Int. J. Engng Ed. Vol. 8, No. 5, pp. 355–366, 1992
Printed in Great Britain.

0742–0269/91 $3.00+0.00

Application of GPSS/FON in Teaching Simulation*

A. M. ZIKIC
Department of Electrical and Electronic Engineering, University of Paisley, Paisley PA1 2BE, Scotland
B. Lj. RADENKOVIC
Department of Organizational Sciences, University of Belgrade, YU-11000 Belgrade, FON, Jove Ilica 154, Yugoslavia

The GPSS/FON simulation language was developed as a joint project of the Department of Organizational Sciences, University of Belgrade and the Department of Electrial and Electronic Engineering, Paisley College, as a portable software package written in Pascal for various computer systems, ranging from PCs to mainframes. The main purpose of this software is to enable a cheap and easy learning and practising environment in the area of discrete event system simulation, particularly suited for students of engineering. In this paper we present the language definition, its implementation and the user interface. Our experience in teaching simulation will be illustrated by an example of a student's work.

SUMMARY OF EDUCATIONAL ASPECTS OF THE PAPER

1. The paper discusses material for a course in modelling and simulation of discrete event systems.
2. Students of mechanical engineering, management, electrical and electronic engineering and computer science. We are planning to introduce the course to students of industrial engineering as well.
3. Electric and electronic engineers are taught at the postgraduate level, all others at the graduate level.
4. Mode of presentation is through lectures, tutorials and laboratory course work, the latter consisting of four compulsory lab/homework exercises.
5. Material is presented in regular compulsory courses.
6. The course is taught for sixteen weeks with two hours of lectures and two hours of tutorials each week. Lab homework time is not strictly timetabled and varies from student to student. However, the lecturer is available twice a week for consultancy purposes.
7. The time needed to complete the homework exercise varies from student to student, and averages around 40 hours. Students that successfully complete the full course need little revision before passing their exam.
8. The novelty of our teaching method lies in the fact that students can use an interactive, integrated and user-friendly environment that enables easy development of models, and quick and effective analysis of results through build-in ANALYZE features discussed in the paper. Not being dependent on proprietor restrictions on software we include a copy of our software in every set of lecture-tutorial sheets, free of charge.

   Our experience shows that students' study is more relaxed if they can use their home computers in addition to the college ones and that they gain better results spending fewer hours studying than previously. For those students that have old 8-bit processor systems, we developed a version of GPSS/FON that is in p-code rather than the fully compiled form. It enables the development (but not analysis) of all the exercise models. Of course, all the college/faculty computing facilities (mainframe, mini- and microcomputers) support the package for the duration of the course.
9. Reference [1] is an indispensable source in teaching GPS and covers the complete version of the language. As our format definition differs from the old IBM standard, and as our package consists of an integrated development/simulation/analysis environment, the student notes and tutoral sheets emphasise these features, covering a minimal theoretical background.

   The notes have a comprehensive coverage of the simulation concepts and language features with the complete syntax and semantics and a short form user-manual for the integrated environment is distributed throughout the text. Lengthy descriptions of the language features, options and definitions of non-supported instructions are excluded and the above

* Paper accepted 1 June 1992.

reference is offered for detail where appropriate. In addition, all terminal and micro-labs have complete GPSS/FON manuals.

INTRODUCTION

GPSS has always been looked upon as a process-oriented language for simulating discrete events. Since its initial release by IBM in 1961, GPSS has been available in its original form on mainframes such as IBM 360/370, and DEC, followed by the development of 'dialects' suitable for minicomputer implementation. At present, our development allows the use of GPSS on virtually any computer system, including some eight-bit processor machines, that support a Pascal compiler.

At many sites, the installation of GPSS has proved to be prohibitive as it comes to approximately US$1000 per user, irrespective of the computer system type. Universities and colleges in eastern Europe and Third World countries quite simply cannot afford to purchase a sufficient number of legal copies needed for teaching a large number of students. On the other hand, it is doubtful if western students can afford to acquire legal copies for home practising. In order to overcome this problem, we decided to write our own version of GPSS that can run on any machine and that is affordable to both institutions and individuals anywhere in the world.

In 1984, we started developing our version of GPSS named GPSS/FON, and finished it in 1987. The Pascal implementation, based on the definitions given in [1], provided 90 per cent of the most frequently used GPSS statements; more than needed in the process of education and other more sophisticated applications [2]. After over four years of practical experience at both sites, we concluded that this implementation was suitable enough for educational training in simulation of discrete event systems. For these reasons we made this program available upon request through e-mail at earn_011@yubgef51.bitnet, for educational purposes only.

THEORETICAL BACKGROUND

In practice, a real system or process, especially control or operation research applications, is described as a collection of changes of the describing variables; these variables are defined as a set of all input and state variables. The event is defined as a change in either the system or in the environment. The systems that fall into this category are known as discrete event systems.

A discrete event system can be described as an ordered septuplet [3]:

$$\mathbf{s} = \langle \boldsymbol{T}, \boldsymbol{X}, \boldsymbol{\Omega}, \boldsymbol{Q}, \boldsymbol{Y}, \boldsymbol{\delta}, \boldsymbol{\lambda} \rangle$$

where:

$\mathbf{T}:[t_0, \infty) \subset \mathbb{R}$ is the time base; $\mathbf{X}:\mathbf{X}_M \cup \{\varnothing\}$, where $\mathbf{X}_M$ is a set of external events; $\boldsymbol{\Omega}$ is a set of the input discrete events segments such that

$$\boldsymbol{\Omega}:\{\omega \mid \omega:\langle t_0, t_1\rangle \to \mathbf{X}\}$$

