	4.17	25.63	14.36	9.14	4.71	8.46	$129.31	$234,698
0.32	27.35	15.76	19.60	4.46	13.97	8.30	7.95	4.17	7.13	$109.01	$257,373
0.35	11.36	13.26	16.70	3.72	11.75	7.12	7.57	3.41	5.26	$80.50	$169,775
1.12	12.10	16.61	20.91	3.13	14.93	9.54	8.18	4.17	6.35	$97.04	$188,646
0.38	26.63	12.13	21.29	4.52	14.35	8.03	7.97	3.85	6.94	$106.09	$204,011
0.37	26.13	11.17	18.71	4.32	15.59	8.72	7.68	3.76	6.75	$103.20	$205,781
0.99	10.07	14.93	18.29	3.61	12.80	7.28	6.32	3.39	5.43	$83.11	$227,638
0.93	8.87	13.63	16.87	2.76	12.75	9.02	5.94	3.16	5.18	$79.11	$255,130
1.08	63.87	15.60	20.38	3.15	13.29	7.85	8.34	3.22	9.58	$146.36	$309,112
0.98	11.04	16.78	17.45	3.73	12.73	14.85	7.59	4.24	6.25	$95.64	$263,775
1.08	9.60	16.32	21.30	7.92	15.03	11.49	8.12	4.57	6.68	$102.11	$217,801
0.87	8.85	17.48	21.68	2.98	13.52	8.35	6.76	3.51	5.88	$89.88	$309,187
1.07	7.39	35.10	22.13	3.92	19.18	9.71	6.57	4.66	7.68	$117.41	$257,128
0.34	7.69	17.62	16.13	15.04	14.82	8.21	8.35	5.34	6.55	$100.09	$220,198
0.25	10.60	16.64	19.95	13.87	14.36	12.36	7.70	4.19	6.99	$106.91	$371,726
0.91	9.89	14.16	15.45	8.25	13.13	7.13	7.20	3.28	5.56	$84.96	$297,785
1.13	10.80	32.66	18.05	1.46	15.03	10.40	8.74	5.90	7.29	$111.46	$211,663
0.26	6.35	13.75	12.18	1.87	10.54	7.37	7.18	3.42	4.40	$67.32	$211,654

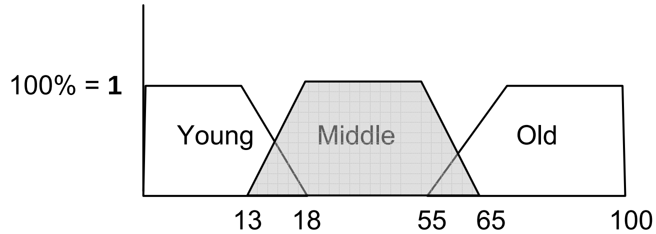
Exercise

For the above ANN, develop and sensitivity analysis sheet with 100 scenarios that are have ± 5% random variability (as done in class. Also, develop an Excel sheet for an ANN with an input buffer, 2-hidden layers, and an output layer. The layers have 7, 4, 3, and 2 elements, respectively. Consider bias elements as in the ANN template on the course web site.

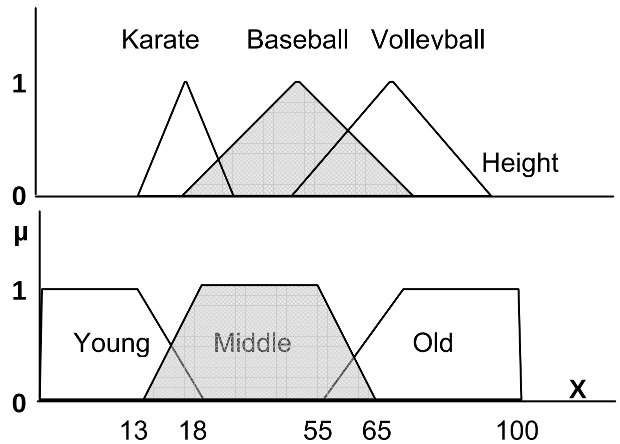
Fuzzy Sets & Fuzzy Logic

1965 Lotfi Zadeh – Fuzzy Sets (**Loose Boundaries**) i.e., everything is a matter of **degree** not True/False.





Other Shapes:



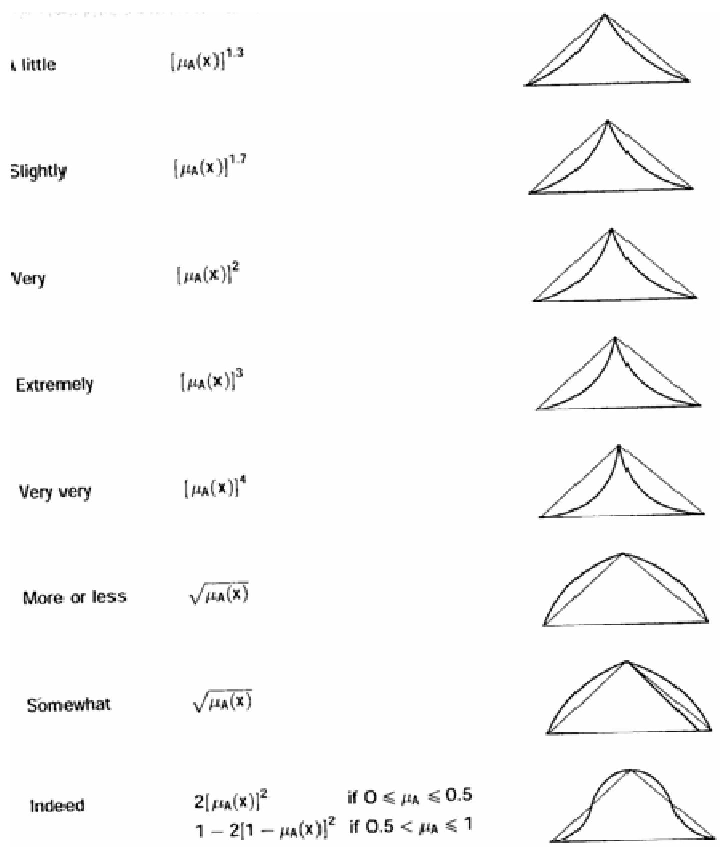
Membership Functions:

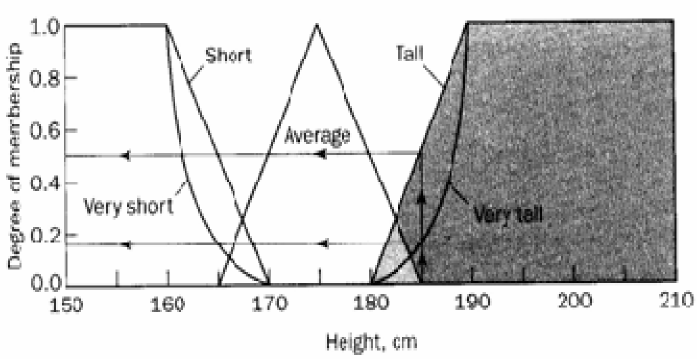
Young: X: (0, 13, 18);
μ: (1, 1 , 0) = Degree of membership

Set = X / μ = (0 / 1, 13 / 1 , 18 / 0)

Middle: ( )

Old: ( )





Operations of Fuzzy Sets:

AND OR NOT

Example….Paper

BASIC CONCEPTS OF THE THEORY OF FUZZY SETS

Editor's Note: To clarify the concept of Fuzzy sets as discussed in the preceding paper, the authors were asked to develop a simple explanation of "Fuzzy Sets." The following is included as a non-refereed addition to the article.

Roozbeh Kangari
and
LeRoy T. Boyer

Information in general can be divided into two main categories: certain and uncertain information. In a certain information system, complete data and information are available. However, most real world problems are based on uncertainty. Uncertain information can be divided into incomplete precise information which utilizes probability methods, and imprecise information for which fuzzy set theory can be used to give verbal statements a numerical clarity without losing their imprecise characteristics.

For example, if a project is completed in 240 days, this is a certain information. However, if there is 20% probability that a project will be delayed, then this incomplete precise (quantitative precision is possible) information is expressed in probability. An example of imprecise information is: "small delay" in project completion. In this case, a "small delay" represents fuzzy or imprecise information which can be analyzed by fuzzy set theory. The objective of this paper is to describe basic concepts of the theory of fuzzy sets.

A fuzzy set is class of objects with a continuum of grades of membership. Such a set is characterized by a membership function which assigns to each object a grade of membership ranging between zero and one. A fuzzy set [A] in universe X is characterized by a membership value $_A(x)$ which associates with each point in X a real number in the interval [0,1] as follows:

$$[A] = [x \mid \mu_A(x)] \qquad (1)$$

in which [A] = a fuzzy set; $\mu_A(x)$ = a membership value between zero and one; and x = an element of universe X. A membership value of one indicates "clearly belonging to" and zero indicates "clearly not belonging to" a set. It indicates the degree of belief regarding a set.

Most complex project management decisions in real world practice are made in terms of expert's intuitive judgement and are expressed linguistically. Therefore, it is important to model the expert's (project manager's) decision-making process with linguistic expressions. The values of linguistic variables are words, phrases, or sentences. For example, "management efficiency" can be considered as a linguistic variable with values: "low", "average", and "high", which are classifications of "management efficiency," but not clearly defined. In other words, they are "fuzzy". The theory of fuzzy sets can be used to mathematically analyze these linguistic variables in a logical manner to obtain an answer to complex management problems. Fuzzy set theory utilizes linguistic variables and provides a method for the transformation of verbal statements to a numerical system.

For example, a project might be considered risky if its management and cost control are weak, or, if resources availability and productivity are low. Fuzzy set theory allows translation of these words into a mathematical system.

In classical set theory, an element is either in the domain of the set, or it is not. In fuzzy set theory, the degree of membership of an element can be any value in the interval [0,1]. If the membership value is one, then the item is definitely a member. If the membership level is zero, the item is definitely not a member. If the membership value is between zero and one, the value stated indicates the belief that the item is a member of the set. Therefore, a fuzzy set consists of a number of elements with assigned degrees of membership. The degrees of membership are collectively called the membership function.

Suppose the fuzzy set [E] represents the universe of discrete "management efficiency" measured on scale of 0% to 100%. Then, "average management efficiency" may be expressed as the fuzzy set:

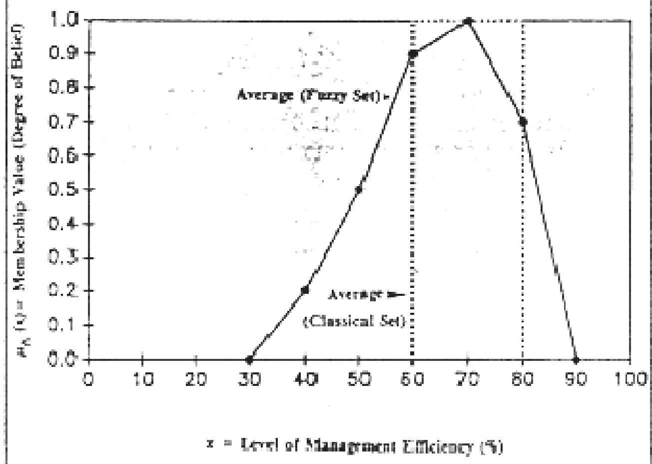


Figure 1. Graphical Representation of "Average Management Efficiency"

[Average Management Efficiency] =
[30%|0.0, 40%|0.2, 50%|0.5, 60%|0.9, 70%|1.0, 80%|0.7, 90%|0.0] (2)

This expression is shown in Figure 1, it means that 70% management efficiency is definitely a member of the set "average management efficiency." 30% or less, and 90% or more are definitely not members of the set; and 80%, 60%, 50%, and 40% are somewhere in between in terms of membership. In classical set theory the set "average management efficiency" will have membership values of one or zero as shown in Figure 1, in which one indicates definitely a member, and zero means not a member. This term in classical set could be expressed as:

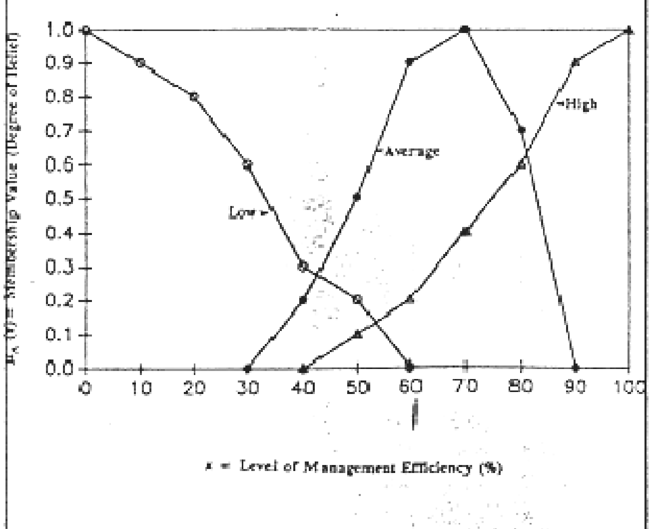


Figure 2. Graphical Representation of Fuzzy Sets

[Average Management Efficiency] =
[50%|0.0, 60%|1.0, 70%|1.0, 80%|1.0, 90%|0.0] (3)

which indicated that 60%, 70%, and 80% management efficiencies are considered "average" with membership values of one. Any value outside this range is not an "average management efficiency" and has a membership value of zero.

