FACULTY OF BUSINESS

RESEARCH AND WORKING PAPER SERIES

McMASTER UNIVERSITY

1280 Main Street West
Hamilton, Ontario, Canada L8S 4M4
Telephone (416) 525-9140

EXPIM: A Knowledge-Based Expert System for Production / Inventory Modelling

by

Mahmut Parlar
Faculty of Business
McMaster University
Hamilton, Ontario, L8S 4M4
Canada

Working Paper #283

August 1987

EXPIM: A Knowledge-Based Expert System for Production / Inventory Modelling

Mahmut Parlar
Faculty of Business
McMaster University
Hamilton, Ontario, L8S 4M4
Canada

August 1987

(EXPERT SYSTEMS; MODELLING; PRODUCTION/INVENTORY)

Abstract

This paper discusses the development of a knowledge based expert system (EXPIM) which can identify and recommend up to 30 production-inventory models. The user (manager) is asked a sequence of questions regarding the problem situation. The system uses backward-chaining to identify what is the correct model to use given the situation described by the user. The paper also discusses the general methodology for building expert systems in management science for the purpose of identifying models in other areas, such as queueing theory and location theory.

1. Introduction

Management scientists and industrial engineers have, in the past forty years, produced hundreds of production-inventory models ranging from the simple extensions of classical EOQ (Hadley and Whitin, 1963) to complex multi-level control systems (Schwarz, 1981). Two decades ago, Eilon and Lampkin (1968) summarized more than 500 papers in inventory control developed and/or published between 1953 and 1965. This proliferation of production-inventory (PI) models have prompted many researchers to publish surveys of "current" status of these models in particular and inventory theory in general at a rate of approximately two per per decade (Veinott 1966; Iglehart 1967; Aggarwal 1974; Nahmias 1978; Wagner 1980; Silver 1981.) With every new review paper, the readers were informed about recent developments in inventory modelling and possible extensions of the current models along with a "wish list" of future research activities. Clearly, all these review articles written by the experts in their field have helped the research community get an excellent overview of inventory modelling and theory, although they probably have not helped the actual users of inventory models very much

In recent years, some researchers have started investigating the possibility of classifying the inventory systems. In particular, Hollier and Vrat (1978) have used a structure similar to that proposed by Kendall (1952) for queueing models. They have attempted to describe any given inventory system using four aspects, i.e. structure, environmental parameters, operating policies and costs. For example an inventory system with Erlang demand, exponential replenishment time, general shelf-life distribution and complete backlogging was symbolized as $E_1/M/G/\infty$. Given the apparent duality between queueing and inventory processes (Prabhu 1965) this classification seemed to be a natural one to adopt for inventory systems. But, it appears that this method has not caught on and no other published papers seem to refer to it.

Recently Menipaz (1982) has proposed another classification of inventory models based on a systematic use of twelve basic assumptions. With all the assumptions holding, one obtains the classical EOQ, but as the assumptions are relaxed a better approximation to the problem is obtained.

In a major research effort started in 1976 and supported by the Hungarian Academy of Sciences (Barancsi et al. 1980; Chikan et al. 1982; and Barancsi et al. 1983) another attempt is made to classify inventory models. These researchers' objective was to make the many of the currently available inventory models more useful for the practitioners. By comparing a code system with 45 codes each having between two to ten possible values they have classified all these models with the purpose of providing a manager with a model that would be most suitable for his problem.

Given the fact that majority of inventory (and management science/operations research) models are never used in implementation of their results, the correct choice of these models becomes very important. A manager may have several inventory models available to him, but if he is not sure which is the right one for the given situation obviously he will not be able to solve his problem with the wrong model. The use of classical EOQ in many inventory situations although it is the wrong model is an often repeated example. The importance of selecting the right (inventory) model and an anecdote related to that issue is given in Ravindran et al. (1987, pp. 369-370).

It is clear that when a manager is faced with a PI problem and he has an expert available for choosing the right model (and perhaps solving it) then he can be confident that the efforts put into analysis will not be wasted. But, as in many discipline, experts in inventory modelling are also scarce and expensive. In recent years, the emergence of the field of expert systems (ES) as a branch of artificial intelligence (AI) has provided some hope for increasing the availability of expert knowledge, albeit in a computerized form.

The premise of this paper is the following. Although the attempts to create systems for classifying inventory models as discussed in the paragraphs above may have provided useful information for decision makers who want to use them, these systems do not have expertise. They all seem to lack an ability to provide explanations of their line of reasoning, i.e. how did they decide to recommend the use of, say, the newsboy model. Except possibly for the Hungarian system, they lack flexibility, i.e. integration of new knowledge incrementally into its existing store of knowledge. We attempt to go one step farther, and build a knowledge based expert computer system which can accurately choose a PI model after consulting the user (manager) by asking him a minimum number of questions pertinent to the problem situation. Our system, named EXPIM (for EXpert Production/Inventory Modeller) has the ability to explain WHY it is asking a particular question and also HOW it reached a conclusion and recommended a particular model. By incorporating uncertainties of both the human expert in his reasoning, and the manager in his responses, EXPIM can also recommend models with reduced confidence factors and can warn the manager of this fact. It also has the ability to automatically run a program once a model choice is made and solve the problem numerically. It permits sensitivity analysis by letting the manager change one or more of his responses to system's queries and come with (possibly) different model recommendations. So far, EXPIM can identify approximately 30 models ranging from the simplest EOQ to more complex coordinated replenishment models. EXPIM runs on the widely available IBM-PC microcomputer (or compatibles) with 512K memory.

An expert system, such as EXPIM, is a computer program whose behaviour duplicates, in some sense, the abilities of a human expert in his area of expertise. Expert systems have been successfully developed in fields such as medical diagnosis (Shortliffe 1976), mineral exploration (Duda et al. 1979) and chemical data interpretation (Lindsay et al. 1980). Excellent reviews of expert systems and their principles have been provided elsewhere (Buchanan and Duda 1983; Stefik et al. 1982; Hayes-Roth et al. 1983; Waterman 1986) and the reader is referred to these sources for detailed descriptions.

In recent years, we have also witnessed an interest in the MS/OR community towards AI/ES and this is exemplified in the large number of papers presented on these topics in TIMS/ORSA Joint National Meetings. A few papers have also been published reviewing AI/ES concepts for the management science community and providing possible research topics for interfacing expert systems and management science (Kastner and Hong 1984; Assad and Golden 1986). Other applications of ES in traditional MS/OR are reported in the areas of design optimization (Azarm and Pecht 1985), university admission decisions (vanBreda 1986), simulation modelling (Khoshnevis and Chan 1986; Doukidis and Paul 1985) and operations analysts (Biswas et al. 1987).

