References

Calkins, H.W., 1983, A pragmatic approach to geographic information systems design, in D. Peuquet and J. O'Callaghan (eds.) *The Design and Implementation of Computer-Based Geographic Information Systems.* (Amherst, NY: IGU Commission on Geographical Data Sensing and Processing).

Cameron, E.A., 1982, Manual digitizing systems, paper presented to the *Annual Convention of the ACSM/ASP*, Denver, Colorado.

Date, C.J., 1977, *An Introduction to Database Systems.* (2nd ed.) (Reading, MA: Addison-Wesley Publishing Co.).

Davis, W.S., 1983, *Systems Analysis and Design: A Structured Approach.* (Reading, MA: Addison-Wesley Publishing Co.).

DeMarco, T., 1979, *Structured Analysis and System Specification.* (Englewood Cliffs, NJ: Prentice-Hall, Inc.).

Marble, D.F., 1982, On the application of software engineering methodology to the development of geographic information systems, in D. Peuquet and J. O'Callaghan (eds.) *The Design and Implementation of Computer-Based Geographic Information Systems.* (Amherst, NY: IGU Commission on Geographical Data Sensing and Processing).

Marble, D.F., J.P. Lauzon and M. McGranaghan, 1984, Development of a conceptual model of the manual digitizing process. (Complete version). *Proceedings, 1984 International Symposium on Spatial Data Handling*, Zurich, Switzerland.

Page-Jones, M., 1980, *The Practical Guide to Structured Systems Analysis.* (New York, NY: Yourdon Press).

Peuquet, D.J. and A.R. Boyle, 1984, *Raster Scanning, Processing, and Plotting of Cartographic Documents.* (Williamsville, NY: SPAD Systems, Ltd.).

Yourdon, E., 1975, *Techniques of Program Structure and Design.* (Englewood Cliffs, NJ: Prentice-Hall, Inc.).

25

Performance evaluation and work-load estimation for geographic information systems

Michael F. Goodchild §

Department of Geography, University of Western Ontario, London, Ontario, Canada N6A 5C2

Brian R. Rizzo

Canada Land Data Systems Division, Lands Directorate, Environment Canada, Ottawa, Ontario, Canada K1A 0E7

Abstract

Agencies acquiring GIS hardware and software are faced with uncertainty at two levels: over the degree to which the proposed system will perform the function required, and over the degree to which it is capable of doing so within proposed production schedules. As the field matures the second is becoming more significant. A formal model of the process of acquiring a GIS is presented, based on the coneptual level of defining GIS sub-tasks. The appropriateness of the approach is illustrated using performance data from the Canada Land Data System. It is possible to construct reasonably accurate models of system resource utilization using simple predictors and least square techniques, and a combination of inductive and deductive reasoning. The model has been implemented in an interactive package for MS-DOS systems.

Introduction

The development of geographic information systems has now reached the point where substantial numbers of turnkey production systems are being acquired from vendors and installed in public and private agencies. In many cases these agencies will have made detailed plans for the use of the system before its selection, including evaluation of work-loads, and will have required potential vendors to respond

Reprinted from 1987, *International Journal of Geographical Information Systems*, **1**, pp. 67–76.

§ **Editors' note:** now at Dept. of Geography, University of California at Santa Barbara, California 93106

directly to these plans. The vendor in turn will have provided information on the extent to which the proposed system is capable of performing the prescribed work, in terms both of specific functions and overall utilization of system resources.

In many ways this ideal, objective and precise model of the process of acquiring a system is rarely achieved in practice. The agency must first identify the specific products which it expects to obtain from the system over the planning period and the numbers of those products. For example, a forest management agency would have to describe the maps and tabular outputs it would wish the system to produce over a period of, say, 5 years. This would be used as the basis for identifying firstly the data sets which would have to be input to the system, based on adequate measures of the volume and complexity of the data sets, and secondly the functions which the system would have to be able to perform on those data sets to generate the required products, based on an adequate taxonomy of GIS functions.