$\mathbf{Q}$ is a set of actual states i of all variables $\xi^{(k)}$ and defined as

$$\mathbf{Q}:\sim(s, e) \mid s = \{\xi_i^{(a)}, \xi_j^{(b)}, \ldots\} \in \mathbf{S}, 0 \leqslant e \leqslant \tau(s)\}$$
$$\tau:\mathbf{S} \to \mathbb{R}^{+}_{0,\infty}$$

where $\mathbf{S}$ is a set of sequential states and e is a time interval spent in a state s; $\boldsymbol{\delta}:\mathbf{Q} \times \mathbf{X} \to \mathbf{Q}$ is a quasi-transfer function defined as

$$\boldsymbol{\delta}(s, e, \varnothing) = s \;\forall\; e < \tau(s)$$
$$\boldsymbol{\delta}(s, \tau(s), \varnothing) = \boldsymbol{\delta}^{\varnothing}$$
$$\boldsymbol{\delta}(s, e, x) = \boldsymbol{\delta}^{ex}(s, e, x)$$

where

$$\boldsymbol{\delta}^{\varnothing}:\xi_i^{(k)} \to \xi_j^{(k)}\; x = \{\varnothing\}$$
$$\boldsymbol{\delta}^{ex}:s_i \to s_j\; x \in \mathbf{X}_M$$

and $\boldsymbol{\lambda}:\mathbf{Q} \to \mathbf{Y}$ is the output function.

In many cases functions $\boldsymbol{\delta}$ and $\boldsymbol{\lambda}$ are stochastic.

The models of this type are useful in describing social, economic, technical, and production systems. Simulation of such types of systems can be obtained by using different simulation strategies [4]. In practice, however, only three types of strategies are used, namely

- event scheduling
- scanning of activities
- process interactions

The process interactions strategy, being the most complex, incorporates all the main features of both the event scheduling, and the activity scanning methods, and for that reason is adopted in our implementation [1].

DEFINITION OF THE GPSS/FON LANGUAGE

Originally, all GPSS instructions were of the fixed formal structure, and only five characters per label were allowed. Our dialect of GPSS implements a free format structure for all statements and a maximum of twelve characters for variable names and labels, the first three of which must be of letter type; the latest version is case insensitive. A short description of the GPSS/FON syntax is given here in EBNF in Figure 1. A detailed description of GPSS/FON syntax is given in [5].

SIMULATION

The simulation process in GPSS/FON V1.5 consists of the following four phases:

(1) Editing a GPSS source code program;
(2) Compiling the model with GPSS assembler into an internal object code;

```
<GPSS Program>        ::= {<Program stm.>}

<Program stm.>        ::= <Statement>|[*<Comment>]

<Statement>           ::=[<Label> b] <Statement code>  b[<Comment>]

<Label>               ::= <Integer>|<Letter><Letter><Letter>[<String>]

<Statement code>      ::= <Operation code> b [<Postfix>] [<Operands> b]
                          [<Extension> b]

<b>                   ::= blank

<Operation code>      ::= ADVANCE|ASSIGN|CLEAR|
                          DEPART|END|ENTER|FUNCTION|GATE|
                          GENERATE|INITIAL|LEAVE|LINK|
                          LOGIC|MARK|QUEUE|RELEASE|RESET|
                          SAVEVALUE|SEIZE|SIMULATE|START|
                          STORAGE|TABLE|TABULATE|TERMINATE|
                          TEST|TRANSFER|UNLINK|VARIABLE

<Postfix>             ::= BOTH|E|G|GE|NE|L|LR|LS|SE|SF|SNE|SNF|NU|U

<Operands>            ::= <Operand>{,<Operand>}

<Operand>             ::= <Num.Operand>|<SNAOperand>| <Address Operand>

<Num. Operand>        ::= <Integer>

<SNA Operand>         ::= C1|CA|CC|CH|CM|CT|F|FC|
                          FN|FR|FT|M1|MP|N|P|PR|Q|QA|QC|QM|QT|QX|
                          R|RN|S|SA|SC|SM|SR|TB|TC|TD|V|W|X

<Address Operand>     ::= <Integer>|$<Label>| *<Integer>


<Extension>           ::= <Function ext.>|<Variable ext.>

<Function ext.>       ::= {/<X>,<Y>}

<X>;<Y>               ::= <Real Number>

<Variable Extension>  ::= <Arithmetic expression>
```

Fig. 1. Short description of GPSS/FON syntax in EBNF.

(3) Executing the simulated process with GPSS processor;
(4) Analysing the simulation results.

The user edits a GPSS source program of his/her choice and performs the simulation by running the following two programs;

GPSSASM ⟨object_file⟩[,⟨list_file⟩] = ⟨source_file⟩
GPSS ⟨object_file⟩

where the filename extensions are optional.

The assembly stage, GPSSASM, performs the syntax analysis and translation into the internal object code. After a successful completion, two files in addition to the *.obj are produced; the first one, *.obl, contains a user readable form of the *.obj file and the second gives a standard assembly list file, *.lst. The results of the simulation obtained by running the GPSS processor, are stored in a file, *.res and are formatted in accordance with the definition given in [1].

The actual simulation is performed by the GPSS processor in three stages:

(1) Loading of the model;
(2) Execution of the control statements;
(3) Generation of results.

The logic of the GPSS/FON processor is shown in Fig. 2.

INTEGRATED ENVIRONMENT

The latest version of GPSS/FON is organized into an integrated environment. Instead of using the system editor and running the assembler and processor programs separately, a user runs only the GP_INT program that provides an easy development and simulation environment. By using the

```
PROGRAM Gpss;
  Global variables declarations;
  Procedures declarations;
  BEGIN
    Initialization of Simulator;
    REPEAT
      Model reading phase;
      Creating first transactions;
      WHILE Termination Counter > 0 DO
        BEGIN
          Clock Update Phase;
          Scanning phase
        END
      Printing the results;
    UNTIL 'End' statement is reached
  END.
```

Fig. 2. Processing logic of the GPSS/FON processor.

mouse or arrow keys, the user selects one of the options from the main menu

FON-EDIT
ASSEMBLE
SIMULATE
ANALYZE

The editor option supports all standard screen editing features; the assembly and simulation options are performed by the unchanged GPSSASM and GPSS programs, respectively.

In the previous version, the analysis of results was performed by listing and analyzing the standard GPSS/FON tables manually. The added feature, ANALYZE, enables an automatic listing of the tables in any desired order. After examining the tables, the user can plot a histogram of desired data on the screen. The order of execution of ANALYZE options is arbitrary.