In Assad and Golden (1986) the authors present some possible directions for using ES in management science. These include i) choice of models which is what EXPIM does and ii) model generation where systems queries prompt the user to add constraints and formulate the objective function in an optimization problem. Binbasioglu and Jarke (1986) provide a discussion of such a model generation in linear programming using an AI language called Prolog.

In an important article, Hahn (1985) discusses the interface between statistical methods and expert systems. As the statistical computer programs are becoming increasingly available to people who are not statistical experts, he proposes that expert statistical systems be constructed which would embed expert guidance in statistical programs. Some programs that run on IBM-PCs which do just that are already available for expert forecasting (Wisard 1986).

To summarize, we see that problems which could previously be solved by human experts only, are now being attacked by using the expertise embedded in expert computer systems. Provided that these systems are constructed properly and are thoroughly tested and validated, they will be useful in many areas of management science as the above examples illustrated.

As the development of these systems frequently require the use of a specialized AI language or an expert system shell, in the next section we will describe the shell we used in our research.

In Section 3, the methodology we used in the development of EXPIM will be discussed, where we will try to explain the steps that should be followed in developing systems such as EXPIM. Since there are many other areas in management science (location theory, queueing theory) which may benefit from expert system technology, our discussion may be useful for the developers of such systems.

Section 4 contains a discussion of the experience gained while structuring and building this expert system. Last Section provides a summary, suggestions for refinements of the model and some possible avenues for further research in this new area.

2. The Expert System Shell

Expert systems can be developed in almost any computer language, including FORTRAN (Weiss and Kulikowski 1984) and any of the popular AI languages LISP and Prolog. When a computer language is used the developer (known in the AI circles as the "knowledge engineer") has more control over the flow of logic, search mechanism etc., but the development time may be very long. On the other hand, using a readily available expert system shell reduces the development time, but the knowledge engineer limits himself to work within the previously determined confines of the shell someone else has written.

Since our objective in this research was to produce an expert system which should be easily available, we chose Texas Instruments Personal Consultant Easy as our development environment which runs on IBM-PC. After examining a few other shells we decided to use PC Easy because of its IBM compatibility, facility of development, sensitivity analysis feature, explanation facility in English language and in corporation of certainty factors.

PC Easy is a production rule based shell where the knowledge base consists of IF-THEN type rules which are chained to each other. Since these production systems were first proposed by Post (1943) as a general computational mechanism they have been applied to wide variety of problems

(Davis and King 1977) including the expert systems. A rule in PC Easy specifies a deduction that can be made in a particular situation, and the premise (IF) and action (THEN) parts can be composed of compound statements connected with _and_ and _or_. For example;

<u>RULE011</u>

If 1) number of products considered is multiple, and

2) supplier is the same for all the items purchased, and

3) mode of transportation is same

Then It is definite that coordination of replenishments is possible.

is a rule which tests whether coordination is possible in an inventory situation.

PC Easy uses a combination of backward-chaining and forward-chaining control strategies. The former, also known as the goal-directed control (Barr and Feigenbaum 1981) reduces the number of questions the user has to answer and eliminates quickly many of the irrelevant choices. Forward chaining, or data-driven control can be used to trigger actions based on special conditions. (Appendix discusses an example of these two control strategies.)

When the system reaches a conclusion regarding the model to choose, the user can ask HOW this conclusion was reached. The explanation facility provides the response by referring to the rule it has used to reach the conclusion, and prints the English translation of that rule.

Sensitivity analysis is also easily done using the REVIEW feature, where the user is given the option to change one or more of his responses which he provided during the consultation. This may be an important factor in problems where conditions may change from time to time.

PC Easy supports certainty factors (CF) which are numerical values indicating a measure of confidence in the value of a parameter. These factors let a knowledge base deal with the reality that facts and opinions are sometimes known with less than absolute certainty. Certainty factors can be incorporated into PC Easy during development and during consultation when the user may feel a doubt in responding to a prompt.

With this brief introduction to PC Easy's features, we now describe in some detail the methodology used in developing EXPIM.

3. Development of EXPIM

An expert system such as EXPIM consists of three elements: the knowledge base, the inference engine and the user interface (Feigenbaum and McCorduck 1983). The knowledge base contains the knowledge extracted from the expert by the "knowledge engineer" who is usually a computer scientist trained in artificial intelligence or someone who is well informed about the expert systems technology and the particular tool he is using. Inference engine part of the system decides which portions of the knowledge base apply to a given problem and what additional data must be supplied by the user. The inference engine also provides explanations of the system's behaviour, such as responding to the WHY and HOW questions asked by the user. Finally, the user interface consists of tools that can translate the questions asked or conclusions reached into plain English.

Currently, there are several methods available for constructing expert systems, e.g. rule-based deduction, statistical patterns classification, etc. (Assad and Golden 1986). As discussed in Reggia and Ahuja (1986), rule-based deductive expert systems should be employed when i) the underlying knowledge is already organized as (IF-THEN) rules and ii) the type of classification is predominantly categorical. The domain of knowledge, i.e. inventory modelling does satisfy these two requirements. Many books on inventory management (Silver and Peterson 1985) give long lists of assumptions leading to the particular model under discussion. For example, the assumptions leading to the basic EOQ model are listed as i) the demand rate is constant and deterministic, ii) the unit variable cost does not depend on replenishment quantity, etc. which can easily be put in an IF-THEN rule. Also, the classification of the inventory models is categorical in the sense that every inventory model can be placed in a particular model group after the user responding to a series of questions with two or at most three alternative answers. For these reasons, we decided to use a rule-based system and adopted PC Easy as our expert system shell

The rules in PC Easy are constructed using parameters which are structures that identify or contain a piece of information that the system uses to arrive at a conclusion. For example, the parameter QUANTITY-DISCOUNT contains information about the applicability of quantity discounts.

We have identified twenty three parameters with which all the rules in the knowledge base are constructed. These parameters, in their abbreviated form are: COORDINATION, DEMAND-LEVEL, DEMAND-PROCESS, DISCOUNT-TYPE, EOQ-MODEL, FIXED-COST, GENERAL-TYPE, INFLATION, LEAD-TIME, LIFE, MODEL, MONITORING, NUMBER-OR-PRODUCTS, QUANTITY-DISCOUNT, REFERENCE-LIST-ARTICLES, REFERENCE-LIST-BOOKS, REPLENISHMENT, SHORTAGES, SPACE, SUBSTITUTE, SUPPLIERS, TRANSPORT and UNIT-VARIABLE-COST. For example, the parameter DEMAND-LEVEL has the following properties as used by PC Easy.