Utilizing concepts of the fuzzy set theory, different classifications of "management efficiency" can be defined. Figure 2 shows the levels of "management efficiency": high, average, and low. Complex project management problems can be divided into a series of detailed questions which may be answered with simple descriptive words. These descriptive words can be defined by fuzzy sets to obtain the relevant results required.

Project management decisions are often based on a combination of objective scientific knowledge, subjective information, and management judgement. The theory of fuzzy sets can be used to incorporate the combination in a logical manner. Fuzzy set techniques can be used to interpret and manipulate information from different sources, evaluate uncertainty, and incorporate judgement to improve the decision-making process.

To manipulate vague concepts, fuzzy sets can be manipulated mathematically by the following union and intersection definitions. The union of fuzzy sets [A] and [B] is defined as the larger value (max) of the support for each set, and expressed by the operation, A "or" B as follow:

$$[A] \cup [B] = [x \mid \mu_{A \cup B}(x)] = [x \mid \max(\mu_A(x), \mu_B(x))] \quad (4)$$

in which U = union of sets.

The intersection of fuzzy sets [A] and [B] is defined as the lower value (min) of the support for each set, and expressed by the operation, A "and" B as follows:

$$[A] \cap [B] = x \mid \mu_{A \cap B}(x)] = [x \mid \min(\mu_A(x), \mu_B(x))] \quad (5)$$

in which ∩ = intersection of sets. For example, if the fuzzy set [A] represents level of quality, and [B] indicates the level of profit in a project, then intersection of the two sets means that this project is satisfactory if "quality" of work and "profit" combined are satisfactory. To further clarify these equations, assume the following example: A project manager has experienced that a project's profitability [Z] is a function of three factors: management efficiency [E], productivity [P], and marketing [M]. Further, a project is profitable if its "marketing" and "management efficiency", or, its "marketing" and "productivity" is medium. Therefore, considering Eqs. 4, and 5, profitability can be represented as:

$$[Z] = [(M) \cap [E]) \cup ([M] \cap [P])$$

Consider the above equation, assume that a project manager is interested to know whether a given project is profitable if it has "Average Management Efficiency", "Average Productivity", and "Excellent Marketing"? The fuzzy sets describing each of these linguistic variables are as follows:

[E] = [30%|0.0, 40%|0.2, 50%|0.5, 60%|0.9, 70%|1.0, 80%|0.7, 90%|0.0] (7)

[P] = [30%|0.0, 40%|0.3, 50%|0.6, 60%|0.8, 70%|1.0, 80%|0.0] (8)

[M] = [40%|0.0, 50%|0.4, 60%|0.8, 70%|1.0, 80%|1.0, 90%|1.0, 100%|1.0] (9)

Utilizing Eq. 5, intersection of fuzzy sets can be calculated as:

[M] ∩ [E] = [40%|0.0, 50%|0.4, 60%|0.8, 70%|1.0, 80%|0.7, 90%|0.0] (10)

[M] ∩ [P] = [40%|0.0, 50%|0.4, 60%|0.8, 70%|1.0, 80%|0.0] (11)

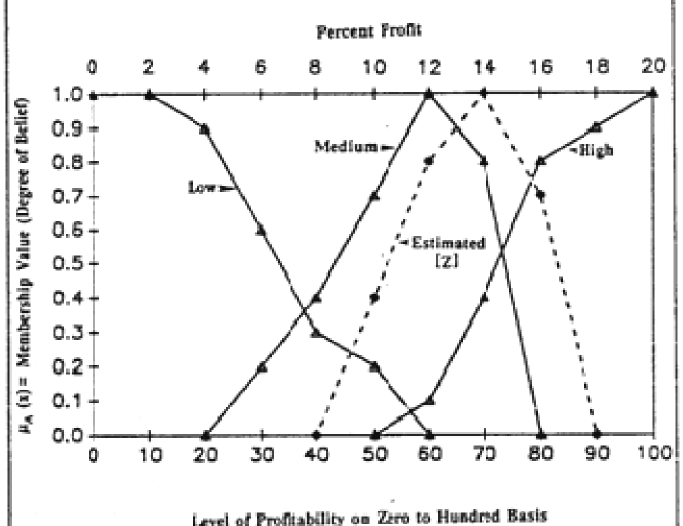


Figure 3. Estimating Profitability of a Project by Fuzzy Sets

Now, considering Eq. 4, union of the above two fussy sets is as:

[Z] = [40%|0.0, 50%|0.4, 60%|0.8, 70%|1.0, 80%|0.7, 90%|0.0] (12)

The fuzzy set [Z] shows the level of profit of this project. The overall result, [Z], is between medium and high profit curves as shown in Figure 3. Therefore, it can be concluded that this project is expected to have a medium-high profit, and it is satisfactory and profitable from project manager's view point.

The procedure of linguistic analysis using fuzzy set theory consists of: assessment of variables in linguistic term (e.g., "low management efficiency"), translations of linguistic terms into fuzzy sets (e.g, Eqs. 7, 8, and 9), mathematical calculations (e.g., Eqs. 10, and 11), evaluation of the overall results in fuzzy sets (Eq. 12), and translation of the fuzzy sets back into linguistic terms (e.g., "medium-high" in the above example). In this manner, complex risk analysis of a project is possible by combining linguistic and numeric information.

The fuzzy set approach is able to describe complicated systems where conventional mathematical methods have failed. It provides a rational and systematic approach to decision-making.

Proper development of reliable membership functions appears to be the primary obstacle to application of the theory of fuzzy sets to practical project management.

Generally, the membership values are assigned based on subjective judgement of experts and on available statistical data. It is important to give special emphasis to the way in which the membership values are assigned since they play an important role in fuzzy set theory.

Fuzzy set theory originated in the work of Lofti A. Zadeh [1] in 1965. Fuzzy set theory has since been considerably developed by Zadeh [2] and some 300 researchers. It has blossomed into a many-faceted field of science inquiry, drawing on and contributing to a wide spectrum of areas ranging from pure mathematics and physics to medicine, management, linguistics, and philosophy.

References

1. Zadeh, L.A. *Fuzzy Sets. Information and Control,* 1965, 8, 338-353.

2. Zadeh, L.A., Fu, K.S., Tamaka, K., and Shimara, J. *Fuzzy Sets and Their Application to Cognitive and Decision Processes.* Academic Press, New York, N.Y., 1975.

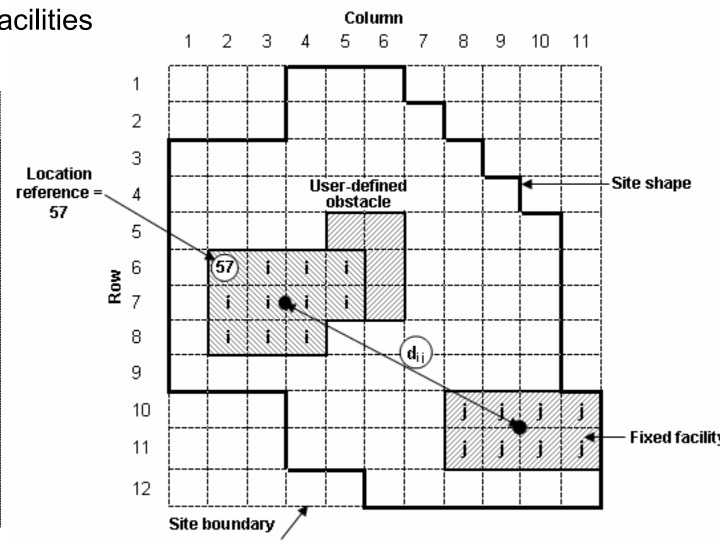
1973 Fuzzy Logic (Linguistic variables):

Fuzzy Logic = **Precisation** of approximate reasoning.

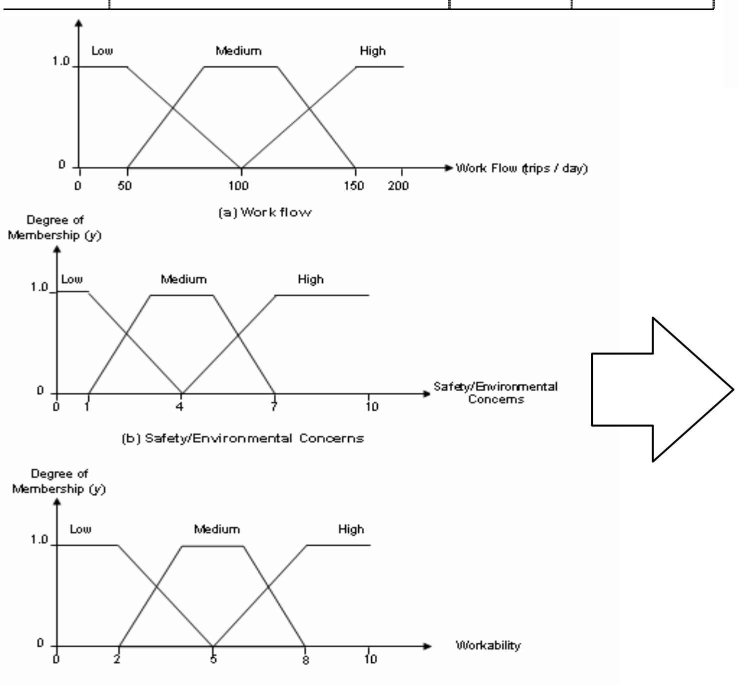
Men are mortal Joe is a man Then, Joe is Mortal	Most Sweeds are tall Joe is Sweed Then, Joe is **likely** to be tall

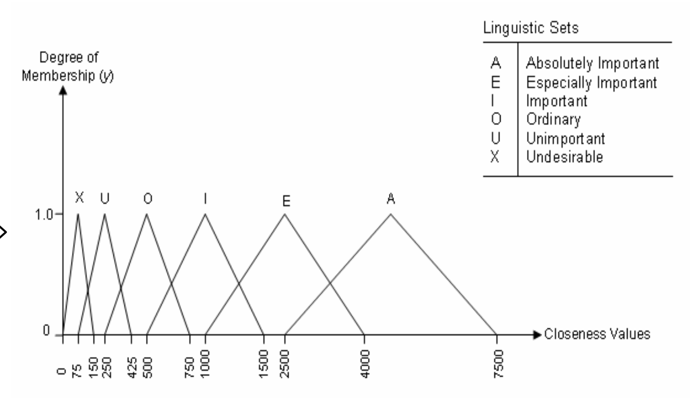
- Car driving; Understanding language
- In Engineering, it is better to be approximately right than precisely wrong. Does it make sense to say that our project will cost from $780,412.6 to $816,764.9?
- Many practical applications on machines and systems involving linguistic variables
- Natural language applications + Better web search engines based on Perception.
- **In Google**, How many horses got Ph.D. from the UW?
- What is the distance between the largest city in Spain and the largest city in France?
- **Fuzzy Logic process**: Fuzzification – IF-THEN rules – Defuzzification
- **Example on Fuzzy Logic:** Layout of Temporary Facilities

Code	*Facility Name*	*A (M^2)*	*Type*
11	Machine Room	60	Fixed
8	Parking Lot	60	Fixed
9	Tank	40	Fixed
3	Information and Guard	20	Fixed
10	Long Term Laydown Yard	400	Normal
6	Cement Warehouse	320	Normal
12	Scaffold Storage Yard Rebar	220	Normal
5	Fab/Storage Yard	60	Normal
1	Offices	40	Normal
7	Testing Laboratory	40	Normal
2	First Aid	20	Normal
4	Toilet on Site	20	Normal