IMPLEMENTATION

The implementation was realized by using the standard definition of Pascal portable on many types of computers that support MS(PC)-DOS, Unix, VAX/VMS, etc. The source, consisting of about 10 000 compact code lines, is machine-independent [2]. However, the IBM-PC version, due to the graphics character of the ANALYZE option, is portable only on those systems that support Borland's Turbo Pascal.

Memory requirements for GPSS/FON are very small. For an average student's exercise, 640 Kbytes of PC memory is sufficient; simulation of a more sophisticated model is performed on a Honeywell X40 (Belgrade) or the VAX integrated network (Paisley), both being multiprocessor systems running under Unix. On the IBM PC AT286, the average transition time flow of one block instruction is about 1 ms; the execution of the same block instruction is at least ten times faster on the multiprocessor machines. These execution times are fast enough for both educational and research purposes [6].

USING GPSS/FON IN TEACHING DISCRETE EVENT SYSTEM SIMULATION

We start teaching GPSS by using a basic example of the production system with a single transaction flow, one servicing place and a single queue. After passing this introductory stage, a student is introduced to more complex models consisting of two or more servicing places and queues like those of small mini-markets. The third stage of training consists of fairly complex models of a coach station with two (or more) transaction flows and several servicing places. At the final stage of training, process communications with synchronized activities are presented to students [7]. At the end of this course, students are asked to build their own models having similar structures to those introduced at the last stage of their training.

To illustrate modelling and simulation with GPSS/FON, we present here an example of a student's work.

EXERCISE

The manager of a mini-market wants to improve turnover, without reducing services to his customers, at least not below the standard of other comparable businesses. At present, he operates below the ultimate capacity of his shop having only twenty baskets, with a fairly low utilization of tills. Based on his experience, queues longer than eleven customers are unacceptable.

Write the simulation model by using the following experimental data:

1. Maximum number of customers in the shop is 45 at any time; currently, the manager runs the shop with 20 baskets only and virtually no queuing at the tills.
2. There are two standard tills for any number of items and one express till for up to 5 items.
3. Shoppers arrive at variable average time intervals, depending on the time of day (see Table 1).
4. A shopper buys 10 items on average, with an empirical distribution, as shown in Table 2.

Table 1. Distribution of shoppers' arrival as a function of time

x.00 hours	9–10	10–11	11–12	12–13	13–14	14–15	15–16	16–17	17–18
Interval (seconds)	300	250	150	50	80	180	120	120	60

Table 2. Probability of a shopper buying *n* items

No of items	1	2	4	7	10	15	20
Probability	0.1	0.2	0.3	0.4	0.5	0.75	0.9

5. If baskets are not available, a shopper leaves the shop (a lost customer).
6. A shopper spends an average time of 20 s per purchased item, and an additional 200 s irrespective of the number of purchased items. Both times are given with exponential distribution and ±50% deviation.
7. A shopper queues at an appropriate till and spends (100 + 10) s per purchased item.
8. The shop opens at 9.00 h and closes at 18.00 h.

Simulate the model, and find the optimum number of baskets. Also advise the manager on possible changes in the existing set-up.

Simulate the model using the current set-up, and compare the results with your optimal findings.

We present here one of the best solutions in a slightly modified form. The model and conclusions are, however, those returned by the student.

SOLUTION

The GPSS/FON simulation model is given in Fig. 3.

ANALYSIS

By using the integrated environment, the student spent three hours, on average, developing the model, and an additional two hours, on average, on the simulation and resimulation of the shop performance. The following analysis was performed by using the listing-search and graphics options of ANALYZE.

1. Optimal results are obtained when 40 baskets were available. Only a minor change is necessary to improve the queuing times and queue lengths.
2. A total of 220 customers came into the shop between 9.00 and 18.00 h, 217 of whom were successful (only 3 lost customers).
3. The baskets were utilized 34.8% overall.
4. The express till was utilized 32.3% and the standard tills 73.2%.
5. Figure 4 presents a distribution of the total time spent in the shop over the number of customers who purchased up to five, and over five items, respectively. The histogram corresponds to the screen plot produced by the ANALYZE graphics option. (The histogram was produced by using Tables 11 and 12 of the *.res file, not reproduced here.)
6. Figure 5 (first model) presents a distribution of the number of queuing customers at one of the standard tills over their position in the queue. The histogram indicates that more than ten customers were tenth, twelfth and sixteenth in the queue, while ten and five customers were fourteenth and eighteenth in the queue, respectively. (The histogram was produced by using Table 13 of the *.res file.) The histogram, however, presents total numbers over the entire working day. In order to further our analysis, data was accumulated for each working hour between 9.00 and 18.00 h and stored in Tables 1–8 of the *.res file.
7. Figures 6a and 6b show the number of customers and their position in the queue as a function

```
*   GPSS/FON model of a mini-market (exercise number 3)
*
*   Lines separated by &&&&&&&&&&&& should be uncommented (* deleted) for the second model
             SIMULATE
             GENERATE FN$ATA,FN$EXPO  Shoppers arrive at the time dependent intervals
*
*
             ASSIGN 3,V$HOUR          Store the table no. for hourly statistics in the third
                                      parameter
             TRANSFER BOTH,,GOUT      Leave if baskets are not available
             ENTER BASKET             A basket is available and the shopper starts shopping
             ASSIGN 1,FN$NOITM        Assign the number of purchased items at random,
*                                     using the given distribution
             ADVANCE V$TIS,FN$EXPO    Assign the time spent in the shop
*
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*            TEST G Q1,20,TSTTILL     Check if 20 people are already waiting at the standard
                                      tills
*            TEST G V$QUE1HF,Q2,STTILL Compare average queue length and transfer a customer
*                                     to the shortest one
*            TRANSFER,ETILL
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*
TSTTILL      TEST G P1,5,ETILL
```