<u>DEMAND-LEVEL</u> (PARAMETER)

TRANSLATION: the level of the future demand

PROMPT: What is the level of the future forecasted demand?

HELP: :tab 2 "If the forecasted future level of demand is basically constant (uniform over time), choose CONSTANT-OVER-TIME." : line 1 :tab 2 "If the forecast indicates lumpy demand where demand varies from time to time, choose VARIABLE-OVER-TIME." :line 1 :tab 2 "Refer to [SIL-PET], pp. 237-239, Section 6.6.4. for an explanation of a numerical technique which can distinguish between these two cases." :line 2

COMMENTS: ASKED

TYPE: SINGLEVALUED

EXPECT: CONSTANT-OVER-TIME VARIABLE-OVER-TIME

USED-BY: RULE003

When PC Easy translates rules into English it uses the TRANSLATION property which the system developer supplies during expert system construction. If at a particular stage, the system needs to know the type of DEMAND-LEVEL, it uses the PROMPT "What is the level of the future forecasted demand?", and prompts the user to choose between CONSTANT-OVER-TIME and VARIABLE-OVER-TIME. If the user is not exactly sure what the question is asking, he may obtain further HELP by pressing a function key and get an explanation as displayed in Figure 1.

\- - - - - - - - - - - - -

Insert Figure 1 here

\- - - - - - - - - - - - -

Sometimes, the values of some parameters are determined internally by the system. This may be necessary when the user may not be able to answer questions very easily pertaining to these parameters. For example, following rule determines in an indirect manner, whether the general model type is EOQ:

<u>RULE004</u> (INFERRING-RULES)

If 1) the forecast for the future demand is DETERMINISTIC, and

2) the level of the future demand is CONSTANT-OVER-TIME,

Then it is definite (100%) that general model type is EOQ.

IF: DEMAND-PROCESS = DETERMINISTIC AND DEMAND-LEVEL = CONSTANT-OVER-TIME

THEN: GENERAL-TYPE = EOQ

DESCRIPTION: General type EOQ.

This is achieved in an indirect way, by asking the user two easier to answer questions regarding the demand for the product under consideration.

Three of the parameters in the above list, i.e. MODEL, REFERENCE-LIST-BOOKS, REFERENCE-LIST-ARTICLES are the goals in the knowledge base which the PC Easy tries to prove. Using the backward-chaining inference mechanism described in the Appendix, system tries to prove the above three goals, one by one. When the goal MODEL is proven, the other two are also automatically proven and the system reaches a conclusion, and lists all the books and articles used in the knowledge base for the user's information.

In its present version, EXPIM has 68 rules with 12 rule groups constructed for ease of development. These rule groups are:

i) EOQ Related Groups

1. EOQ and Extensions
2. "Not Classical EOQ"
3. EOQ Possibilities

ii) 4. Deterministic Non-EOQ

iii) Newsboy Model Related Groups

5. Newsboy
6. Newsboy Possibilities

iv) Stochastic "Non-Newsboy" Groups

7. Continuous Review
8. Continuous Review Possibilities
9. Periodic Review
10. Periodic Review Possibilities

v) Auxiliary Rule Groups

11. Inferring
12. Conclusions

The first rule group has 14 rules each of which giving rise to an EOQ type model ranging from the classical EOQ to production lot size with lost sales. As an example the following Rule #5, if satisfied, recommends the classical EOQ model with a message to the user warning him to check his assumption before using this model. The rule also provides a published reference for this model along with the name of the basic language computer program (EOQ.BAS) the user can run after the consultation is over. (The programs are provided with the knowledge base).

RULE005 [EOQ-AND-EXTENSIONS-RULES]

If
1) general model type is EOQ, and
2) lead-time is ZERO, and
3) number of products considered is ONE, and
4) inventory shortage is NOT-PERMISSIBLE, and
5) replenishment rate of the order <or production quantity> is ALL-AT-ONCE, and
6) amount paid to the supplier for each unit purchased is INDEPENDENT-OF-ORDER-QUANTITY and
7) 1) future inflation expectations is LOW, or
   2) future inflation expectations is STABLE, and
8) item's lifetime is SUFFICIENTLY-LONG, and
9) storage space availability is SUFFICIENT, and
10) coordination of replenishments is NOT-FEASIBLE,

Then it is definite (100%) that the model you should use is Classical EOQ. This is the oldest inventory model available in inventory literature. Because of its simplicity in computations, it has frequently been misused. It is a good idea to re-check your assumptions before adopting this model. REFERENCE: [SIL-PET], pp. 174-180. Sections 5.1.5.3. COMPUTER PROGRAM: EOQ.BAS

IF GENERAL-TYPE = EOQ AND LEAD-TIME = ZERO AND NUMBER-OF-PRODUCTS = ONE AND SHORTAGES = NOT-PERMISSIBLE AND REPLENISHMENT = ALL-AT-ONCE AND UNIT-VARIABLE-COST = INDEPENDENT-OF-ORDER-QUANTITY AND

INFLATION = LOW OR INFLATION = STABLE AND LIFE = SUFFICIENTLY-LONG AND SPACE = SUFFICIENT AND COORDINATION = NOT-FEASIBLE

THEN: MODEL = TEXTVAL :LINE 2 :TAB 2 "Classical EOQ." :LINE 2 :TAB 2 "This is the oldest inventory model available in inventory literature. Because of its simplicity in computations, it has frequently been misused. It is a good idea to carefully reconsider your assumptions before adopting this model." :LINE 2 :TAB 2 "REFERENCE : [SIL-PET], pp. 174-180. Sections 5. 1-5.3." :LINE 2 :TAB 2 "COMPUTER PROGRAM : EOQ.BAS" :LINE 2

DESCRIPTION: Classical EOQ [100]

It is important to note that, along with the EOQ type general model requirement nine other requirements are present in order for the model to recommend the Classical EOQ model. These are (LEAD-TIME = Zero), NUMBER-OF-PRODUCTS (=One), SHORTAGES(=Not Permissible), REPLENISHMENT (= All at Once), UNIT-VARIABLE-COST (= Independent of Order Quantity), INFLATION (= Low or Stable), LIFE (= Sufficiently long), SPACE (= Sufficient) and COORDINATION (= Not Feasible). If any one of these is not satisfied, but the others are, then the system recommends a model which is the extension of the classical EOQ.

When the general type is EOQ, and if any two of the above nine parameters are not satisfied in Rule #5 above, then we can no longer have a model which qualifies as an EOQ extension. The second rule group "Not Classical EOQ" contains a total of eight new rules each reaching the conclusion that possible choice for an EOQ model is not classical EOQ. Although it was possible to make up a single rule with $\binom{9}{2} = 36$ conditions in its IF part, we preferred to use eight different rules for facility of debugging. Following Rule #44 is an example to the rules in this group.