Site and Facilities Representation

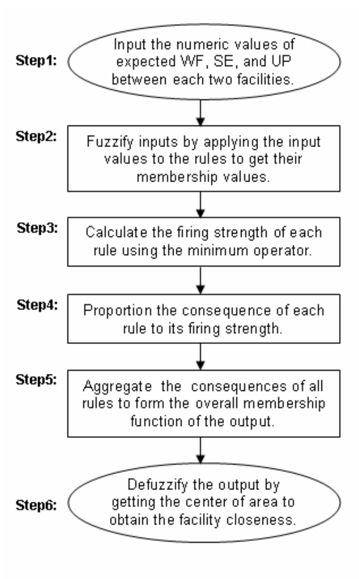




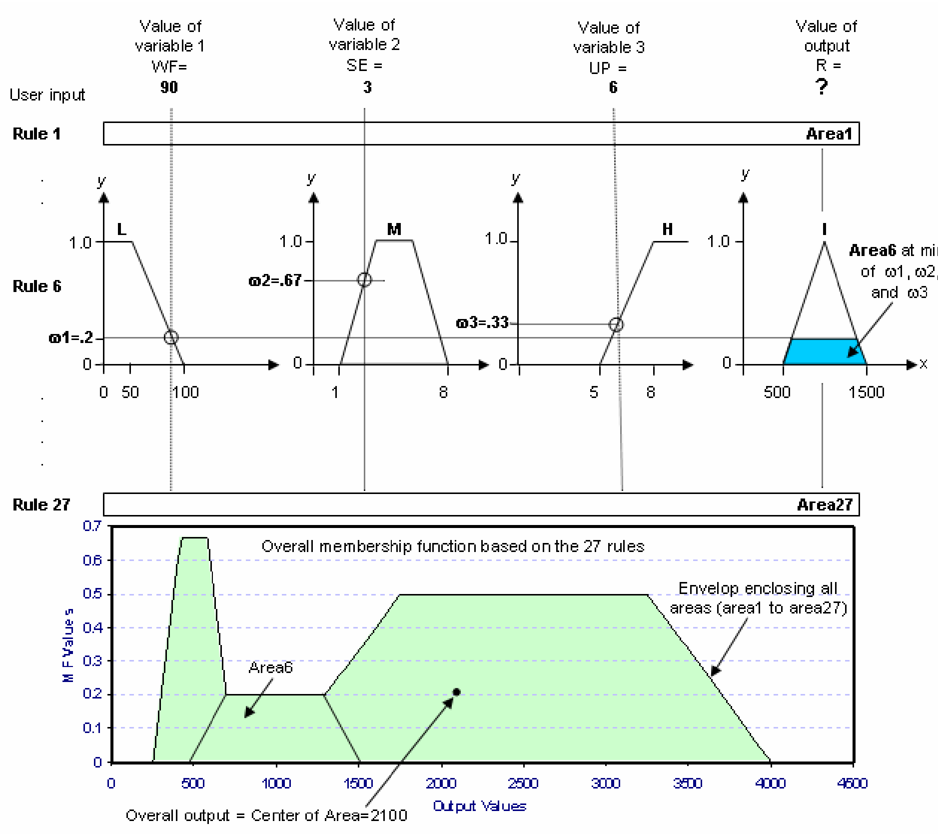
Fuzzy Output Variable "Closeness Rating"

Fuzzy Decision Rules

Rule no.	*Work Flow*	*Safety/Environmental Concerns*	*User's Preference*	*Closeness Rating*
1	Low (L)	Low (L)	Low (L)	Ordinary (O)
2	Low (L)	Low (L)	Medium (M)	Important (I)
3	Low (L)	Low (L)	High (H)	Especially Important (E)
4	Low (L)	Medium (M)	Low (L)	Unimportant (U)
5	Low (L)	Medium (M)	Medium (M)	Ordinary (O)
6	Low (L)	Medium (M)	High (H)	Important (I)
7	Low (L)	High (H)	Low (L)	Undesirable (X)
8	Low (L)	High (H)	Medium (M)	Unimportant (U)
9	Low (L)	High (H)	High (H)	Ordinary (O)
10	Medium (M)	Low (L)	Low (L)	Important (I)
11	Medium (M)	Low (L)	Medium (M)	Especially Important (E)
12	Medium (M)	Low (L)	High (H)	Absolutely Important (A)
13	Medium (M)	Medium (M)	Low (L)	Ordinary (O)
14	Medium (M)	Medium (M)	Medium (M)	Important (I)
15	Medium (M)	Medium (M)	High (H)	Especially Important (E)
16	Medium (M)	High (H)	Low (L)	Unimportant (U)
17	Medium (M)	High (H)	Medium (M)	Ordinary (O)
18	Medium (M)	High (H)	High (H)	Important (I)
19	High (H)	Low (L)	Low (L)	Especially Important (E)
20	High (H)	Low (L)	Medium (M)	Absolutely Important (A)
21	High (H)	Low (L)	High (H)	Absolutely Important (A)
22	High (H)	Medium (M)	Low (L)	Important (I)
23	High (H)	Medium (M)	Medium (M)	Especially Important (E)
24	High (H)	Medium (M)	High (H)	Absolutely Important (A)
25	High (H)	High (H)	Low (L)	Ordinary (O)
26	High (H)	High (H)	Medium(M)	Important (I)
27	High (H)	High (H)	High (H)	Especially Important (E)



Steps of Fuzzy Rule-Based Module



C.O.G = $\Sigma (\mu . x) / \Sigma \mu$

Calculating the Overall Membership function for the Output Variable

Project Scheduling Using Fuzzy Set Concepts

By Bilal M. Ayyub[1] and Achintya Haldar,[2] M. ASCE

Abstract: Probabilistic methods are being used increasingly in construction engineering. However, when a parameter is expressed in linguistic rather than mathematical terms, classical probability theory fails to incorporate the information. The linguistic variables can be translated into mathematical measures using fuzzy set and system theory. A construction management problem, i.e., estimation of the duration of an activity, is solved using this theory. In order to implement the proposed technique, various membership functions need to be estimated using judgment or with the assistance of experts. The proposed technique is not sensitive to small variations in the membership values. This is a very desirable property. However, the method is sensitive to the choice of the fuzzy relations. The uncertainty in the fuzzy relations can be modeled along with other sources of uncertainty. The mean and variance of the parameters involved in the problem under consideration are estimated here using a new method. The method maximizes the product of the sum of the membership associations for a certain frequency of occurrence and the corresponding frequency of occurrence. One of the main advantages of the proposed technique is that it can be easily implemented in existing computer programs for project scheduling.

Introduction

The construction phase is one of the most important aspects of a civil engineering structure. The success of a project depends on how well the construction phase is carried out. Efficient and economical construction is particularly important because of the increasing complexity of the structures being built, the availability of improved materials and construction equipment, the high level of competition in the industry, high interest rates, high labor cost, inflation, and regulations—all of which make construction more challenging than ever before. Thus, the economy of a project is highly dependent on accurate and elaborate analysis in the early stages of construction.

Practically every construction project is complex to some extent. In a large, complex project, there are hundreds or even thousands of operations and activities. Construction engineers use several techniques, with varying degrees of complexity, to handle project scheduling. Bar charts were one of the early tools for project scheduling (19–21). While bar charts were improved into sophisticated networks, operation research techniques such as linear programming, simulation, time and motion studies, work study methods, value engineering, statistical quality control, and inventory control were increasingly used in the construction industry. Essentially, the initial function of operation research was the analysis of existing construction operations to find more efficient per-

---

[1]Asst. Prof., Dept. of Civ. Engrg., Univ. of Maryland, College Park, Md. 20742.
[2]Asst. Prof., School of Civ. Engrg., Georgia Inst. of Tech., Atlanta, Ga. 30332.

Note.—Discussion open until November 1, 1984. To extend the closing date one month, a written request must be filed with the ASCE Manager of Technical and Professional Publications. The manuscript for this paper was submitted for review and possible publication on June 13, 1983. This paper is part of the ***Journal of Construction Engineering and Management***, Vol. 110, No. 2, June, 1984. ©ASCE, ISSN 0733-9364/84/0002-0189/$01.00. Paper No. 18889.

formance methods. In 1956, the Critical Path Method (CPM) was first formulated and implemented on a computer to schedule construction projects (1,15,18,20,21). In 1957, a technique called the Program Evaluation and Review Technique (PERT) was developed to integrate and coordinate contractors working on a single project (1,3,13,15,19–21). This method uses probability theory, and it is considered a stochastic Critical Path Method. PERT enables management to plan and control projects by knowing the probabilities of occurrence of events. Recently, a method called Graphical Evaluation and Review Techniques (GERT) was developed. GERT is the simplest way of showing the dovetailed operations in a construction project, and is useful when performance of all the operations is not necessary for the completion of a project. Nowadays, the use of these methods is increasing due to the ease of implementing them on computers (1,15,21).

All these methods can be broadly divided into two groups: deterministic and probabilistic. When the information needed for a particular method is assumed to be known during the analysis, it can be considered deterministic. Bar charts and CPM may be classified as deterministic methods. In reality, however, most of the information used in these methods is nondeterministic in nature. In other words, a particular value of a parameter, such as the duration of an activity, is not known with certainty. The incorporation of uncertainty in the parameters in project scheduling techniques leads to probabilistic methods. In these methods, each parameter is generally expressed in terms of mean, standard deviation, coefficient of variation (COV), and appropriate probability distribution. The mean value indicates the expected or average value of a parameter, e.g., duration. The standard deviation indicates the dispersion or scatter of the data from the mean value. The COV is a nondimensional quantity which is the ratio of mean and standard deviation, and is a measure of uncertainty in the parameter (2,4,14). PERT and GERT can be classified as probabilistic methods.

Basically, whether the method is deterministic or probabilistic, all the parameters need to be estimated. However, some parameters may not be estimated properly, since some of the factors that affect these parameters cannot be quantified. Instead, they are qualified. Good or bad weather can be considered as a factor that influences the duration of an activity. However, future weather conditions can be at best described as good or bad, and there is no standard acceptable numerical value attached to this qualitative statement. Consequently, these factors were not properly incorporated in the past in the estimation of the parameters. For example, PERT requires a subjective data interpretation and estimation of the duration of an activity in the form of most probable, pessimistic, and optimistic values (3,15). This subjective estimation procedure does not properly consider the different factors which affect the duration and may result in an inaccurate estimate, and, consequently, in construction delays and losses. The objective of this paper is to propose a method of incorporating such qualitative factors in the estimation of parameters. In a probabilistic formulation, this is basically to study the effects of such qualitative factors on the statistics (mean, standard deviation, COV, etc.) of the parameter. In this paper, the effects of qualitative factors are evaluated using the fuzzy set concept.

Problem Description

Construction projects are divided into activities. The relationship and sequence of these activities are presented in the form of a network. Each activity requires a certain amount of resources which may include time, labor, material, or money. The objective of a construction manager is to find the combination of resources which will minimize the total cost of not only one activity but of all the activities involved in the project, and to finish the project on time. In order to estimate the completion time of a project, the time required to finish each activity (duration of activity) needs to be estimated. The nominal duration, or the mean value and the standard deviation of the duration, or the probability distribution and its parameters of the duration of each activity need to be estimated, depending on which scheduling method is being used, i.e., CPM, PERT, GERT, or simulation techniques. Obtaining reasonable activity duration estimates is important because all subsequent calculations and decisions are based on these estimates. There are many factors which affect the duration of an activity, e.g., weather, labor skill (which changes with time because of the learning effect), superintendent experience, type of equipment used, and level of operators' experience. The effect of these factors on the duration of an activity depends on the activity being considered. For example, the pouring of concrete in an open area is highly sensitive to weather conditions compared to other factors. The construction engineer or superintendent estimates the duration of the activities using experience and judgment. The level of experience and judgment will affect the final outcome and result in uncertainties in the durations. These uncertainties need to be modeled mathematically.

The major problem lies with the factors that are expressed in linguistic, rather than mathematical terms. Good or bad weather, long or short experience, etc., fall into this category. Even the sensitivity of the activity's duration to any of these factors is measured in linguistic terms, e.g., highly sensitive, strong influence, etc. Not only are future weather conditions uncertain at the present time, the definition of good or bad weather complicates the problem. Uncertainties in future weather conditions can be modeled mathematically (2,4–6,11,12); however, additional sources of uncertainty due to the qualitative assessment of good or bad weather need to be considered. The linguistic variables can be translated into mathematical measures by fuzzy sets and systems theory. Conventional procedures like PERT can still be used if updated probabilistic input is used to obtain the required information.

In this paper, the concept of fuzzy sets and systems is introduced and applied in construction project scheduling. Weather conditions and labor skill are considered here to help explain the applicability of the fuzzy set concept in construction scheduling. However, any number of similar factors can be modeled accordingly. The proposed method for systematically qualifying the linguistic factors is realistic and simple. It could easily be implemented in the available computer programs of some project scheduling techniques, such as PERT.

Elements of Fuzzy Set Theory

Since fuzzy sets theory was introduced by Zadeh (23), it has been receiving more and more attention from researchers in many different fields (7–10,22). This section summarizes the fundamental definitions and operations of the theory of fuzzy sets that will be used in this paper, as proposed by Zadeh and others (17,24–26).

As mentioned earlier, qualitative factors or linguistic variables (terms) are routinely used in construction project scheduling. These linguistic measures add to the overall uncertainty in the final outcome of any decision process. In order to incorporate these uncertainties in the analysis, the linguistic terms (or variables) need to be translated into mathematical measures.

A linguistic variable is defined as a variable, the values of which are words, phrases, or sentences in a given language. For example, labor experience can be considered as a linguistic variable if the values of this variable, such as "long experience" or "short experience," are not clearly defined but are meaningful classifications nonetheless.