Fig. 3. GPSS/FON simulation model of a mini-market.

```
*
* Two tills for any number of items (>5)
*
STTILL   QUEUE 1                  Queues at standard tills
         TABULATE *3              Accumulate statistical data for a table *3; (1-8)
         TABULATE 13              Accumulate statistical data for the time spent in the
*                                 shopping area
         ENTER 1                  Come to a till
         DEPART 1                 Leave a queue
         ADVANCE V$STIM,V$DEV     Time spent at a till
         LEAVE 1                  Leave a till
         TABULATE 11              Accumulate statistical data for queuing times at
                                  standard tills
         TRANSFER,EXIT            Heading towards the exit
*
* One "express' till (<5 items)
ETILL    QUEUE 2                  Queue at the express till
         ENTER 2                  Come to a till
         DEPART 2                 Leave a queue
         ADVANCE V$STIM,V$DEV     Time at the till
         LEAVE 2                  Leave a till
         TABULATE 12              Accumulate statistical data for queing times at the
                                  express till
EXIT     LEAVE BASKET             Leave the basket
         TERMINATE                End of shopping
GOUT     TERMINATE                Leave the shop
* Timer
         GENERATE 2880            2880 t.u. 8 simulated hours
         TERMINATE 1
* Declare the permanent entities
*
1        TABLE Q1,0,1,35
2        TABLE Q1,0,1,35
3        TABLE Q1,0,1,35
4        TABLE Q1,0,1,35          Hourly statistics about the queues at the standard tills
5        TABLE Q1,0,1,35
6        TABLE Q1,0,1,35
7        TABLE Q1,0,1,35
8        TABLE Q1,0,1,35
*
11       TABLE M1,48,48,15
12       TABLE M1,48,48,10
13       TABLE Q1,0,4,10
*
1        STORAGE 2                Two standard tills
2        STORAGE 1                One express till
*
BASKET   STORAGE 40               Max. number of shoppers
*
STIM     VARIABLE P1+10           Service time
DEV      VARIABLE V$STIM/2        Deviation from STIM
*
TIS      VARIABLE V$STIM*2        Time spent in the shop
*
HOUR     VARIABLE C1/360+1        Working hours, to be assigned to the third parameter
*
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*QUE1HF  VARIABLE Q1/2            Average length of the queue at a standard till
*&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
*
NOITM    FUNCTION RN2,D7          Number of items—empirical distribution
         0.1,1/.2,2/.3,4/.4,7/.5,10/.9,15/.95,20
*
EXPO     FUNCTION RN1,C24         Exponential distribution
         0.0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/
         0.75,1.38/.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/
         0.94,2.81/0.95,2.99/0.96,3.2/0.97,3.5/0.98,3.9/0.99,4.6/
         0.995,5.3/0.998,6.2/0.999,7/9.9998,8
*
ATA      FUNCTION C1,C9           Average time of arrival—empirical distribution
         0,30/360,25/720,15/1080,5/1440,8/1800,18/2160,12/2420,6/2780,500
*
         START 1                  Start the simulator
         END
```

Fig. 3. (*cont.*)

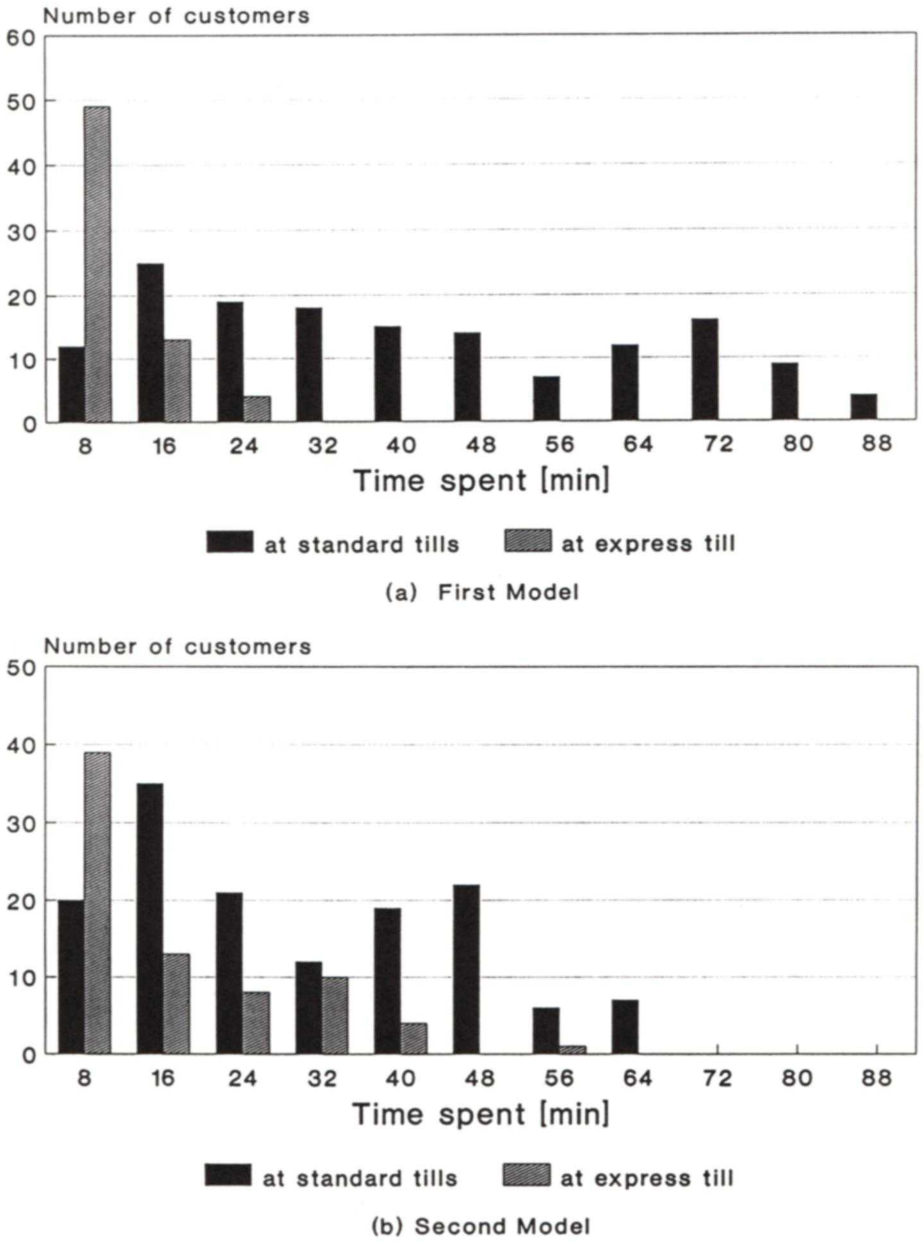


Fig. 4. Distribution of the number of customers over the total time spent in the shop.