<u>RULE004</u> [NOT-CLASSICAL-EOQ-RULES]

If 1) general model type is EOQ, and

2) 1) 1) lead-time is not ZERO, and

2) number of products considered is MULTIPLE, or

2) 1) lead-time is not ZERO, and

2) 1) inventory shortage is PERMISSIBLE-BACKLOGGING, or

2) inventory shortage is PERMISSIBLE-LOST-SALES, or

3) 1) lead-time is not ZERO, and

2) replenishment rate of the order <or production quantity> is GRADUALLY, or

4) 1) lead-time is not ZERO, and

2) amount paid to the supplier for each unit purchased is DEPENDENT-ON-ORDER-QUANTITY, or

5) 1) lead-time is not ZERO, and

2) future inflation expectations is HIGH, or

6) 1) lead-time is not ZERO, and

2) item's lifetime is LIMITED, or

7) 1) lead time is not ZERO, and

2) storage space availability is LIMITED, or

8) 1) lead-time is not ZERO, and

2) coordination of replenishments is POSSIBLE,

Then it is definite (100%) that possible choice for an EOQ model is Not Classical EOQ.

If: GENERAL-TYPE = EOQ AND (LEAD-TIME != ZERO AND NUMBER-OF-PRODUCTS = MULTIPLE) OR (LEAD-TIME != ZERO AND (SHORTAGES = PERMISSIBLE-BACKLOGGING OR SHORTAGES = PERMISSIBLE-LOST-SALES)) OR (LEAD-TIME != ZERO AND REPLENISHMENT = GRADUALLY) OR (LEAD-TIME != ZERO AND UNIT-VARIABLE-COST = DEPENDENT-ON-ORDER-QUANTITY) OR (LEAD-TIME != ZERO AND INFLATION = HIGH) OR (LEAD-TIME != ZERO AND LIFE = LIMITED) OR (LEAD-TIME != ZERO AND SPACE = LIMITED) OR (LEAD-TIME != ZERO AND COORDINATION = POSSIBLE)

THEN: EOQ-MODEL = "Not Classical EOQ"

Here, the symbol != means "is not".

The third rule group has a collection of rules which would be activated when the general type is EOQ and the possible choice for an EOQ model is not the classical EOQ. When these conditions are satisfied, and if any one of the above nine parameters has a value which would make the model approximate an EOQ extension, then these rules would recommend that model, but with a substantially reduced confidence factor. An example is Rule #56 below, which recommends using EOQ with quantity discounts (confidence factor 30), but cautions the user to be careful.

<u>RULE056</u> [EOQ-POSSIBILITIES-RULES]

If
1) general model type is EOQ, and
2) possible choice for an EOQ model is Not Classical EOQ, and
3) amount paid to the supplier for each unit purchased is DEPENDENT-ON-ORDER-QUANTITY,

Then there is weakly suggestive evidence (30%) that the model you should use is EOQ with quantity discounts. Caution must be exercised in using this model as not all the assumptions leading to the model are satisfied. Please check your input values using the REVIEW command. REFERENCE : [HAD-WHI], pp. 62-66.

COMPUTER PROGRAM : EOQDISC.BAS.

IF: GENERAL-TYPE = EOQ AND EOQ-MODEL = "Not Classical EOQ" AND UNIT-VARIABLE-COST = DEPENDENT-ON-ORDER-QUANTITY

THEN: MODEL = TEXTVAL :LINE 2 :TAB 2 "EOQ with quantity discounts." :LINE 1 :TAB 3 "Caution must be exercised in using this model as not all the assumptions leading to the model are satisfied. Please check your input values using the REVIEW command." :LINE 1 :TAB 3 "REFERENCE :[HAD-WHI], pp. 62-66" :LINE 1 :TAB 3 "COMPUTER PROGRAM :EOQDISC.BAS" CF 30

DESCRIPTION: EOQ with quantity discounts [Unit-variable-cost 30]

The fourth rule group contains rules which are related to deterministic non-EOQ type models, such as the time varying, periodic review lot-size model (Wagner and Whitin 1958). Rule groups 5 to 10 contain rules recommending stochastic models only. All of these rules start by determining the general model type and then ask other questions trying to reach the goal MODEL and make a recommendation.

It should perhaps be clear by this time that our expert system classifies all of the inventory models into four mutually exclusive (and collectively exhaustive) groups. This is achieved by identifying the value of the parameter GENERAL-TYPE as one of

1. EOQ,
2. Deterministic Non-EOQ,
3. Newsboy, and
4. Stochastic Non-Newsboy.

At the outset, to assign a value to GENERAL-TYPE, the user is asked a question on DEMAND-PROCESS and DEMAND-LEVEL or LIFE depending on the answer given to the first question. Answering these two prompts, immediately places the system in one of the four groups, thereby effectively eliminating all the other irrelevant rules pertaining to the three groups. For example, if GENERAL-TYPE is found to be NEWSBOY, the rules in EOQ and Extensions group, etc. are not considered at all since they would not be relevant to a stochastic demand model. This process is described in Figure 2.

\- - - - - - - - - - - -

Insert Figure 2 here

\- - - - - - - - - - - -

Now, the systems would attempt to prove whether the classical Newsboy or any one of its extensions is the applicable model, using e.g. rules such as Rule #68.

RULE068 [NEWSBOY-RULES]

If 1) general model type is NEWSBOY, and

2) number of products considered is MULTIPLE, and

3) substitute products exist, and

4) fixed order <or set-up> cost is ZERO,

Then it is definite (100%) that the model you should use is Classical Newsboy with Substitutable Products REFERENCES : [PAR-GOY], Opsearch, Vol. 21, pp. 1-15, (1984) for the two item case with exact solution: [PAR], Opsearch, Vol. 23, pp. 250-257, (1986) for the multi-item case with an heuristic. COMPUTER PROGRAM : NEWSBSUB.BAS

IF: GENERAL-TYPE = NEWSBOY AND NUMBER-OF-PRODUCTS = MULTIPLE AND SUBSTITUTE AND FIXED-COST = ZERO

THEN: MODEL = TEXTVAL :LINE 2 :TAB 2 "Classical Newsboy with Substitutable Products" :LINE 1 :TAB 3 "REFERENCES : [PAR-GOY], Opsearch, Vol. 21, pp. 1-15, (1984) for the two item case with exact solution: [PAR], Opsearch, Vol. 23, pp. 250-257, (1986) for the multi-item case with an heuristic" :LINE 2 :TAB 2 "COMPUTER PROGRAM : NEWSBSUB.BAS"

DESCRIPTION: Classical Newsboy with Substitutable Products [100]

Although it seems obvious, it is very important to emphasize that a structure such as above would cut down the search space of the problem and reduce the time required to find the correct model. For any expert system, the first few questions the user answers, should eliminate many irrelevant outcomes and make the search for the goal more efficient.