Let X be a universe, or a set of elements, x's, and let A be a subset of X. Each element, x, is associated with a membership value to the subset A, $\mu_A(x)$. If A is an ordinary, non-fuzzy, or crisp set, then the membership function is given by

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \text{ belongs to } A \\ 0 & \text{if } x \text{ does not belong to } A \end{cases} \quad \text{(1)}$$

Eq. 1 means that there are only two possibilities for an element x, either being a member of A, i.e., $\mu_A(x) = 1$, or not being a member of A, i.e., $\mu_A(x) = 0$. In this case, A has sharp boundaries. On the other hand, if the membership function is allowed to take values in the interval (0,1), A is called a fuzzy set. Therefore, A does not have sharp boundaries and the membership of x to A is fuzzy. For example, let x be the level of experience of labor which may range from excellent experience, i.e., $x = 1.0$, to "never been to a construction site," i.e., $x = 0$. By dividing the range of labor experience into increments of 0.1, "short experience," A, as a linguistic variable, can be defined as

$$\text{short experience}, \quad A = [x_1 = 1|\mu_A(x_1) = 0,\ x_2 = 0.9|\mu_A(x_2) = 0,$$
$$x_3 = 0.8|\mu_A(x_3) = 0,\ x_4 = 0.7|\mu_A(x_4) = 0,\ x_5 = 0.6|\mu_A(x_5) = 0,$$
$$x_6 = 0.5|\mu_A(x_6) = 0,\ x_7 = 0.4|\mu_A(x_7) = 0.1,\ x_8 = 0.3|\mu_A(x_8) = 0.5,$$
$$x_9 = 0.2|\mu_A(x_9) = 0.7,\ x_{10} = 0.1|\mu_A(x_{10}) = 0.9,$$
$$x_{11} = 0|\mu_A(x_{11}) = 1.0] \quad \text{(2)}$$

Or, in short, it can be expressed as

$$\text{short experience}, \quad A = (0.4|0.1,\ 0.3|0.5,\ 0.2|0.7,\ 0.1|0.9,\ 0.0|1.0) \quad \text{(3)}$$

The fuzziness in the definition of short experience is obvious from Eq. 2 or 3 as opposed to Eq. 1. It is clear from Eq. 2 or 3 that different values of x or grades of experience have different membership values, $\mu_A(x)$, to the subset A, "short experience." The values of x are 0.4, 0.3, 0.2, 0.1, and 0, and the corresponding membership values are 0.1, 0.5, 0.7, 0.9, and 1.0, respectively. Other values of x have zero membership values to the subset A. These membership values are generally assigned

based on subjective judgment with the help of experts and can be updated with more applications of the method in various projects. This is examined in detail in the example. If a crisp set were used in this example, the value of x would be 0.0 with a membership value of 1.0. Similarly, long experience, B, can be defined as

$$\text{long experience}, \quad B = (1.0|1.0,\ 0.9|0.9,\ 0.8|0.7,\ 0.7|0.2,\ 0.6|0.1) \quad \text{(4)}$$

In general, any subset A may be represented by m discrete values (or continuous intervals) of x together with membership values (or continuous membership functions), μ_A, as follows:

$$A = [x_1|\mu_A(x_1),\ x_2|\mu_A(x_2),\ \ldots,\ x_m|\mu_A(x_m)] \quad \text{(5)}$$

in which = should be interpreted as "is defined to be;" and | = a delimiter.

In order to use fuzzy sets in practical problems, some operational rules similar to those used in classical set theory need to be defined. Some of the operational rules used in this paper are described here.

Simple Operations.—The union, U, of fuzzy subsets A and B of a universe, X, corresponds to the connective "or," and its membership function is

$$\mu_{A \cup B}(x) = \max[\mu_A(x),\ \mu_B(x)] \quad \text{(6)}$$

The intersection, $\cap$, of fuzzy subsets A and B correspond to the connective "and" and its membership function is

$$\mu_{A \cap B}(x) = \min[\mu_A(x),\ \mu_B(x)] \quad \text{(7)}$$

For example, consider "superintendent experience" as a linguistic variable, to be expressed by the fuzzy subset

$$C = (1.0|1.0,\ 0.9|0.8,\ 0.8|0.6,\ 0.7|0.4,\ 0.6|0.2) \quad \text{(8)}$$

and "long labor experience" is represented by Eq. 4. Then, the labor or superintendent experience can be expressed by the union of the fuzzy subsets B and C, and is given by

$$B \cup C = (1|1,\ 0.9|0.9,\ 0.8|0.7,\ 0.7|0.4,\ 0.6|0.2) \quad \text{(9)}$$

On the other hand, the labor and superintendent experience can be expressed by the intersection of the fuzzy subsets B and C, and is given by

$$B \cap C = (1|1,\ 0.9|0.8,\ 0.8|0.6,\ 0.7|0.2,\ 0.6|0.1) \quad \text{(10)}$$

The complement of a fuzzy subset A is denoted by $\bar{A}$, and its membership function is

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \text{(11)}$$

Fuzzy Relation.—A fuzzy relation, R, or cartesian-product, $A \times B$, between two fuzzy subsets A (subset of a universe X) and B (subset of a universe Y) has the following membership function:

$$\mu_R(x_i, y_j) = \mu_{A \times B}(x_i, y_j) = \min \mu_A(x_i),\ \mu_B(y_j) \quad \text{(12)}$$

The relation is usually expressed in matrix form as

$$R = A \times B = A \begin{cases} x_1 \\ x_2 \\ \vdots \\ x_n \end{cases} \overbrace{\begin{array}{cccc} y_1 & y_2 & \cdots & y_m \\ \hline \mu_R(x_1,y_1) & \mu_R(x_1,y_2) & \cdots & \mu_R(x_1,y_m) \\ \mu_R(x_2,y_1) & \mu_R(x_2,y_2) & \cdots & \mu_R(x_2,y_m) \\ \vdots & & & \vdots \\ \mu_R(x_n,y_1) & \mu_R(x_n,y_2) & \cdots & \mu_R(x_n,y_m) \end{array}}^{B} \quad \ldots\ldots (13)$$

With this notation, $\mu_R(x_i, y_j)$ indicates the support, or membership, value for the ordered pair (x_i, y_j), and is a measure of association between x_i and y_j. It is computed as the minimum value of the membership values $\mu_A(x_i)$ and $\mu_B(y_j)$.

A fuzzy relation can be expressed in a conditional form. For example, let the relation, R, be defined as: if labor experience is short, then the rate of accidents is medium. Defining "short experience" by Eq. 3 and a medium rate of accidents as

medium rate of accidents = (0.7|0.2, 0.6,|0.7, 0.5|1, 0.4|0.7, 0.3|0.2) .. (14)

the fuzzy relation, R, becomes

R = (15)

medium rate of accidents \ short experience	0.4	0.3	0.2	0.1	0
0.7	0.1	0.2	0.2	0.2	0.2
0.6	0.1	0.5	0.7	0.7	0.7
0.5	0.1	0.5	0.7	0.9	1.0
0.4	0.1	0.5	0.7	0.7	0.7
0.3	0.1	0.2	0.2	0.2	0.2

Note that the fuzzy subsets "short experience" and "medium rate of accidents" are from two different universes, namely, "experience" and "rate of accidents," respectively. The membership values of the first row in Eq. 15 are evaluated as follows:

$\mu_R(0.7, 0.4) = \min(0.2, 0.1) = 0.1$

$\mu_R(0.7, 0.3) = \min(0.2, 0.5) = 0.2$

$\mu_R(0.7, 0.2) = \min(0.2, 0.7) = 0.2$

$\mu_R(0.7, 0.1) = \min(0.2, 0.9) = 0.2$

$\mu_R(0.7, 0) = \min(0.2, 1.0) = 0.2$

The union of two relations, say R and S, is denoted by $R \cup S$ and has the following membership function:

$$\mu_{R\cup S}(x_i, y_j) = \max[\mu_R(x_i, y_j),\ \mu_S(x_i, y_j)] \quad \ldots\ldots (16)$$

On the other hand, the intersection of two fuzzy relations, $R \cap S$, has the following membership function:

$$\mu_{R\cap S}(x_i, y_j) = \min[\mu_R(x_i, y_j),\ \mu_S(x_i, y_j)] \quad \ldots\ldots (17)$$

More generally, if R_i, for $i = 1, 2, \ldots, n$, are fuzzy relations, then

194

$$\mu_{\bigcup_{k=1}^{n} R_k}(x_i, y_j) = \max_{k=1}^{n} [\mu_{R_k}(x_i, y_j)] \quad \ldots\ldots (18)$$

$$\mu_{\bigcap_{k=1}^{n} R_k}(x_i, y_j) = \min_{k=1}^{n} [\mu_{R_k}(x_i, y_j)] \quad \ldots\ldots (19)$$

The compliment of fuzzy relations has the following membership functions:

$$\mu_{\bar{R}}(x_i, y_j) = 1 - \mu_R(x_i, y_j) \quad \ldots\ldots (20)$$

Fuzzy Composition.—If R is a fuzzy relation from X to Y, and S is a fuzzy relation from Y to Z, the composition of R and S is a fuzzy relation that is described by the following membership function:

$$\mu_{RoS}(x_i, z_k) = \max_{y_j} \{\min[\mu_R(x_i, y_j),\ \mu_S(y_j, z_k)]\} \quad \ldots\ldots (21)$$

Eq. 21 basically evaluates a fuzzy relation between the fuzzy subsets X and Z using the fuzzy relations of X and Z to the common fuzzy subset Y.

An interesting case of fuzzy composition is the composition of a fuzzy subset A with a relation R. The membership function is described by

$$\mu_{AoR}(y_j) = \max_{x_i} \{\min[\mu_A(x_i),\ \mu_R(x_i, y_j)]\} \quad \ldots\ldots (22)$$

Eq. 22 gives the membership values for a fuzzy subset of y's induced by the fuzzy subset of x's, i.e., induced by the fuzzy subset A.

Eq. 15 represents a fuzzy relation between a medium rate of accidents and short experience. Suppose a new fuzzy subset for a medium rate of accidents has been proposed, and is given by

A = (0.7|0.3, 0.6|1, 0.5|0.99, 0.4|0.8, 0.3|0.1) ..... (23)

Then, the expectation of "short experience" of labor, B, is given by the composition AoR, i.e.

B = (0.4|0.1, 0.3|0.5, 0.2|0.7, 0.1|0.9, 0|0.99) ..... (24)

The first element in Eq. 24 is obtained by taking the maximum value of [min (0.3, 0.1), min (1, 0.1), min (0.99, 0.1), min (0.8, 0.1), min (0.1, 0.1)] = max (0.1, 0.1, 0.1, 0.1, 0.1) = 0.1.

The fuzzy condition statement, "if A_1 then B_1 else if A_2 then B_2 ... else if A_n then B_n" is defined to be

$$(A_1 \times B_1) \cup (A_2 \times B_2) \ldots \cup (A_n \times B_n) \quad \ldots\ldots (25)$$

As an illustration of the application of fuzzy sets to construction project scheduling, the following example is presented. For further details of the fuzzy set theory, the reader is referred to Refs. 7–10, 16, and 22.

Example

In this example, a procedure is presented for estimating the probability mass function of the duration of an activity, which is the basis of

195

any probabilistic project scheduling method. There are many factors which affect the duration of an activity, as was mentioned earlier. For the purpose of illustration, only two factors which affect the duration of a construction activity are considered here in estimating the duration statistics. These two factors are: (1) Weather, which is classified into good, G; medium, M; and bad, B; and (2) skill or labor experience, which is classified into high, H; medium, M; and low, L.

The frequency of occurrence, F, of each classification of the preceding factors, and the adverse consequences of occurrence, C, on the duration of an activity are estimated in linguistic terms, as shown in Table 1. The objective now is to estimate the duration of the activity using the information in Table 1. Then, the impact of these factors on the duration is studied.

The following translation of linguistic variables into fuzzy subsets are assumed. The membership values would vary from project to project. The assumed membership values are selected for illustration purposes only. The selection of a particular set of membership values and its effects on the statistics of the duration are examined later:

$$\text{large} = (0.8|0.5, 0.9|0.9, 1|1) \quad \text{(26)}$$

$$\text{small} = (0|1, 0.1|0.9, 0.2|0.5) \quad \text{(27)}$$

$$\text{medium} = (0.3|0.2, 0.4|0.8, 0.5|1, 0.6|0.8, 0.7|0.2) \quad \text{(28)}$$

$$\text{very large} = (\text{large})^2 = (0.8|0.25, 0.9|0.81, 1|1) \quad \text{(29)}$$

$$\text{quite small} = (\text{small})^{1.25} = (0|1, 0.1|0.88, 0.2|0.42) \quad \text{(30)}$$

$$\text{very small} = (\text{small})^2 = (0|1, 0.1|0.81, 0.2|0.25) \quad \text{(31)}$$

The concepts of Eqs. 29–31 were proposed by Blockley (8) and Yao (22) and are quite logical from a practical point of view.