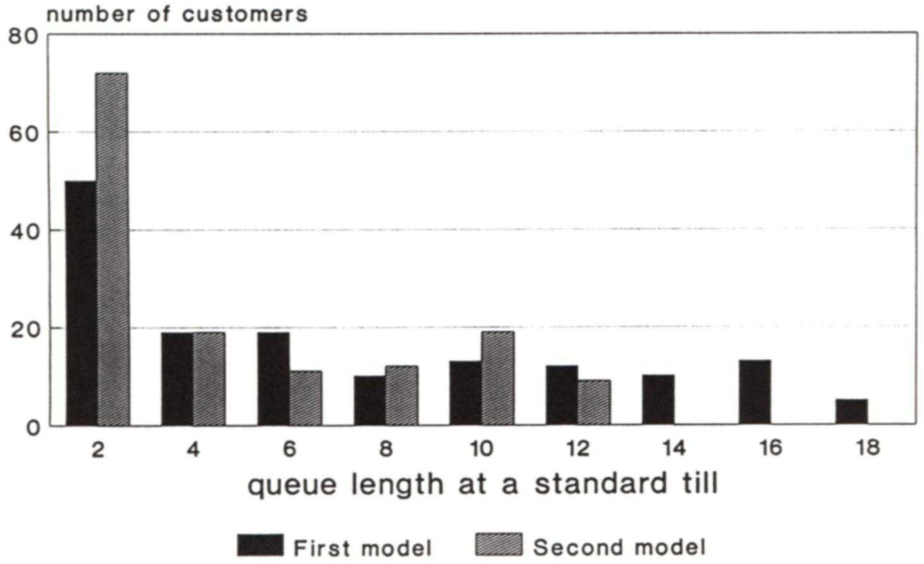
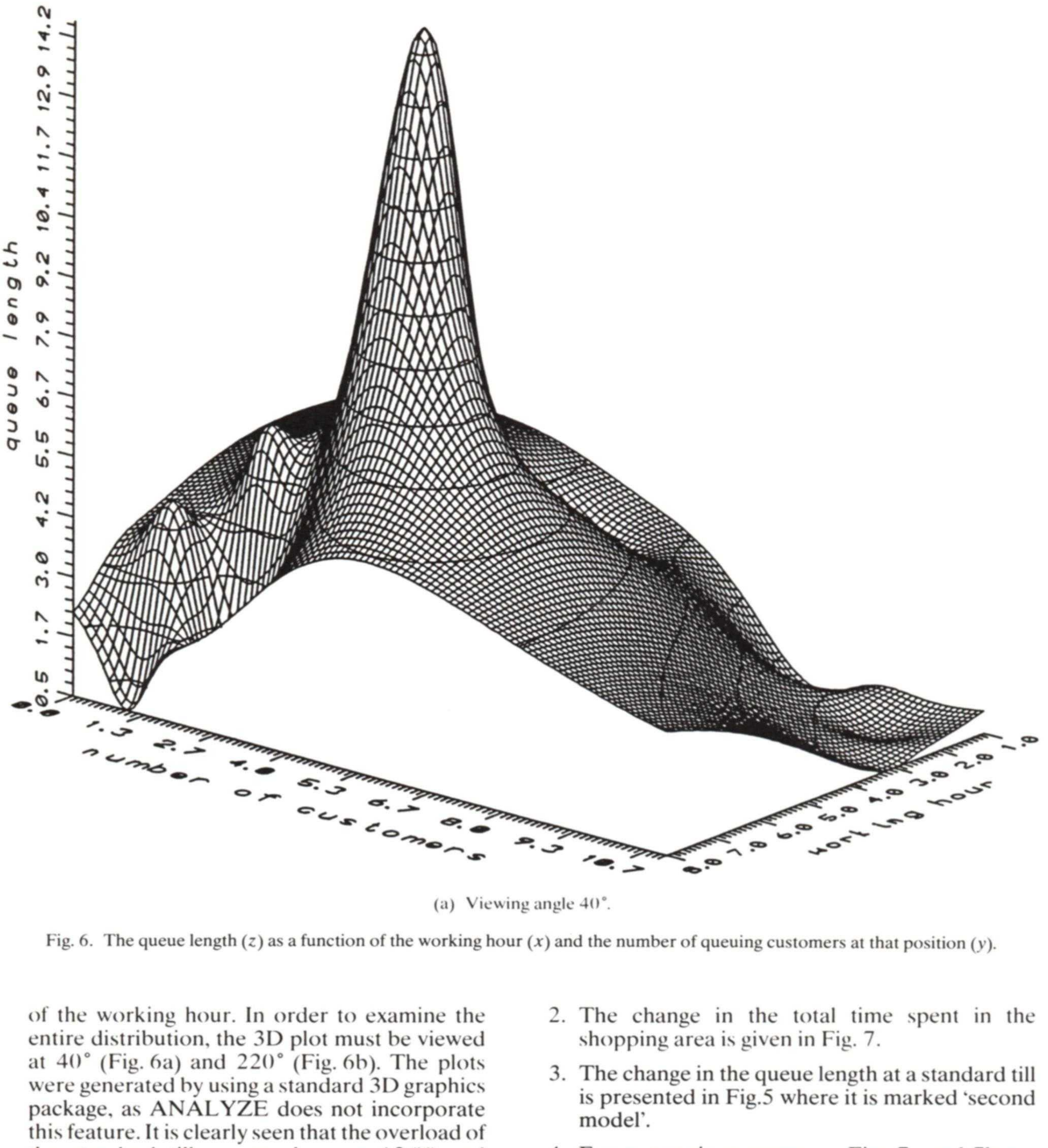


Fig. 5. Distribution of the number of queuing customers as a function of their position in the queue.