Clearly, there are some cases where the user's responses are so unusual that, a relevant model does not exist for the given situation. An example consultation for such a case is provided in Figure 4, where the system concludes that no (known) models are available. A result such as this may be useful to researchers working in inventory management by providing them with possibly new research topics not analyzed before.

- - - - - - - - - - - - -

Insert Figure 3 Here

- - - - - - - - - - - - -

Finally, the following figure describes the results of a particular consultation session where the system recommends the Periodic lot size model. The reference [SIL-PET] is (Silver and Peterson 1985). In the second part, the user can REVIEW his previous choices and re-run the system.

- - - - - - - - - - - - -

Insert Figure 4 Here

- - - - - - - - - - - - -

As we mentioned before, the system can identify approximately 30 production-inventory models found in the current literature. Most of the models in every major production-inventory textbook, including Silver and Peterson (1985), Hax and Candea (1984), Love (1979), Johnson and Montgomery (1974) and Naddor (1966) are included. We should note that the current system can be easily extended and refined as new models become available. This is done by adding new rules to the knowledge base and by possibly changing some of the older rules which interact with the new ones. It is obvious that as long as new inventory models are published and become available, this expert System will continue growing and will be richer in content and expertise.

4. Lessons Learned

As the use of expert systems in management science applications is relatively new, it would be worthwhile to summarize our experiences gained during the development of EXPIM. It is hoped that other researchers who are working on similar projects, e.g. queueing expert systems would find these hints useful.

We recommend that the developer use an expert system shell instead of an AI language such as LISP or Prolog. The selection of the shell should be based on its capability to do backward chaining with a forward chaining option. Using backward chaining, many irrelevant choices can be eliminated quickly in the early stages. It is advisable to use a shell which can interface with outside programs and files. For example, many expert systems may have to do heavy mathematical computations during or at the conclusion of a consultation (e.g. running a program to solve the EOQ model in EXPIM). Similarly, they may need to read data from outside files or from databases instead of asking the user to supply these data values. As management science application would invariably involve some computations, these features become important. Many shells have the capability to answer WHY and HOW questions, and do sensitivity analyses. It is crucial that the developer choose one with these capabilities.

As for the development of the knowledge base, the developer should group the models into mutually exclusive and collectively exhaustive groups so that any available model can be placed in exactly one of the groups. For example, in the case of queueing expert system, one may initially create two groups, i) single server models and ii) multi-server models.

In each group a rule should be written for the best known model e.g. classical EOQ in deterministic EOQ and M/M/1 in single server queues. Then, extensions of this fundamental model should be considered by writing slightly different variations of the rule for the fundamental model. Finally, new rules should be created which would choose these known models with lower confidence factors because of some of the unsatisfied assumptions, e.g. Rule #56 discussed in Section 3.

It is also important to have some idea who the users might be before the development starts. It would be useful to have rules which could reach conclusions by asking the user easy to understand questions, e.g. Rule #4 discussed in Section 3.

When the expert system reaches a conclusion by recommending a particular model, it should also give the user the reference (book or article) where the model can be found. This way, the user can analyze the model in more detail and perhaps see a few numerical examples relevant to the model.

5. Summary and Conclusions

In this paper we have discussed in detail, the conceptualization and development of an expert system (EXPIM) which can identify up to 30 production and inventory models. In the absence of a human expert, the manager who is interested in using these models can access the expert system and after responding to a few questions, he can get a recommendation from the system. EXPIM is written in Texas Instruments' Personal Consultant Easy expert system shell and runs on IBM-PC micro computers with at least 512K memory.

As mentioned in Section 4, there is a potential for creating other expert systems in management science for classifying different collection of models. For example, queueing theory, with its plethora of models is a prime candidate for this research. Location theory is another possibility. We hope that our points summarized in section 4 will aid other researchers who want to enter this new and interesting research field which combines management science and artificial intelligence.

We note that to make these expert systems more "intelligent", they should have the ability to reach some intermediate conclusions on their own. For example, in EXPIM, instead of asking the user about the DEMAND-LEVEL, the system should ideally be able to determine it on its own. This would require the system to access external data files, do some computations as discussed in Silver and Peterson (1985, p.238) to obtain a measure of the variability of demand pattern and then determine the value of the parameter DEMAND-LEVEL. Current version of EXPIM does not have this capacibility, but it would be possible to include it in a later version.

In the early days of expert system development, artificial intelligence researchers used to recommend that a knowledge engineer should not be his own expert (Nii 1983). This is still true to a certain extent but these days the availability of easy to use shells have made it easier for the domain experts to become proficient in the use of these tools and have assumed the role of a knowledge engineer. Perhaps the convenience and low price of microcomputers which can run AI software have played an important role in this development.

To conclude, it is our belief that expert systems will be playing an important role in management science especially in classification and choice of models as we discussed in this paper and also in automatic model building (Binbasioglu and Jarke 1986).

ACKNOWLEDGEMENT

Financial support from the Natural Sciences and Engineering Research Council of Canada under Grant No. A5872 is gratefully acknowledged.

List of Figures

Appendix

Backward and Forward Chaining in PC Easy

(From PC Easy Reference Guide, pp. 3-19/3-30)

Backward Chaining

Backward chaining is the primary means by which PC Easy arrives at a solution to a problem--the knowledge base goal. The driving force behind a PC Easy consultation is the attempt to set a value for each of the one or more parameters listed in the knowledge base GOALS property.

PC Easy begins the attempt to set the value of a goal parameter by looking for a rule whose THEN statement assigns a value to the parameter. When it finds a rule, it tests it, or determines whether the conditions expressed in the rule's IF statement are true.

To determine whether the conditions expressed in the rule's IF statement are true, PC Easy may need to find the value of one or more parameters included in the IF statement. This search may lead to another rule that sets the value of one or more of these parameters.

If the conditions in the IF statement are true, the rule passes. If the conditions in the IF statement are not true, the rule fails.

If a rule passes, PC Easy fires the rule--carries out the action specified in its THEN statement. PC Easy fires a rule only once during a consultation.

Example

In this example, PC Easy finds the value of the goal parameter GIFT by backward chaining. GIFT-TYPE and RECEIVER are parameters in the knowledge base.