Combining the frequency of occurrence, F, and the adverse consequences, C, for each factor $i = 1, \ldots, 6$ and using Eq. 14, the following fuzzy relations can be calculated:

$$F_1 \times C_1 = \begin{array}{c|ccc} \text{frequency = small} & \multicolumn{3}{c}{\text{consequences = large}} \\ & 0.8 & 0.9 & 1 \\ \hline 0.0 & 0.5 & 0.9 & 1 \\ 0.1 & 0.5 & 0.9 & 0.9 \\ 0.2 & 0.5 & 0.5 & 0.5 \end{array} \quad \text{(32)}$$

$$F_2 \times C_2 = \begin{array}{c|ccccc} \text{frequency = medium} & \multicolumn{5}{c}{\text{consequences = medium}} \\ & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 \\ \hline 0.3 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.8 & 0.8 & 0.8 & 0.2 \\ 0.5 & 0.2 & 0.8 & 1 & 0.8 & 0.2 \\ 0.6 & 0.2 & 0.8 & 0.8 & 0.8 & 0.2 \\ 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{array} \quad \text{(33)}$$

TABLE 1.—Quantitative Description of Frequency of Occurrences and Consequences

i (1)	Factor (2)	Frequency of occurrence, F_i (3)	Adverse consequences on duration, C_i (4)
1	Weather, B	small	large
2	Weather, M	medium	medium
3	Weather, G	medium	very small
4	Labor, L	large	medium
5	Labor, M	medium	quite small
6	Labor, H	quite small	very small

$$F_3 \times C_3 = \begin{array}{c|ccc} \text{frequency = medium} & \multicolumn{3}{c}{\text{consequences = very small}} \\ & 0 & 0.1 & 0.2 \\ \hline 0.3 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.8 & 0.8 & 0.25 \\ 0.5 & 1 & 0.81 & 0.25 \\ 0.6 & 0.8 & 0.8 & 0.25 \\ 0.7 & 0.2 & 0.2 & 0.2 \end{array} \quad \text{(34)}$$

$$F_4 \times C_4 = \begin{array}{c|ccccc} \text{frequency = large} & \multicolumn{5}{c}{\text{consequences = medium}} \\ & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 \\ \hline 0.8 & 0.2 & 0.5 & 0.5 & 0.5 & 0.2 \\ 0.9 & 0.2 & 0.8 & 0.9 & 0.8 & 0.2 \\ 1 & 0.2 & 0.8 & 1 & 0.8 & 0.2 \end{array} \quad \text{(35)}$$

$$F_5 \times C_5 = \begin{array}{c|ccc} \text{frequency = medium} & \multicolumn{3}{c}{\text{consequences = quite small}} \\ & 0 & 0.1 & 0.2 \\ \hline 0.3 & 0.2 & 0.2 & 0.2 \\ 0.4 & 0.8 & 0.8 & 0.42 \\ 0.5 & 1 & 0.88 & 0.42 \\ 0.6 & 0.8 & 0.8 & 0.42 \\ 0.7 & 0.2 & 0.2 & 0.2 \end{array} \quad \text{(36)}$$

$$F_6 \times C_6 = \begin{array}{c|ccc} \text{frequency = quite small} & \multicolumn{3}{c}{\text{consequences = very small}} \\ & 0 & 0.1 & 0.2 \\ \hline 0 & 1 & 0.81 & 0.25 \\ 0.1 & 0.88 & 0.81 & 0.25 \\ 0.2 & 0.42 & 0.42 & 0.25 \end{array} \quad \text{(37)}$$

The total effect of all the factors on the activity duration is obtained by taking the union of Eqs. 32 through 37, using Eq. 18, i.e.

Total effect, $T = (F_1 \times C_1) \cup (F_2 \times C_2) \ldots \cup (F_6 \times C_6)$ .... (38)

Therefore

$T =$ (rows: frequency; columns: consequences) .... (39)

frequency \ consequences	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	1	0.81	0.25	0	0	0	0	0	0.5	0.9	1
0.1	0.88	0.81	0.25	0	0	0	0	0	0.5	0.9	0.9
0.2	0.42	0.42	0.25	0	0	0	0	0	0.5	0.5	0.5
0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0	0	0
0.4	0.8	0.8	0.42	0.2	0.8	0.8	0.8	0.2	0	0	0
0.5	1	0.88	0.42	0.2	0.8	1	0.8	0.2	0	0	0
0.6	0.8	0.8	0.42	0.2	0.8	0.8	0.8	0.2	0	0	0
0.7	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0	0	0
0.8	0	0	0	0.2	0.5	0.5	0.5	0.2	0	0	0
0.9	0	0	0	0.2	0.8	0.9	0.8	0.2	0	0	0
1	0	0	0	0.2	0.8	1	0.8	0.2	0	0	0

To establish a fuzzy relation, $R(c,d_a)$, between the fuzzy subset of consequences, C, and the fuzzy subset of duration of the activity in days, D_a, let the duration be very large if the consequences are large; let the duration be large if the consequences are medium; and let the duration be small if the consequences are small, i.e.

$R{:}D_a$ = very large = (15|0.04, 18|0.64, 20|1), if C is large;

$R{:}D_a$ = large = (15|0.2, 18|0.8, 20|1), if C is medium;

$R{:}D_a$ = small = (18|0.2, 15|0.5, 10|1), if C is small .... (40)

The components of R can be combined using Eqs. 13, 18, and 25. Therefore

$R_1 = C_1 \times D_{a1} =$ .... (41)

consequences = large \ duration = very large	15	18	20
0.8	0.04	0.5	0.5
0.9	0.04	0.64	0.9
1	0.04	0.64	1

$R_2 = C_2 \times D_{a2} =$ .... (42)

consequences = medium \ duration = large	15	18	20
0.3	0.2	0.2	0.2
0.4	0.2	0.8	0.8
0.5	0.2	0.8	1
0.6	0.2	0.8	0.8
0.7	0.2	0.2	0.2

$R_3 = C_3 \times D_{a3} =$ .... (43)

consequences = small \ duration = small	10	15	18
0	1	0.5	0.2
0.1	0.9	0.5	0.2
0.2	0.5	0.5	0.2

Taking the union of R_1, R_2 and R_3, the relation, R, is obtained

$R = R_1 \cup R_2 \cup R_3 =$ .... (44)

consequences \ duration	10	15	18	20
0	1	0.5	0.2	0
0.1	0.9	0.5	0.2	0
0.2	0.5	0.5	0.2	0
0.3	0	0.2	0.2	0.2
0.4	0	0.2	0.8	0.8
0.5	0	0.2	0.8	1
0.6	0	0.2	0.8	0.8
0.7	0	0.2	0.2	0.2
0.8	0	0.04	0.5	0.5
0.9	0	0.04	0.64	0.9
1	0	0.04	0.64	1

A subjective estimation of the duration can be calculated by taking the composition of T and R, which are given in Eqs. 39 and 44, respectively. Therefore, by using Eq. 21, $T_o R$ becomes

$T_o R =$ ..... (45)

frequency \ duration	10	15	18	20	; row summation
0	1	0.5	0.64	1	3.14
0.1	0.88	0.5	0.64	0.9	2.92
0.2	0.42	0.42	0.5	0.9	2.24
0.3	0.2	0.2	0.2	0.2	0.80
0.4	0.8	0.5	0.8	0.8	2.90
0.5	1	0.5	0.8	1	3.30
0.6	0.8	0.5	0.8	0.8	2.90
0.7	0.2	0.2	0.2	0.2	0.80
0.8	0	0.2	0.5	0.5	1.20
0.9	0	0.2	0.8	0.9	1.90
1	0	0.2	0.8	1	2.00

Eq. 45 gives the membership values for different durations of the activity and frequencies of occurrence considering the total effect of the factors given in Table 1. According to Yao (22), a subset, D_a, can be chosen from Eq. 45 as a fuzzy representation of the duration of the activity. The membership value for each value of the duration of the activity in D_a is equal to the largest membership value in the corresponding column for that duration in Eq. 45. However, if Yao's method is used for this case, some of the information given in Eq. 45 would not be considered, i.e., the frequency of occurrence of the total effect of the factors. Therefore, the writers suggest the following method: choose from Eq. 45 a row (subset) which maximizes the product of the row summation given in Eq. 45 and the corresponding frequency. The last row of Eq. 45 gives the maximum value of this product for the problem under consideration. Therefore, the following fuzzy subset of the activity duration, D_a, is chosen:

$$D_a = (10|0.0,\ 15|0.2,\ 18|0.8,\ 20|1) \quad (46)$$

According to Zadah (25), the probability mass function of the duration activity can be calculated as follows:

$$P(d_a = 10) = \frac{0}{0.2 + 0.8 + 1} = 0 \quad P(d_a = 15) = \frac{0}{0.2 + 0.8 + 1} = 0.1$$

$$P(d_a = 18) = \frac{0.8}{0.2 + 0.8 + 1} = 0.4 \quad P(d_a = 20) = \frac{1}{0.2 + 0.8 + 1} = 0.5 \quad (47)$$

Therefore, estimates of the mean value, $\bar{d}_a$, and standard deviation of the duration, σ_{D_a}, of the activity duration and calculated as follows:

$$\bar{d}_a = 15 \times 0.1 + 18 \times 0.4 + 20 \times 0.5 = 18.7 \text{ days}$$

$$\sigma^2_{D_a} = 15^2 \times 0.1 + 18^2 \times 0.4 + 20^2 \times 0.5 - (18.7)^2 = 2.41$$

$$\sigma_{D_a} = 1.552 \text{ days}; \quad COV(D_a) = 0.083 \quad (48)$$

Similarly, for each activity in the construction project, the probability mass function, mean value and standard deviation of the duration can be calculated. Using the PERT method and the statistics of the activity duration as obtained by the proposed techniques, a project schedule can be determined. It is also possible to generate the duration of each activity as a random variable from the probability mass function calculated by the proposed technique for use in any simulation technique for project scheduling.

If Yao's method (22) is used, the following fuzzy subset of the activity duration, D_a, would result (refer to Eq. 45):

$$D_a = (10|1,\ 15|0.5,\ 18|0.8,\ 20|1) \quad (49)$$

It should be noted that $d_a = 10$ has a membership value equal to 1 although it is taken from a row corresponding to zero, or 0.5 frequency of occurrence. The probability mass function and the statistics of the duration for Eq. 49 would be

$$P(d_a = 10) = 0.303; \quad P(d_a = 15) = 0.152; \quad P(d_a = 18) = 0.242$$

$$P(d_a = 20) = 0.303; \quad \bar{d}_a = 15.726 \text{ days}; \quad \sigma_{D_a} = 4.1 \text{ days};$$

200

$$\text{and} \quad COV(D_a) = 0.261 \quad (50)$$

By comparing Eqs. 47 and 50, some important observations can be made. The duration of the activity considered here is expected to be between 10 and 20 days. According to Eq. 47, the probability mass function of the duration is a unimodal function with the modal value (the most probable value) of 20 days. However, according to Eq. 50 (Yao's method), the most probable values are 10 and 20 days which are at the ends of the range. It has a bimodal probability mass function. Any value within the range is less likely. From a practical point of view, this is very unlikely. In probabilistic project scheduling, it is common to model the duration of an activity by a unimodal probability mass function. Therefore, the proposed method might be more realistic than Yao's method.

The success in incorporating the impact of weather and labor skill depends on the assumptions used in translating the linguistic variables into fuzzy sets, i.e. Eqs. 26–31 and 40. The more this technique is used and compared with the actual impact of these factors on activity duration, the higher the level of success will be in choosing the proper membership values in the definition of the linguistic variable. It is possible to define different grades of any linguistic variable, e.g.

$$\text{Large 1} = (0.8|0.2,\ 0.9|0.8,\ 1|1)$$

$$\text{Large 2} = (0.8|0.3,\ 0.9|0.85,\ 1|1)$$

$$\text{Large 3} = (0.8|0.4,\ 0.9|0.9,\ 1|1)$$

$$\text{Large 4} = (0.8|0.5,\ 0.9|0.95,\ 1|1)$$

$$\text{Large 5} = (0.8|0.7,\ 0.9|0.99,\ 1|1) \quad (51)$$

or $\text{Large 1} = (0.8|0.2,\ 0.9|0.8,\ 1|1)$

$$\text{Large 2} = (\text{Large 1})^{0.9}$$

$$\text{Large 3} = (\text{Large 1})^{0.8}$$

$$\vdots$$

$$\text{Large 5} = (\text{Large 1})^{0.6} \quad (52)$$

or any other appropriate definition.

In order to evaluate the sensitivity of the proposed techniques to membership values in the definition of the linguistic variables, the example is solved again using different membership values in defining the fuzzy relation, R, between the consequences and the duration, i.e., Eq. 40. If Eq 40 is changed to

$$R{:}D_a = \text{very large} = (15|0.02,\ 18|0.81,\ 20|1), \quad \text{if} \quad C \text{ is large};$$

$$R{:}D_a = \text{large} = (15|0.15,\ 18|0.9,\ 20|1), \quad \text{if} \quad C \text{ is medium};$$

$$R{:}D_a = \text{small} = (18|0.2,\ 15|0.5,\ 10|1), \quad \text{if} \quad C \text{ is small} \quad (53)$$

then, according to the proposed method, D_a would be as follows:

$$D_a = (10|0,\ 15|0.15,\ 18|0.81,\ 20|1) \quad (54)$$

$$\text{and} \quad P(d_a = 15) = \frac{0.15}{0.15 + 0.81 + 1} = 0.077$$

201

$P(d_a = 18) = 0.413 \quad P(d_a = 20) = 0.510$ .... (55)

The statistics of D_a can be shown to be: $\bar{d}_a = 18.791$ days, $\sigma_{D_a} = 2.035$, and $COV(D_a) = 0.108$. By comparing Eqs. 46 and 54 and the corresponding statistics, it is clear that the proposed technique is not sensitive to small variations in the membership values.

If a different fuzzy relation, R, between the consequences and the duration is used, e.g.