In responding, the vendor would first compare the list of required functions to the capacities of the system, a straightforward task given a precise and well-defined taxonomy of GIS functions. A hardware configuration would be selected which would provide adequate resources for the prescribed work-load, in terms of standard measures such as utilization of c.p.u. and disk storage, with a suitable margin for safety. Finally, the package would be proposed to the agency with a detailed commentary on the extent to which it did not satisfy the agency's requirements.

At this point, the agency is faced with the difficult task of evaluation . The simple 'yes' or 'no' response to functional capability is likely to be replaced by multiple shades of grey: 'Polygon overlay is under development and will be available in three months'; 'Polygon overlay is possible but extremely slow'; 'Polygon overlay is present but will not handle all special cases'. More difficult is the assessment of projected work-load. How reliable are the vendor's estimates of execution times of polygon overlays for projected workload, or input rates for digitizing? Will the same estimates be achievable in the agency's own environment and given the proposed arrangements for the maintenance of hardware and software?

The ideal course of action at this point would be to bench-mark the proposed system, thereby checking and refining the vendor's own stated assessment of functionality, and to make independent estimates of performance under the proposed work-load. The first objective would require a bench-mark script which tested each of the required GIS functions and assessed the response against an ordinal scale. We refer to this as a qualitative bench-mark of the system. Of particular interest are special cases: geometric conditions which are known to defeat simple routines for polygon overlay, for example. The second objective, which implies a quantitative bench-mark of measured resource use, is more difficult, and is discussed at length below.

The model of the process of acquiring a GIS presented above has several points of weakness, but can bring some degree of objectivity and regularity to what is otherwise an extremely risky and uncertain process. It assumes that at two points, in the generation of the vendor's response and in the agency's bench-mark evaluation, it is possible to predict the demands that a known volume of work will place on a known system configuration. This is the task which the computer science literature refers to as performance evaluation, and it will become increasingly important in the GIS field as the context changes steadily from research and development to production. The purpose of this paper is to investigate the extent to which performance evaluation is possible and useful within the GIS context, since there appear to be several reasons for believing that this context is significantly different from the Canada Land Data System (Canada Geographic Information System), which has been operating in a production mode with stable hardware and software and experienced staff for some years.

Performance evaluation

Performance evaluation relies on the assumption that it is possible to predict resource use for future tasks from a relatively small amount of data gathered by observation using tasks of known characteristics. To do so, it must be possible to break any task down into a number of standard types of sub-task, and to develop predictive models for each one. Performance against a future task is then predicted from the sum of its sub-tasks.

Early methods of performance evaluation relied heavily on defining sub-tasks at the level of the individual machine or Fortran instruction. Standards such as the Gibson mix (see, for example, Jones 1975) provided relative frequencies of the use of instructions in a general computing environment. However, the move to multi-tasking and interactive operating systems in the late 1960s made such methods inadequate: task performance could no longer be modelled as the accumulation of individual instruction sub-tasks because of effects of queuing and of the sequence of instructions (for reviews, see Chandy and Reiser, 1977; Beilner and Gelende, 1977; Ferrari, 1978; Hellerman and Conroy, 1975).

In the GIS field the appropriate level of sub-task would appear to be predetermined by the nature of current software. The agency acquiring a system is likely to specify its requirements at the conceptual level in order to avoid bias in favour of any particular GIS. Thus polygon overlay is probably an acceptable sub-task since it does not presume any particular data structure or algorithm, and can be performed in either raster or vector mode. But a lower level, such as the Fortran instruction, would not be acceptable because algorithms from different vendors for the same polygon overlay would have quite different instruction mixes. Similarly, data input is an acceptable sub-task, but scanning and digitizing are not because their use may vary from one vendor's proposal to that of another. The conceptual level of sub-task is also appropriate because it is the level at which most current system/user interfaces operate, and because it is a suitable level for the agency to use in the initial definition of products desired from the system. For example, it is relatively easy for forest managers to define an updated forest inventory map as the result of overlaying polygons of recent fires on existing forest inventory polygons, but any lower level of sub-task definition would presume substantial familiarity with one or more GIS.