1. The consultation begins.
2. PC Easy find a rule that assigns a value to GIFT:

IF statement: GIFT-TYPE = PERSONAL
THEN statement: GIFT = FLOWERS

To determine whether to apply the action of this rule and set the value of GIFT TO FLOWERS, PC Easy must search for the value of GIFT-TYPE.

3. PC Easy finds a rule that assigns a value to GIFT-TYPE:

IF statement: RECEIVER = SPOUSE
THEN statement: GIFT-TYPE = PERSONAL

To test this rule, PC Easy must find the value of RECEIVER.

4. Because no rule assigns a value to RECEIVER and RECEIVER has a PROMPT property, PC Easy prompts the client:

Is the receiver of the gift your spouse?

5. If the client answers yes, PC Easy assigns RECEIVER the value SPOUSE.

Because the condition stated in the IF statement of the rule is true, PC Easy carries out the action of the THEN statement and assigns GIFT-TYPE the value PERSONAL.

6. Because GIFT-TYPE has the value PERSONAL, the condition stated in the IF statement of the original rule is true, and PC Easy determines that the value of GIFT is FLOWERS.

Forward Chaining

In forward chaining, PC Easy tries an antecedent rule when it has assigned a value to one of the parameters in the antecedent rule's IF statement. In EXPIM an antecedent rule is used to invoke a rule whose THEN statement does not conclude values for any parameter. For example, when the model type is determined to be STOCHASTIC-NON-NEWSBOY and if the user selects NON-PERMISSIBLE for inventory shortage parameter, then he gets a message indicating the unavailability of any models for the current situation. The following antecedent Rule #74 describes an example to forward chaining in PC Easy.

RULE074 [PERIODIC-REVIEW-RULES/antecedent]

If 1) general model type is STOCHASTIC-NON-NEWSBOY, and

2) inventory shortage is NON-PERMISSIBLE,

Then it is definite (100%) that the model you should use is There is no available model. Please note that you chose PERMISSIBLE when the computer asked you about the SHORTAGES. Prior to that you had specified that DEMAND was STOCHASTIC, and LIFETIME was SUFFICIENTLY-LONG. Under these conditions it is impossible to require that inventory should always be positive. Please REVIEW your responses and change them accordingly.

IF: GENERAL-TYPE = STOCHASTIC-NON-NEWSBOY AND SHORTAGES = NOT-PERMISSIBLE

THEN: MODEL = TEXTVAL :LINE 1 :TAB 3 "There is no available model. Please note that you chose PERMISSIBLE when the computer asked you about the SHORTAGES. Prior to that you had specified that DEMAND was STOCHASTIC, and LIFETIME was SUFFICIENTLY-LONG. Under these conditions it is impossible to require that inventory should always be positive. Please REVIEW your responses and change them accordingly".

DESCRIPTION: Warning the user if positive inventory is required in stochastic case

ANTECEDENT: YES

Bibliography

AGGARWAL, S.C., "A Review of Current Inventory Theory and Its Applications", Int. J. Prod. Res., 12, #4, (1974), 443-482.

ASSAD. A.A. and B.L. GOLDEN, "Expert Systems, Microcomputers and Operations Research", Comput. & Oper. Res., 13, #2/3, (1986), 301-321.

AZARM, S. and M. PECHT, "An Approach for Building a Rule-Based System for Design Optimization", Working Paper No. TR-85-51, Dept. of Mechanical Engineering, The U. of Maryland, 1985.

BARANCSI, E., G. BANKI, R. BORLOI, A. CHIKAN, P. KELLE, T. KULCSAR and GY. MESZENA, "Analysis of a System of Inventory Models", in Proc. First Int. Symp. on Inventories, Budapest, Hungary, pp. 297-308, (1980).

BARANCSI, E., _____, _____, _____, _____, _____, _____, "A Report of Research on Inventory Models," Eng. Costs and Prod. Econ., 7, (1983), 127-136.

BARR, A. and E.A. PEIGENBAUM, The Handbook of Artificial Intelligence, Vol. 1, William Kaufmann, Los Altos, Calif., 1981.

BINBASIOGLU, M., and M. JARKE, "Domain Specific DSS Tools for Knowledge Based Model Building", Dec. Sup. Syst., 2, (1986), 213-223.

BISWAS, G., R. ABRAMCZYK and M. OLIFF, "OASES: An Expert System for Operations Analysis - The System for Cause Analysis", IEEE Trans. on Sys. Man and Cyb., SMC-17, #2, (1987), 133-145.

BUCHANAN, B.G. and R.O. DUDA, "Principles of Rule-Based Expert Systems", in M. YOVITS, (Ed.), Advances in Computers, Vol. 22, (1983), 163-216.

CHIKAN, A., P. KELLE and GY. MESZENA, "A Comparison of Various Classification Systems of Inventory Models", in Proc. Second Int. Symp. on Inv., Budapest, Hungary, 661-667, 1982.

DAVIS, R. and J. KING, "An Overview of Production Systems", in E. ELCOCK and D. MICHIE (Eds.), Machine Intelligence, Ellis Horwood, Chichester, England, 300-332, 1977.

DOUKIDIS, G.I. and R.J. PAUL, "Research Into Expert Systems to Aid Simulation Model Formulation", J. Opl. Res. Soc., 36, #4, (1985), 319-325.

DUDA, R.O., J. GASCHNIG and P.E. HART, "Model Design in the PROSPECTOR consultant system for Mineral Exploration", in D. MICHIE (Ed.), Expert Systems in the Microelectronic Age, Edinburgh Univ., Edinburgh, 153-167, (1979).

EILON, S. and W. LAMPKIN, Inventory Control Abstracts. (1953-1965), Oliver and Boyd, Edinburgh, 1968.

FEIGENBAUM, E.A. and P. MCCORDUCK, The Fifth Generation, Addison-Wesley, Reading, Mass. 1983.

HADLEY, G. and T.M. WHITIN, Analysis of Inventory Systems, Prentice-Hall, Englewood Cliffs, N.J. 1963.

HAHN, G.J., "More Intelligent Statistical Software and Statistical Expert Systems: Future Directions", Amer. Statistician, 39, #1, (1985), 1-8.

HAX, A.C. and D. CANDEA, Production and Inventory Management, Prentice-Hall, Englewood Cliffs, NJ, 1984.

HAYES-ROTH, F., D.A. WATERMAN and D.B. LENAT, (Eds.), Building Expert Systems, Addison-Wesley, Reading, Mass., 1983.

HOLLIER, R.H. and P. VRAT, "A Proposal for Classification of Inventory Systems", Omega, 6, #3, (1978), 277-279.

IGLEHART, D.L., "Recent Results in Inventory Theory", J. of Ind. Eng., 18, #1, (1967), 48-51.