$R{:}D_a$ = large = (15|0.2, 18|0.8, 20|1) if C is large;

$R{:}D_a$ = medium = (10|0.5, 15|0.9, 18|0.9, 20|0.5) if C is medium;

$R{:}D_a$ = small = (18|0.2, 15|0.5, 10|1) if C is small .... (56)

then, the following subset, D_a, can be obtained

D_a = (10|0.5, 15|0.9, 18|0.9, 20|0.5) .... (57)

and $P(d_a = 10) = 0.179$, $P(d_a = 20) = 0.179$, $P(d_a = 15) = 0.321$ ... (58)

and $P(d_a = 18) = 0.321$. The statistics of D_a corresponding to Eq. 58 can be estimated as: $\bar{d}_a = 15.964$ days; $\sigma_{D_a} = 3.30$ days, and $COV(D_a) = 0.2068$.

Eq. 57 is quite different than Eqs. 46 and 54. Therefore, the proposed technique is sensitive to the choice of the fuzzy relation, R, between the consequence and the duration of the activity.

Conclusions

Different probabilistic methods with various degrees of complexity are being used in construction engineering. However, when a parameter is expressed in linguistic rather than mathematical terms, classical probability theory fails to incorporate the information. The linguistic variables can be translated into mathematical measures by fuzzy sets and systems theory. A construction management problem is solved in this paper using this concept. It is expected that this concept will be used in similar construction management problems in the future. In order to implement the proposed technique, various membership functions need to be estimated, which could be difficult in some cases. However, they could be estimated with the assistance of experts, and the information can be refined as this method is used more frequently.

It is observed here that the proposed technique is not sensitive to small variations in the membership values. This is a very desirable property. However, the method is sensitive to the choice of the fuzzy relation between the consequences and the duration of an activity. This is expected. This relation could be modified with more applications in various projects.

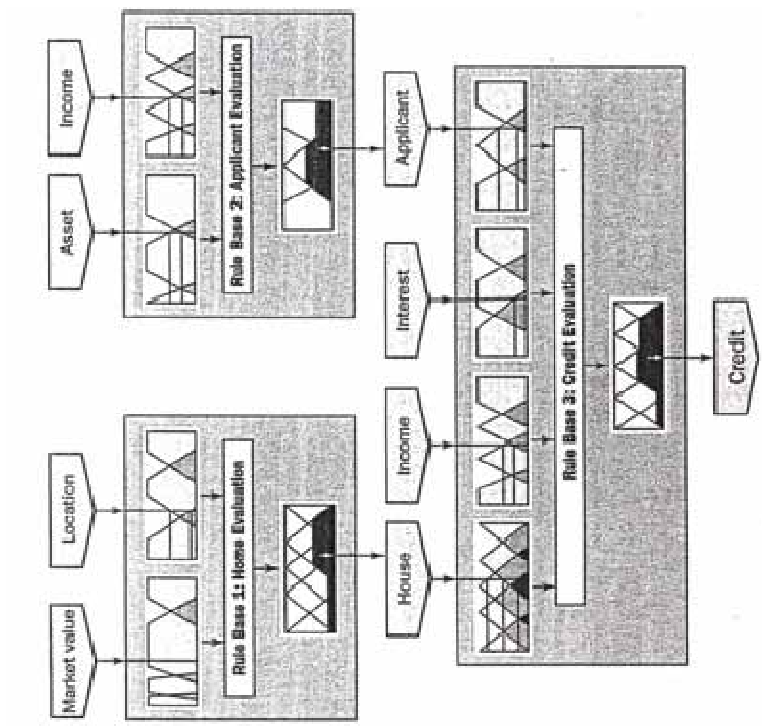
The uncertainty in the fuzzy relations needs to be transformed in such a way that it can be used with other sources of uncertainties obtained by using classical statistical methods. Several methods can be used for this purpose. A new method is proposed for the problem under consideration that maximizes the product of the sum of the membership associations for a certain frequency of occurrence and the corresponding frequency of occurrence. It is expected to be superior to other available

methods since it utilizes all the available information. One of the main advantages of the proposed technique is that it can be easily implemented in existing computer programs for project scheduling.

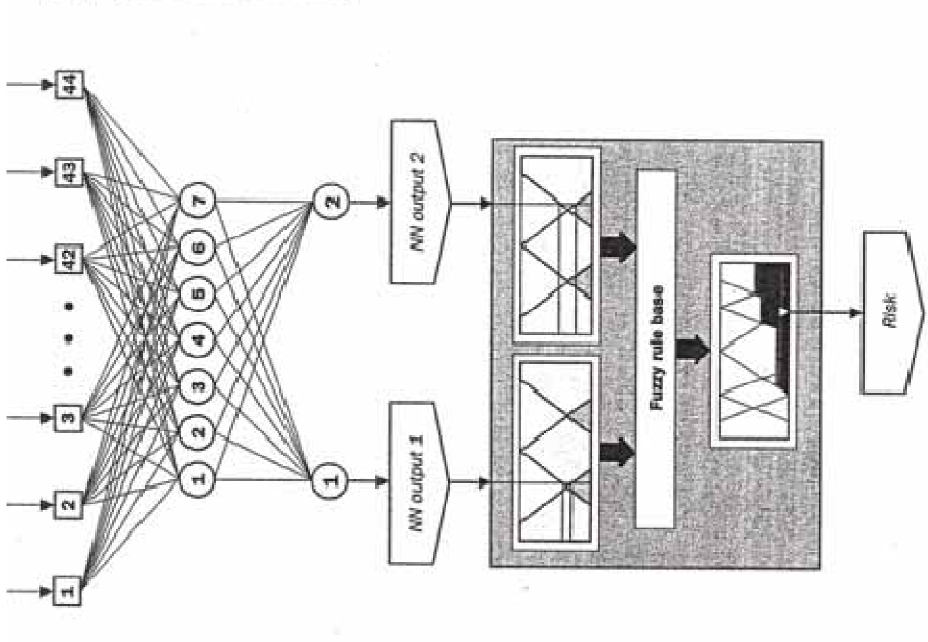
Appendix.—References

1. Ahuja, H. N., *Construction Performance Control by Network*, John Wiley & Sons, Inc., New York, N.Y., 1976, pp. 23–126.
2. Ang, A. H-S., and Tang, W. H., *Probability Concepts in Engineering Planning Design*, Vol. 1, John Wiley & Sons, Inc., New York, N.Y., 1975.
3. Anonymous, PERT Summary Report, *Phase I*, Special Project Office, Bureau of Naval Weapons, Department of Navy, Washington, D.C., July, 1958.
4. Ayyub, B. M., and Haldar, A., "Practical Structural Reliability Techniques," *Journal of Structural Engineering*, ASCE (to be published).
5. Benjamin, N. B. H., and Davis, C., "The Impact of Weather on Construction Planning," presented at the October 18–22, 1971, ASCE Annual and National Environmental Engineering Meeting, held at St. Louis, Mo. (Preprint 1551).
6. Benjamin, B. H., and Greenwald, T. W., "Simulating Effects of Weather on Construction," *Journal of the Structural Engineering Division*, ASCE, Vol. 99, No. CO1, Proc. Paper 9888, July, 1973, pp. 389–404.
7. Blockley, D. I., "Analysis of Structural Failures," *Proceedings*, Institution of Civil Engineers, Vol. 62, Part 1, Feb., 1977.
8. Blockley, D. I., "Predicting the Likelihood of Structural Accidents," *Proceedings*, Institution of Civil Engineers, Vol. 59, Part 2, Dec., 1975.
9. Blockley, D. I., "The Role of Fuzzy Sets in Civil Engineering," *Fuzzy Sets and Systems*, Vol. 2, North-Holland Publishing Co., Amsterdam, The Netherlands, 1979, pp. 267–278.
10. Brown, C. B., "A Fuzzy Safety Measure," *Journal of the Engineering Mechanics Division*, ASCE, Vol. 105, Proc. Paper 14915, Oct., 1979, pp. 855–872.
11. Buchert, K. P., "Use of Weather Information in Planning Shell Construction," *Proceedings*, International Association of Shell Structures, Conference on Shell Structures and Climatic Conditions, July, 1972, pp. 95–102.
12. Davis, C. S., "Construction Project Simulation Effects of Weather," thesis presented to the University of Missouri, at Columbia, Mo., in 1972, in partial fulfillment of the requirements for the degree of Master of Science.
13. Fazar, W., "The Origin of PERT," *The Controller*, Vol. 30, Dec., 1962, p. 598.
14. Haldar, A., "Statistical Methods," *Numerical Methods in Geomechanics*, J. B. Martins, ed., NATO Advanced Study Institute Series, D. Reidel Publishing Company, Boston, Mass., 1981, pp. 473–506.
15. Halpin, D. W., and Woodhead, R. W., *Construction Management*, John Wiley & Sons, Inc., New York, N.Y., 1980, pp. 313–337.
16. Ishizuka, M., Fu, K. S., and Yao, J. T. P., "A Rule-Inference Method for Damage Assessment," ASCE, St. Louis, Mo., Oct., 1981, (Preprint 81-502).
17. Kaufmann, A., *Introduction to the Theory of Fuzzy Subsets*, Academic Press, New York, N.Y., 1975.
18. Kelley, J. E., Jr., "Critical Path Planning and Scheduling: Mathematical Basis," *Operations Research*, Vol. 9, No. 3, 1961, pp. 296–320.
19. Miller, R. W., *Schedule Cost and Profit Control with PERT*, McGraw-Hill Book Co., New York, N.Y., 1963.
20. O'Brien, J. J., *CPM in Construction Management*, McGraw-Hill Book Co., New York, N.Y., 1965.
21. O'Brien, J. J., (ed.), *Scheduling Handbook*, McGraw-Hill Book Co., New York, N.Y., 1969.
22. Yao, J. T. P., "Damage Assessment of Existing Structures," *Journal of the Engineering Mechanics Division*, ASCE, Vol. 106, No. EM4, Proc. Paper 15653, Aug., 1980, pp. 785–799.
23. Zadeh, L. A., "Fuzzy Sets," *Information and Control*, Vol. 8, 1965, pp. 338–353.

Hybrid Applications:



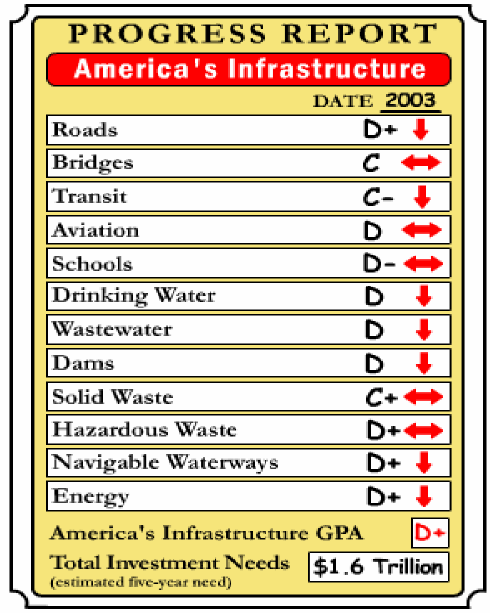
Hierarchical fuzzy model for mortgage loan assessment

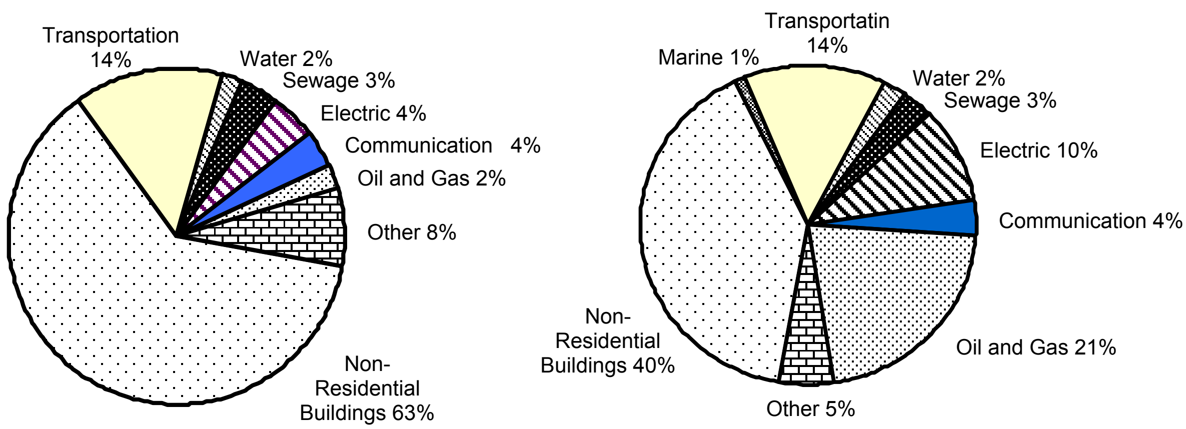


Hierarchical structure of the neuro-fuzzy system for risk assessment

Infrastructure Asset Management

While the civil infrastructure is the foundation for economic growth, a large percentage of its assets are rapidly deteriorating due to age, aggressive environment, and insufficient capacity for population growth. In 2003, the American Society of Civil Engineers released a report card on the infrastructures in the USA that gave failing grades to many infrastructure systems, and identified the need for $1.6 trillion (US) to bring the assets to acceptable condition (ASCE 2003). Similarly, the environmental, social, and transportation infrastructure systems in Canada require huge investments that amount to approximately $10 billion (US) annually for 10 years (Federation of Canadian Municipalities 1999). Since the environmental, social, and transportation sectors represent about 25% of the Canadian infrastructure expenditures (Figure1, Statistics Canada 1995), it can be assumed that the infrastructure system as whole requires an investment of about $40 billion per year for ten years. Despite of this large need, the Infrastructure Canada Program allocated only $2 billion (US) for the year 2000 to all infrastructure sectors (Federation of Canadian Municipalities 2001), thus covering only about 5% of the need. With the non-residential buildings being largest sector of the infrastructure (approximately 40%), such sector is expected to suffer the largest shortfall in expenditures on rehabilitation and repair.





a) USA b) Canada

Average yearly expenditures by type of infrastructure

Important Questions:

- What assets do you own?
- What is it worth of each asset?
- What is its current condition of asset components?
- What is the remaining service life of the asset?
- What is the predicted condition in the future?
- What do you fix first?
- How do you fix it?
- What is the condition after the repair?
- How will you execute the many repairs?