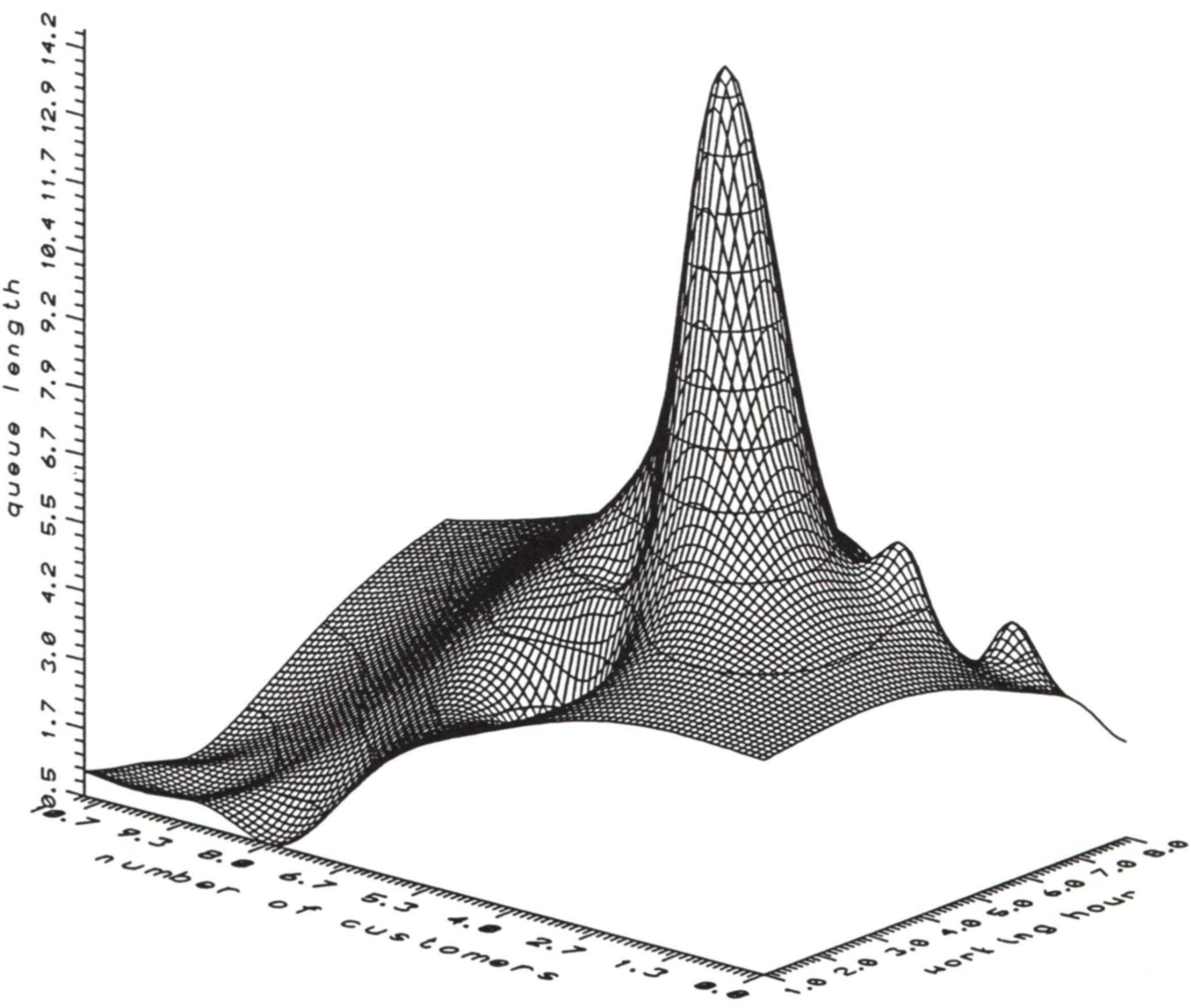
(a) Viewing angle 40°.

Fig. 6. The queue length (z) as a function of the working hour (x) and the number of queuing customers at that position (y).

of the working hour. In order to examine the entire distribution, the 3D plot must be viewed at 40° (Fig. 6a) and 220° (Fig. 6b). The plots were generated by using a standard 3D graphics package, as ANALYZE does not incorporate this feature. It is clearly seen that the overload of the standard tills occurs beween 12.00 and 14.00 h. The manager may be advised to either introduce an additional till between these hours or utilize all the available tills as standard ones, thus slightly inconveniencing those customers that purchase up to five items.

As the introduction of an additional till, used for only a couple of hours a day, can hardly be justified, the model was changed so as to simulate the second possibility. The new results can be summarized as follows:

1. The express till is now utilized 36% and the standard tills 63.9%.
2. The change in the total time spent in the shopping area is given in Fig. 7.
3. The change in the queue length at a standard till is presented in Fig.5 where it is marked 'second model'.
4. For comparison purposes, Figs 7a and 7b are given below. The viewing angles are 35° and 220° respectively.
5. The maximum queue length was 11 at standard tills and 5 at the express till. The total number of customers that checked through the express till was 75, while 142 customers were checked through standard tills, and 34 (45.3%) customers checked through the express till without queuing.



(b) Viewing angle 220°.

Fig. 6. (*cont.*)

COMPARISON

It is worth comparing the optimum results, with those of the first model if, 20 customers were set as the upper limit at any time. For an easier comparison, simulation results are given in Table 3.