JOHNSON, L.A. and D.C. MONTGOMERY, Operations Research in Production Planning, Scheduling and Inventory Control, John Wiley, New York, 1974.

KASTNER, J.K. and S.J. HONG, "A Review of Expert Systems", Euro. Jnl. of Opnl. Res., 18, #3 (1984), 285-292.

KENDALL, D.G. "Some Problems in the Theory of Queues", J.R. Stat. Soc., B-13, #2 (1951), 151-185.

KHOSHNEVIS, B. and A. CHEN, "An Expert Simulation Model Builder", Working Paper No. 84-04, Dept of ISE, U. of Southern California, 1986.

LINDSAY, R., B.G. BUCHANAN, E.A. FEIGENBAUM and J. LEDERBURG, DENDRAL, MCGRAW-HILL, New York, 1980.

LOVE, S.F., Inventory Control, McGraw-Hill, New York, 1979.

MENIPAZ, E., "A Taxonomy and Shibboleth of Inventory Models", in Proc. Second Int. Symp. on Inventories, Budapest, Hungary, 745-752, 1982.

NADDOR, E., Inventory Systems, John Wiley, New York, 1966.

NAHMIAS, S., "Inventory Models", in J. BELZER, A.G. HOLZMAN and A. KENT (Eds.), The Encycl. of Comp. Sci. and Techn., Vol. 9, Marcel Dekker, 1978.

NII, H.P., "Heuristics for Knowledge Engineering", reported in (Feigenbaum and McCorduck, p. 83, 1983).

POST, E., "Formal Reductions of the General Combinational Problem", Am. Jnl. of Math., 65, (1943), 197-268.

PRABHU, N.V., Queues and Inventories, John Wiley, New York, 1965.

RAVINDRAN, A., D.T. PHILLIPS and J.J. SOLBERG, Operations Research; Principles and Practice, 2nd Ed., John Wiley, New York, 1987.

REGGIA, J.A. and S.B. AHUJA, "Selecting an Approach to Knowledge Processing", Comp. and Oper. Res., 13, #2/3, (1986), 329-331.

SCHWARZ, L.B. (Ed.), Multi-level Production/Inventory Control Systems: Theory and Practice, North-Holland, Amsterdam, 1981.

SHORTLIFFE, E.H., Computer Based Medical Consultations: Mycin, Elsevier, New York, 1976.

SILVER, E.A. and R. PETERSON, Decision Systems for Inventory Management and Production Planning, 2nd Ed., John Wiley, New York, 1985.

SILVER, E.A., "Operations Research in Inventory Management: A Review and Critique", Oper. Res., 29, #4, (1981), 628-645.

STEFIK, M., J. AIKINS, R. BALZER, J. BENOIT, L. BIRNBAUM, F. HAYES-ROTH and E. SACERDOTI, "The Organizations of Expert Systems, A Tutorial", Artificial Intelligence, 18, (1982), 135-173.

VAN BREDA, M.F., "Expert Systems in University Admission Decisions", Presented at the TIMS/ORSA Meeting, Los Angeles, (April 1986).

VEINOTT, A.F., "The Status of Mathematical Inventory Theory", Mgt. Sci., 12, #11, (1966), 745-777.

WAGNER, H. and T.M. WHITIN, "Dynamic Version of the Economic Lot Size Model", Mgt. Sci., 5, #1, (1958), 89-96.

WAGNER, H., "Research Portfolio for Inventory Management and Production Planning Systems", Oper. Res., 28, (1980), 445-475.

WATERMAN, D.A., A Guide to Expert Systems, Addison Wesley, Reading, Mass. 1986.

WEISS, S.M. and C.A. KULIKOWSKI, A Practical Guide to Designing Expert Systems, Rowman and Allanheld, Totowa, N.J. 1984.

WISARD Software Co., Wisard Forecaster for IBM-PC, Green Bay, WI, 1986.

EXPIM : Expert Production-Inventory Modeller

```
What is the level of the future forecasted demand?

  [CONSTANT-OVER-TIME]   Help:
   VARIABLE-OVER-TIME
                            If the forecasted future level of demand is
                          basically constant (uniform over time) , choose
                          CONSTANT-OVER-TIME.
                            If the forecast indicates lumpy demand where
                          demand varies from time to time, choose
                          VARIABLE-OVER-TIME.
                            Refer to [SIL-PET], pp. 237-239, Section 6.6.4.
                          for an explanation of a numerical technique which
                          can distinguish between these two cases.

                           ** End - RETURN/ENTER to continue

  1. Use the arrow keys or first letter of item to position the cursor.
  2. Press RETURN/ENTER to continue.
```

Figure 1

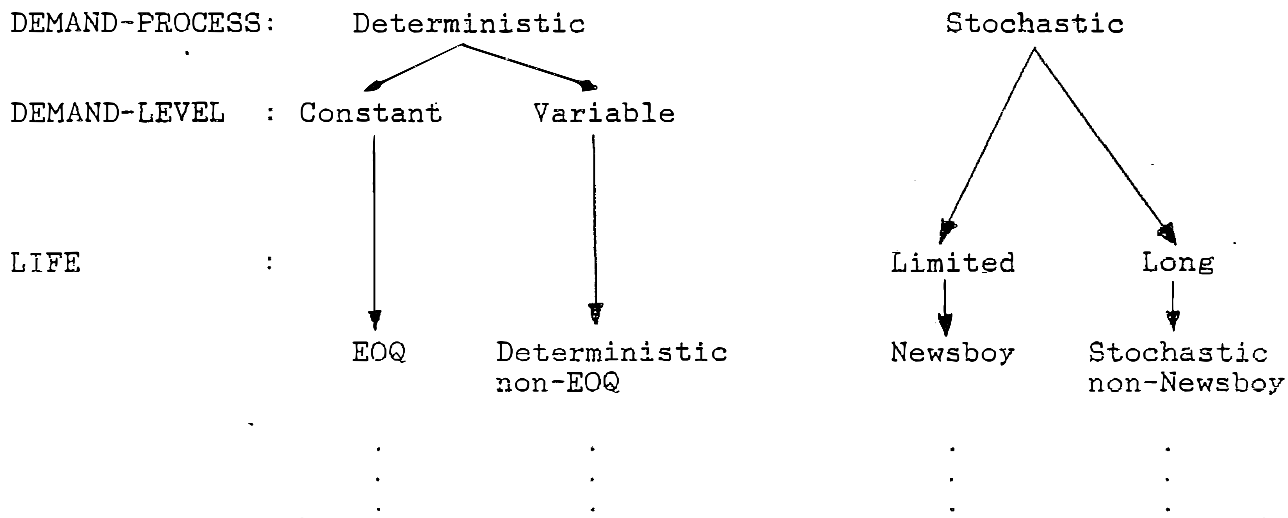
DEMAND-PROCESS:
Deterministic
Stochastic
DEMAND-LEVEL :
Constant
Variable
LIFE :
Limited
Long
EOQ
Deterministic non-EOQ
Newsboy
Stochastic non-Newsboy