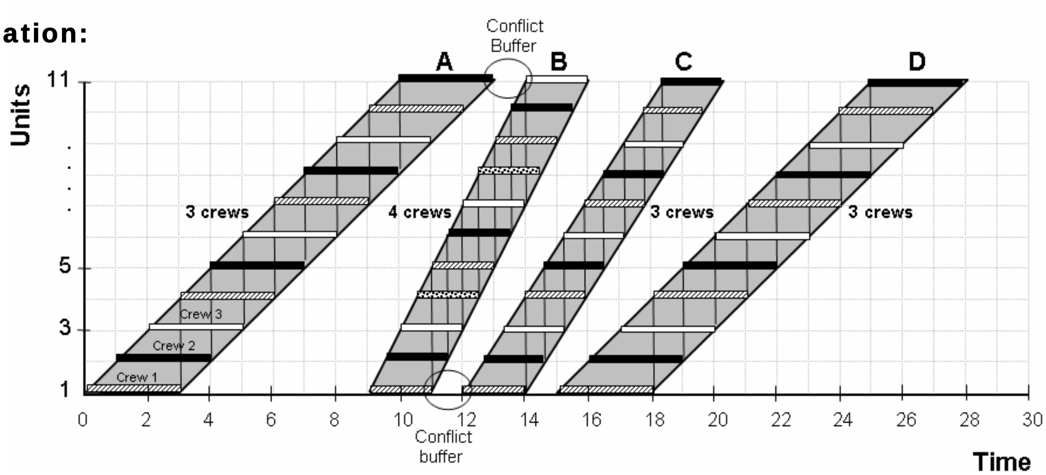
Important references: Please download from the course web page.

Scheduling Repetitive & Linear Projects

- Problems with CPM & PDM
- Resource-Driven Scheduling
- Crew Work Continuity
- Learning Phenomenon

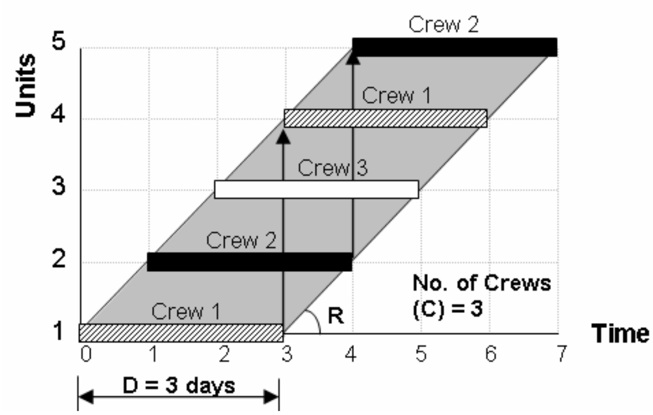
Integrated CPM & LOB Calculations:

New Representation:

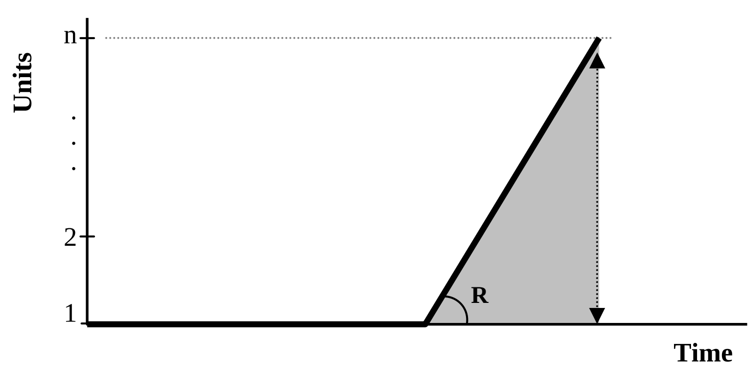


Crew Synchronization Calculations:

Crews (**C**) = (**D**) x (**R**)

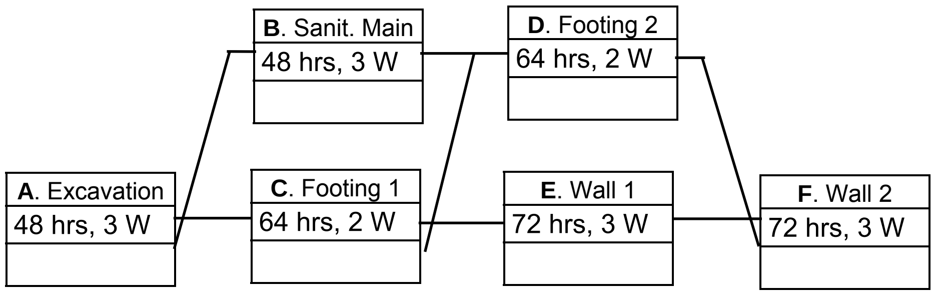


Calculating a Desired Progress Rate (R):



Example:

For this small project, the work hours and the number of workers for each activity are shown. if you are to construct these tasks for 5 houses in 21 days, calculate the number of crews that need in each activity. Draw the schedule and show when each crew enters and leaves the site;

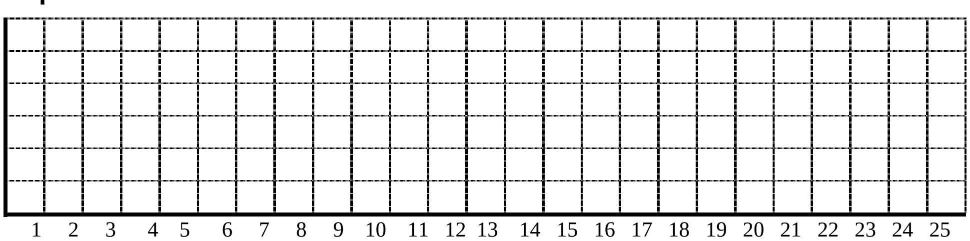


Step 1: CPM Calculation

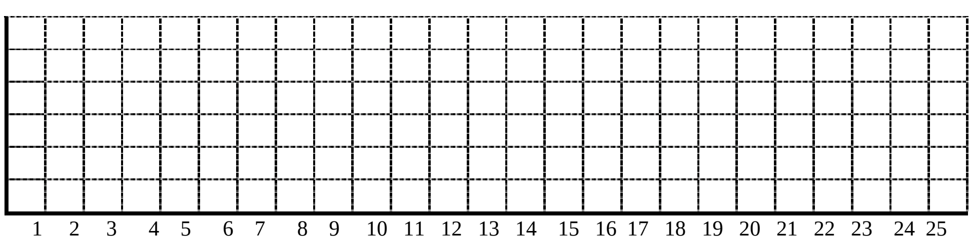
Step 2: LOB Calculations Deadline T_L = 21; T_1 = ___ ; n = 5

Activity	Duration (D)	Total Float (TF)	Desired Rate (R) $(n-1) / (T_L - T_1 + TF)$	Min. Crews (C) = D x R	Actual Crews (C_a)	Actual Rate (R_a) = C_a / D
A						
B						
C						
D						
E						
F						

Step 3: Draw the Chart



Draw the critical path



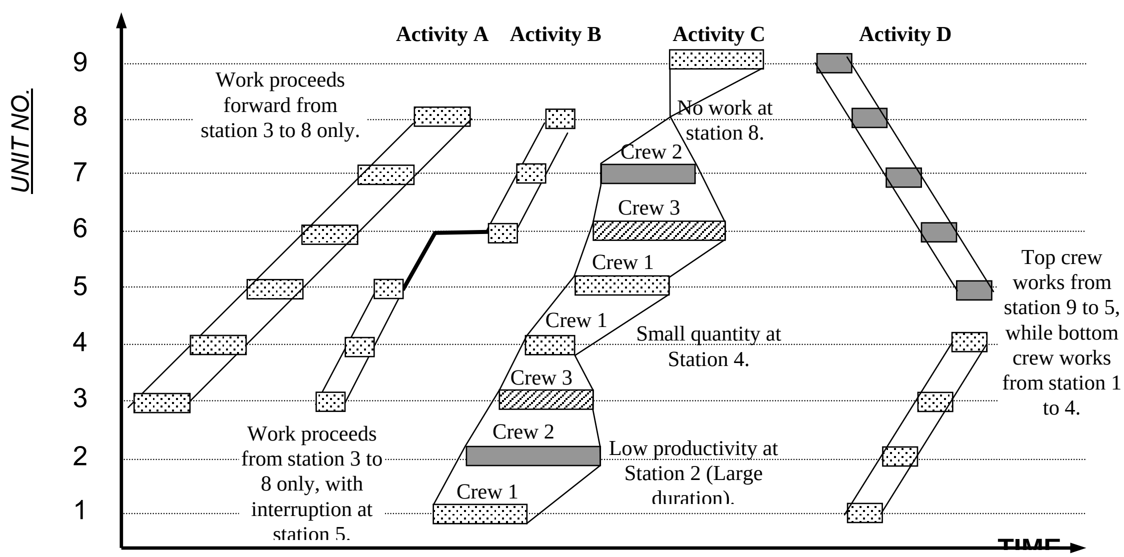
Assume:

- Same no. of Crews
- Activity A in unit 2 has double the duration
- Unit 4 does not need excavation.

More Advanced Linear scheduling Model

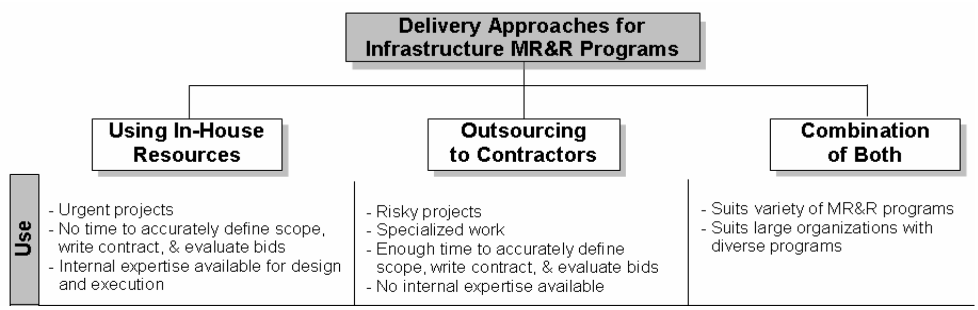
Flexible features for scheduling the activities include: color-coded or pattern-coded crews; varying quantities; productivity impact; crew interruption time; crew staggering; crew work sequence; and activities' progress speeds (slopes of lines). It is noted that the schedule is efficiently arranged with crew work continuity maintained. Also, overlapping is avoided by simply showing the activities of each path in the work network separately. In addition:

1. Activities are not necessarily repeated at all sections.
2. Activities can proceed in an ascending or descending flow. This provides work flow flexibility and provides for a way to fast-track projects;
3. Each activity has up to 3 methods of construction (e.g., normal work, overtime, or subcontractor) with associated time, cost, and crew constraints. The model can then be used to select the proper combination of methods that meet the deadline, cost, and crew constraints;
4. Activities can have non-standard durations and costs at selected sections;
5. Work interruption (layoff period) can be specified by the user at any unit of any activity; and
6. Conditional methods of construction can be specified by the user.



Infrastructure Networks with Distributed Sites: A Bigger Challenge

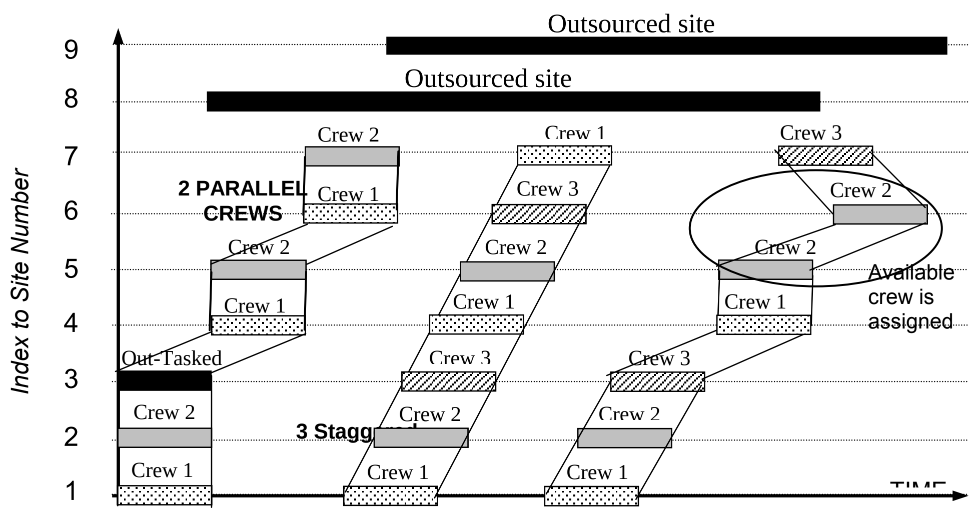
Buildings, Hispitals, Schools, Highway Spots, Bridges



Delivery approaches for MR&R programs

Effect of Site order

Scheduling of crews along multiple sites



Project Control

Organize site? Recording of site events? Work Status? Comparing Planned versus Actual? Progress Payments? Managing Changes? Updating? Corrective Actions? Delay Resposibility? Forecasting/ Time Extension? Cost compensation? Productivity Assessment? Saving All As-Built Details? Lessons Learned?