ADVICE TO MANAGER

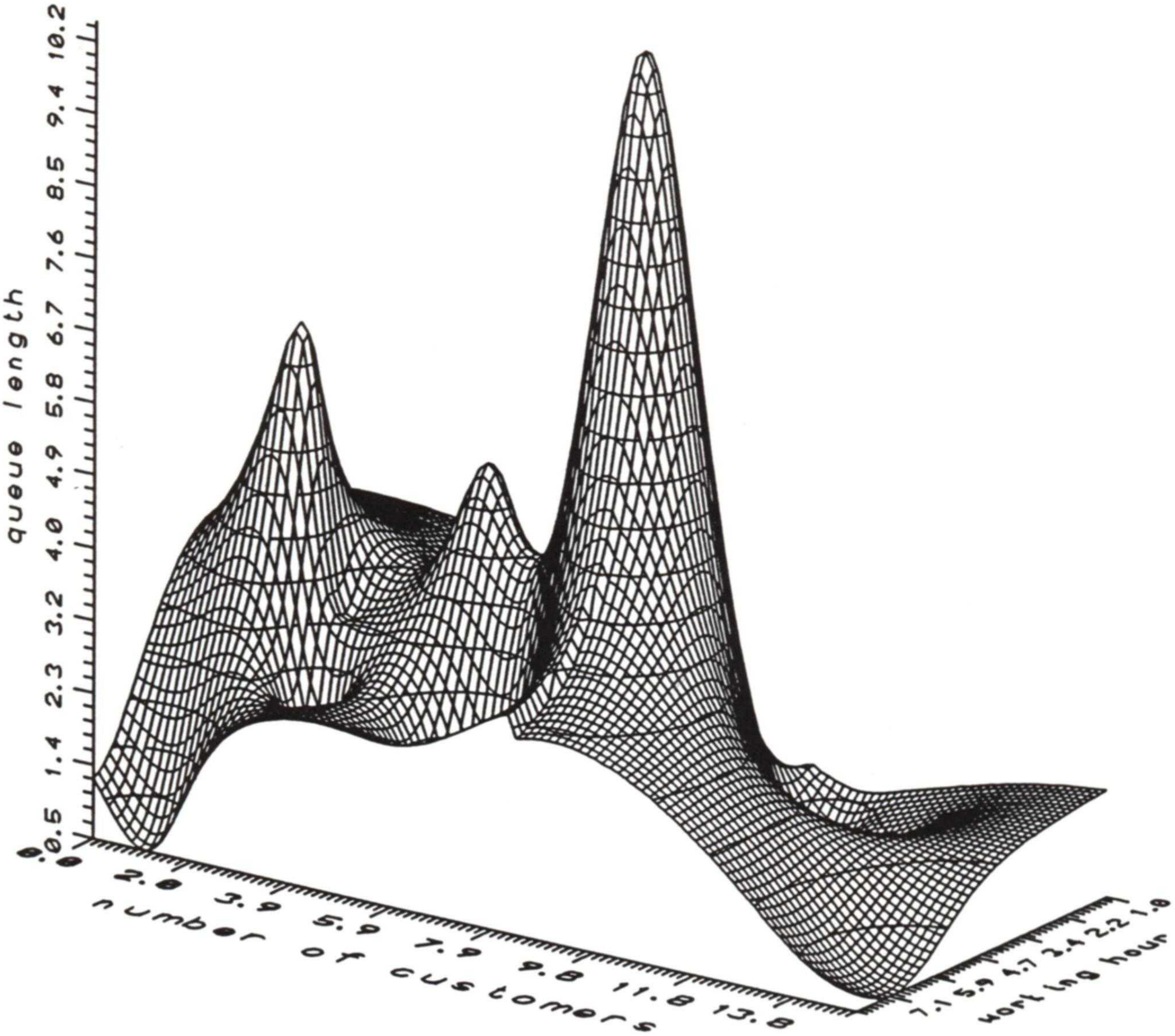
The manager is advised to:

- increase the number of baskets to 40;
- use all the tills as standard ones between 12.00 and 18.00 h or at any time when queues at standard tills increase to over ten customers.

The manager must be warned that there may be times when prospective customers fail to enter the shop after seeing unacceptably long queues, even if a basket is available. Before any advice regarding this situation can be offered, more experimental data (empirical facts) must be gathered, as attitudes of customers vary largely from one area to another. It was also noted that although the customers purchased ten items on average, it is more probable that they purchased more than ten items, than less. If the purchasing habits of the local customers change, the manager is advised to rerun the simulation.

CONCLUSION

Our experience, summarized in this paper, shows that the GPSS/FON dialect of the standard GPSS simulation language has many advantages



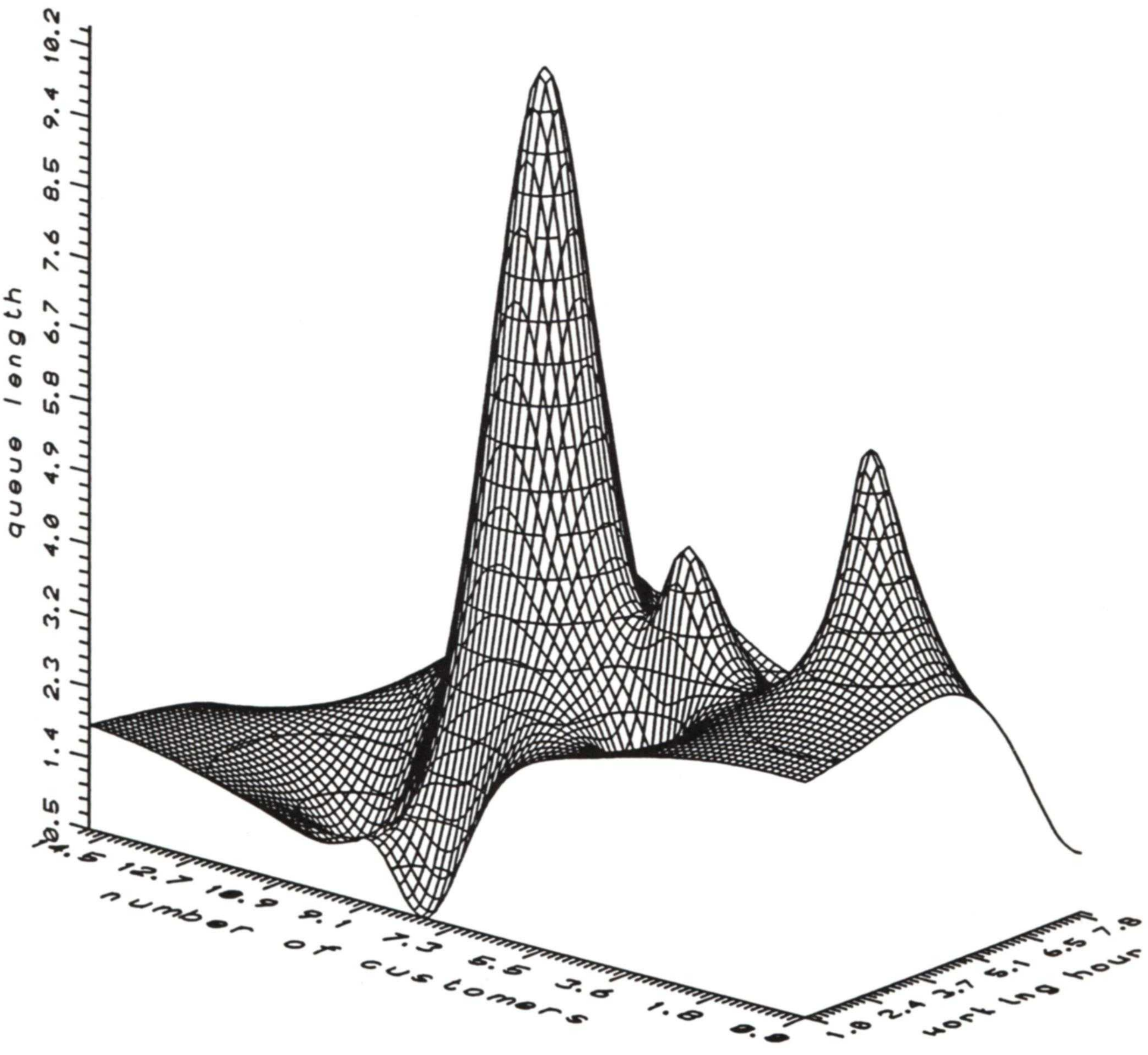
(a) Viewing angle 35°.