Figure 2

```
(" Consultation record for:  EXPIM : Expert Production-Inventory Modeller"
"the forecast for the future demand       :: STOCHASTIC"
"item's lifetime                          :: SUFFICIENTLY-LONG"
"periodic monitoring of the inventory... :: YES"
"lead-time                                :: STOCHASTIC"
"fixed order <or set-up> cost             :: POSITIVE"
"inventory shortage                       :: PERMISSIBLE-LOST-SALES"
"number of products considered            :: MULTIPLE"
"substitute products exist                :: YES"
)
```

CONCLUSIONS:

I was unable to make any conclusions regarding the model you should use

List of books used in this knowledge base is as follows:

Not available since there are no known (or easily implementable) models for solving the inventory problem described by the user in this consultation session.

Potential research topics may be obtained from this consultation's results by referring to the combination of inputs provided by the user. Please use the REVIEW feature to recall your inputs, i.e. the parameters you supplied during the consultation

List of articles used in this knowledge base is as follows:

Again, not available for the reasons given above

Figure 3

EXPIM : Expert Production-Inventory Modeller

Conclusions:

The model you should use is as follows:

Periodic review lot-size
This model could be solved using the Wagner-Whitin exact algorithm or by using Silver-Meal heuristic or by using a few other heuristics reported in the literature.
REFERENCE : [SIL-PET], pp. 227-243, Sections 6.5-6.7

List of books referenced in this knowledge base is as follows:

BOOKS:
[BAN-FAB] : J. Banks and W.J. Fabrycky, 'Procurement and Inventory Systems Analysis', Prentice-Hall, 1987.
[BER] : D.P. Bertsekas, 'Dynamic Programming and Stochastic Control', Academic Press, 1976.
[ELS-BOU] : E.A. Elsayed and T.O. Boucher, 'Analysis and Control of Production Systems', Prentice-Hall, 1985.
[HAD-WHI] : G. Hadley and T.M. Whitin, 'Analysis of Inventory Systems', Prentice-Hall, 1963.
** More - RETURN/ENTER to continue

EXPIM : Expert Production-Inventory Modeller

Conclusions:
Management and Production Planning', 2nd Edition, John Wiley, 1985.

Li Review:

J Yes

[⊢] •	the forecast for the future demand	::	DETERMINISTIC
Co ⊢ •	the level of the future demand	::	VARIABLE-OVE...
19 ⊢ •	periodic monitoring of the inventory...	::	YES
⊢ •	lead-time	::	ZERO
Su ⊢ •	number of products considered	::	ONE
1- ⊢ •	inventory shortage	::	NOT-PERMISSIBLE
⊢ •	replenishment rate of the order <or ...	::	ALL-AT-ONCE
wi ⊢ •	supplier's quantity discount	::	NOT-AVAILABLE
25 ↓ ⊢ •	future inflation expectations	::	LOW

1. Use arrow keys or first letter of item to position cursor.
I 2. Select all applicable responses.
sh 3. After making selections, press RETURN/ENTER to continue.

Figure 4

Faculty of Business

McMaster University

WORKING PAPERS - RECENT RELEASES

257. Joseph B. Rose, "Statutory Expedited Grievance Arbitration: The Case of Ontario", July, 1986.

258. John W. Medcof, "The Effects of Event Valence and Degree of Attribution Upon Action Recommendations", September 1986.

259. Paul D. Dowling and Robert F. Love, "One-Dimensional Floor Layout Solutions Using Squared Distances", September, 1986.

260. Harish C. Jain, "Affirmative Action/Employment Equity in Canada: Findings of a Survey", September, 1986.

261. Min Basadur, "Catalyzing Interfunctional Efforts to Find and Creatively Solve Important Business Problems", September, 1986.

262. Roy J. Adams and Isik Zeytinoglu, "Labour-Management Dispute Resolution in Canadian Heavy Industry: The Hilton Works Case", September, 1986.

263. Rick D. Hackett, "New Directions in The Study of Employee Absenteeism: A Research Example", October 1986.

264. Ali-Reza Montazemi, "An Analysis of Information Technology Assessment and Adoption in Small Business Environments", October, 1986.

265. Isik U. Zeytinoglu, "Part-time Workers: Unionization and Collective Bargaining in Canada", November, 1986.

266. Norman P. Archer and M.W. Luke Chan, "The Architecture of Chinese-English Microcomputer Systems", December, 1986.

267. Norman P. Archer, "Structuring Problems for Analysis with Electronic Spreadsheets", January, 1987.

268. Norman P. Archer, "A Life Cycle Approach to the Implementation of Integrated Information Technology", January, 1987.

269. Joseph B. Rose, "An Overview of the Canadian and Australian Industrial Relations Systems", February, 1987.

270. Joseph B. Rose, "A North American Perspective on Multi-Employer Cohesion in Australian Construction", February, 1987.

271. Christopher K. Bart, "Budgeting Gamesmanship: Survival Tactics in a Hostile Environment", February, 1987.

272. Thomas E. Muller, "Modeling the Tourist's Urban Experience: A New Approach to International Tourism Product Development Based On Urban Quality of Life, February, 1987.

273. A. William Richardson, "Cost-Volume-Profit Analysis and the Value of Information: An Evaluation for the Normal and Lognormal Distributions", February, 1987.

274. Roy J. Adams, "Industrial-Relations Systems: An International Comparison", March, 1987.

275. John W. Medcof, "An Integration of Some Attribution Theories", March, 1987.

276. Y.L. Chan and A.R. Montazemi, "An Experimental Study of the Information Choices of Security Analysts", March, 1987.

277. Isik Zeytinoglu, "Part-time Workers and Collective Bargaining in Internal Labour Markets", May, 1987.

278. Abraham Mehrez, "A Multi-Objective Approach to a Mixed Integer Non-Linear Covering Problem", July, 1987.

279. Abraham Mehrez, Yufei Yuan, Amiram Gafni, "Stable Solutions Vs. Multiplicative Utility Solutions For the Assignment Problem", July, 1987.

280. Harish C. Jain, Rick D. Hackett, "Employment Equity Programs In Canada: Public Policy And A Survey", July, 1987.

281. Norm P. Archer, "MBA Information Systems Curriculum Needs: A Business Survey", August, 1987.

282. Chris K. Bart, "New Venture Units: Organizing for New Products", September, 1987.