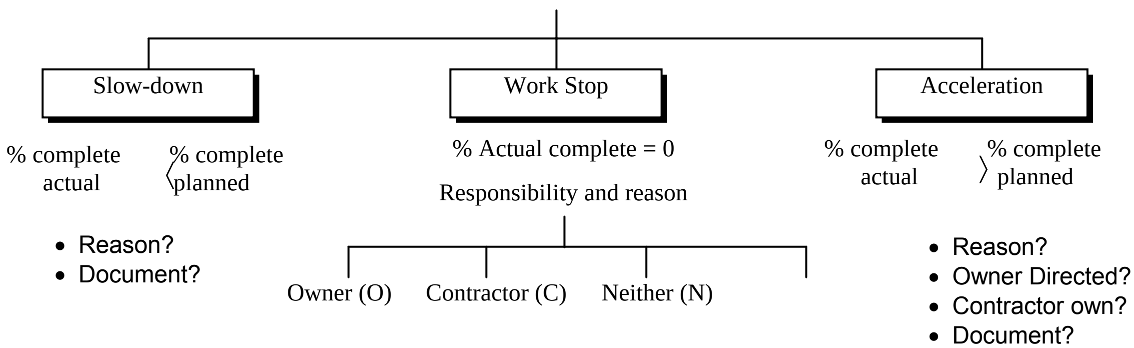
a) **Organized Site** = Safety + Productivity + Good Circulation + Cost & Time Savings

(1) identifying necessary facilities and determining their appropriate sizes;
(2) determining the inter-relationships among the facilities on the site; and
(3) optimizing the placement of the facilities on the site plan.

b) **Recording Site events**

Calculate activity % complete, Camcorders, Time-Lapse Camera, Minutes, **Project Web Site**

Activity % Complete

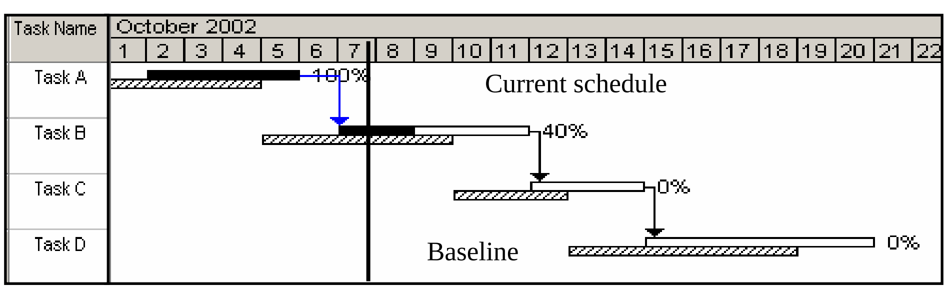


Calculating activity % complete: pages 291& 292

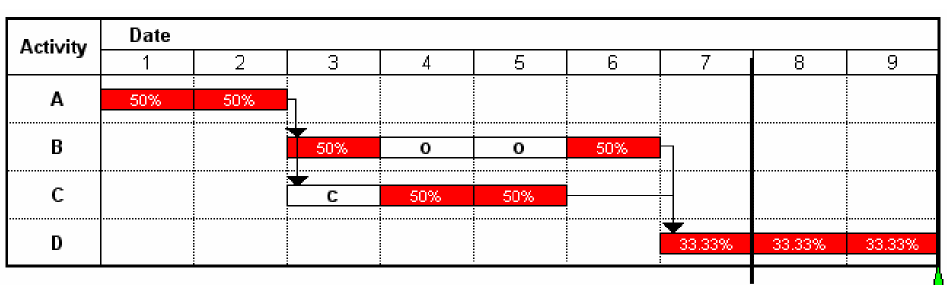
Calculating the overall project % complete: page 293

c) **Using Software**

How to show Delays?
Slow versus Fast?
Reasons for work stops?



Can we readily decide which party is responsible for the two days delay beyond the deadline?

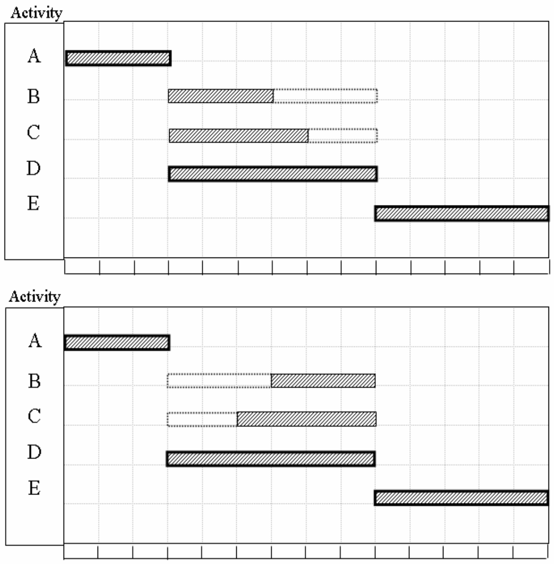


d) **Techniques for Performance Evaluation**

1. S-Curve Envelope:

Contractor's cost Control

Early versus Late bar chart

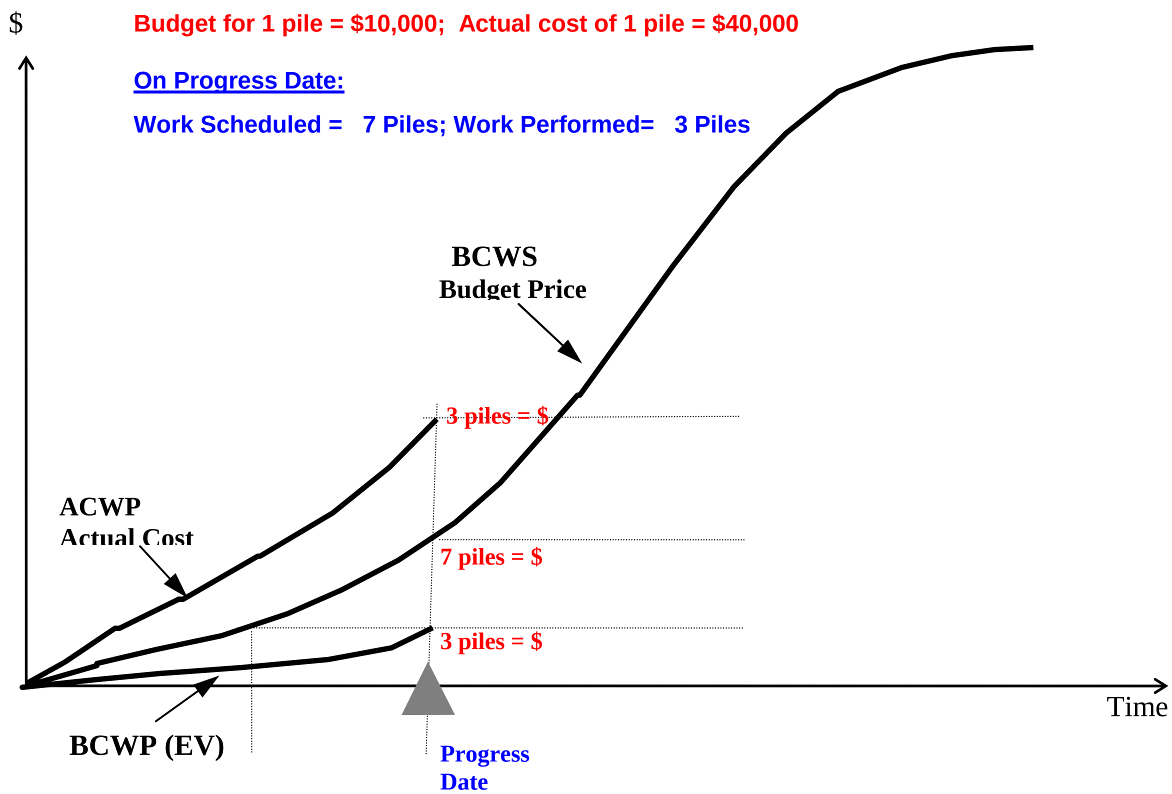


2. Earned-Value Analysis:

Budget for 1 pile = $10,000; Actual cost of 1 pile = $40,000

On Progress Date:

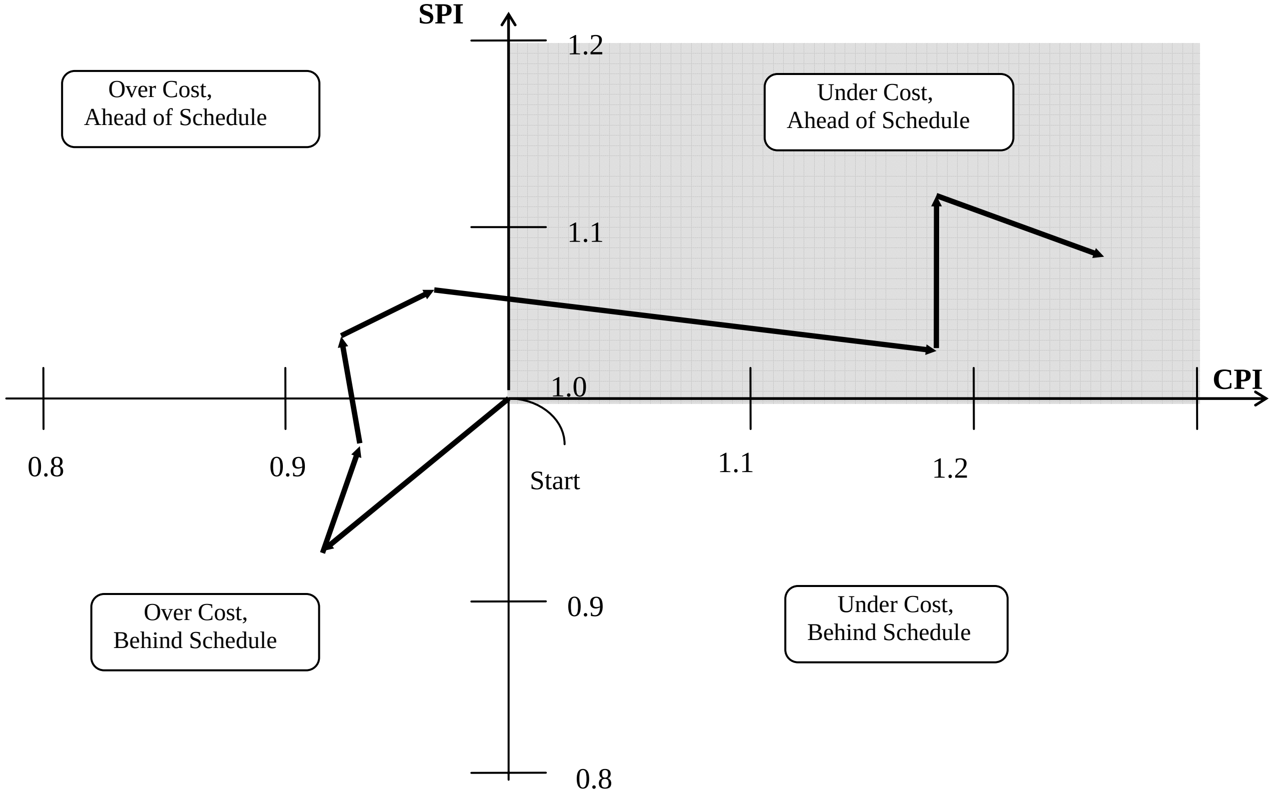
Work Scheduled = 7 Piles; Work Performed= 3 Piles



Schedule Performance Index (SPI) $= \dfrac{\textbf{BCWP}}{\textbf{BCWS}}$

- Time variance?
- Estimate at completion?

Cost Performance Index (CPI) $= \dfrac{\textbf{BCWP}}{\textbf{ACWP}}$



e) **Agenda for Success:**

- Get Good Designers: Beware of Bargain Shopping;
- Watch Low Bids Carefully: Work at Cost Spells Trouble;
- Fail to Plan and you Plan to Fail;
- Keep the Work Site Organized;
- Monitor the Gaps;
- No Pay Causes Delay;
- Time = Money;
- Communication; and Documentation.

f) **New Concept For Project Control (Critical Chain):**

- Estimate with safety removed (50% chance);
- Incentive for early finish;
- Focus on predecessors' finish;
- Project buffer (50%);
- Simple monitoring of buffer penetration;
- Earned-Value for cost analysis.