Fig. 7. Distribution of the number of customers over the total time spent in the shop using the second model. The queue length (z) as a function of the working hour (x) and the number queuing customers at that position (y), using the second model.

Table 3. Comparison of the current and the optimal shop performance

Model used	No. of customers	Lost customers	Baskets utilized	Queue length at standard tills	Queue length at express tills
First	220	32	36.4%	8 max.	2 max.
Optimum	220	3	24.6%	11 max.	11 max.

Model used	No. waiting at standard tills	No. waiting at express tills	Utilization of standard tills	Utilization of express tills
First	42 (29%)	41 (76%)	61.6%	26.3%
Optimum	40 (28%)	34 (45%)	63.9%	36%



(b) Viewing angle 220°.

Fig. 7. (*cont.*)

over the standard version in teaching environments, and virtually no drawbacks in research and other practical applications. The added features, such as the free format structure, and longer permissible variable names and labels, as well as the integrated environment of the IBM-PC version, provide a student or designer with a fast development and simulation environment. The ANALYZE option cuts down the simulation and analysis time to a small fraction of the time needed otherwise.

Our statistical analysis of students' performance shows that the IBM–PC version halves the development time, and improves the time spent on analysis by four times on average. The amount of work that can be forced upon an average student, and thus the acquired knowledge and experience that can be gained, has doubled since the new version was introduced, with no appreciable increase in effort required.

REFERENCES

1. T. J. Schriber, *Simulation Using GPSS*, Wiley, New York (1974).
2. B. ZLj. Radenkovic, *Simulation system for discrete-stochastic simulation and its implementation on mini- and microcomputers*, M.Sci. Thesis, University of Belgrade, Yugoslavia (in Serbian) (1987).

3. B. P. Zeigler, *Multifaceted Modeling and Discrete Event System Simulation*, Academic Press, London (1984).
4. J. W. Hopper and K. D. Reily, Simulation strategies, *Int. J. Comput. and Inf. Sci.*, **11**, 2 (1982).
5. B. Lj. Radenkovic, *Modelling and simulation using GPSS/FON*, Laboratory Manual, Faculty of Organizational Sciences, Belgrade (1988).
6. A. M. Zikic and B. Lj. Radenkovic, A Fully Portable Implementation of GPSS Language on 8 and 16 bits Microcomputers, *Proceedings 7th International Conference on Systems Engineering*, 773–779, Las Vegas (1990).
7. B. Lj. Radenkovic and A. Markovic, An application of GPSS/FON Language in Teaching of Simulation at University of Belgrade, *Proceedings of 1992 SCS Western Simulation Multiconference*, Paper No. 7032, Newport Beach, California (1992).
8. G. S. Fishman, *Principles of Discrete Event Simulation*, Wiley, New York (1978).
9. B. Lj. Radenkovic and A. Karkovic, *Theory of Computer Simulation*, Faculty of Organizational Sciences—Internal Publication, Belgrade (1990).

Dr Aleksandar Zikic is a lecturer at the University of Paisley, a position he has held since 1986. Previously he was lecturer, then consultant, and lastly head designer at the Institute of Applied Physics. He started his career at the University of Belgrade, Yugoslavia, as a junior lecturer in 1974, having graduated from there with B.Sci. in physics. He was granted his M.Sci. in instrumentation design in 1978, and his Ph.D. in control in 1984, again from the University of Belgrade. Between 1980 and 1982, and during 1985, Dr Zikic was a visiting researcher at Coventry Polytechnic, in the Department of Systems and Control. His interests are: real time computer control, mathematical modelling and computer simulation of analogue, discrete and discrete-event systems, and development and implementation of numerical algorithms.

Dr Bozidar Radenkovic is currently Assistant Professor at the Department of Organizational Sciences, University of Belgrade. He was granted his M.Sci. in 1987 and Ph.D. in 1990 at the University of Belgrade in the area of computer simulation. From 1987 until 1990, he worked at the Faculty of Organizational Sciences, as an assistant working on computer simulation and simulation languages. Since 1990, after becoming Assistant Professor for computer simulation he developed a version of the simulation system CSMP for continuus time system simulation and a self-contained portable version of GPSS for discrete-event systems. His primary area of interest is implementation of simulation languages for discrete event systems simulation.