There is a long history of debate in the performance evaluation field over the extent to which one should regard the system as a black box, observing the response of the system to given inputs in a purely empirical context, or whether the approach should be to some degree determined by knowledge of the algorithms being used. For example, we might expect the major factor determining execution time in an algorithm for a raster polygon overlay to be the size of the raster cell, whereas a vector algorithm would be more likely to depend on counts of polygons. Lehman (1977) makes this point and notes that the need for empirical, black box evaluation of performance is in fact somewhat paradoxical since the system under study is in principle perfectly understood. An interesting commentary on the field by Wegner (1972, p. 374) urges 'a proper balance between quantitative statistical techniques and qualitative techniques of structural analysis', although, somewhat surprisingly:

> 'Computer science is different in character from empirical disciplines such as agriculture or physics. Agriculture and physics are concerned with the study of natural phenomena, while computer science is concerned with the study of man-

made phenomena. A computer system generally has a far larger number of independently variable components than the systems studied in agriculture or physics.'

The debate would seem to be more complex in the GIS field where there is no control over the choice of algorithm used to perform a given sub-task, and where some of the operations being modelled are manual or contain substantial manual components. For example, it is essential to have a satisfactory model of digitizer throughput, including the time spent by operators correcting errors, if one is to make adequate projections of the number of shifts necessary to complete a given work-load of digitizing. In fact this has been one of the more uncertain elements in many acquisitions of GIS.

There is, of course, no chance that predictions of system use made from the results of performance evaluation will be perfectly accurate. Many of the factors influencing throughput cannot be predicted in advance, and others can be predicted only with considerable uncertainty. Obvious candidates in the first category are various types of failure of hardware and software. The task is best seen as a compromise between an excessively elaborate model on the one hand, which would require too much data and rigid adherence to planned production schedules and would be too sensitive to uncertainties, and, on the other, too little effort at assessing the degree to which the planned work-load lies within the capacity of the proposed system. We assume that the alternative of no prior evaluation of work-load is unacceptable.

The empirical or statistical approach to performance evaluation has been discussed in a number of articles (see, for example, Gomaa, 1976; Grenander and Tsao, 1972; Yeh, 1972; Bard and Suryanarayana, 1972; Racite, 1972), and the associated problems of experimental design have been discussed by Nelder (1979). The conventional technique is ordinary least squares regression, although Grenander and Tsao (1972) comment that its use cannot be too rigid since it is usually impossible to meet the inferential assumptions of the technique. Racite (1972) discusses the use of non-linear regression.

Formal model

A formal model and notation for the process of acquisition and bench-marking, following the conceptual outline given above, are now presented.

The agency has defined a set of products $R_1, R_2, \ldots, R_i, \ldots$, each one in the form of a map or tabular printout or some combination of the two, and each one requiring the execution of a sequence of GIS operations or sub-tasks. The number of each type of product required in each year j of the planned period is denoted by Y_{ij}. The sub-tasks are defined by an ordered set which may include several executions of the same type of sub-task, for example several polygon overlays. The sequence of sub-tasks for product i is denoted by

$$S_i = \{S_{i1}, S_{i2}, \ldots S_{it} \ldots\} \tag{1}$$

where each sub-task is drawn from a library L, $S_{it} \in L$ for all i, t.

Each sub-task a in the library is associated with a number of measures of use, drawn from a standard set M. Each measure m_{ak}, $a \in L$, $k \in M$, represents some demand on the system, such as c.p.u. time, operator time, plotter time or requirements for disk storage, with appropriate units of measurement. The value for each measure for a given task can be predicted from one or more predictors P_{akn}, drawn from a standard set P; $a \in L$, $k \in M$, $n \in P$. The predictors for each measure are quantities such as numbers of polygons which can be estimated in advance for each of the required products and used to estimate total resource utilization. Note that the set of predictors for a given measure may vary from sub-task to sub-task. The predictive equations for each measure are functions

$$m_{ak} = f(P_{ak1}, P_{ak2}, \ldots P_{akn}, \ldots) \tag{2}$$

calibrated by least squares regression or other means. The precise choice of function will be determined by a combination of empirical investigation and analysis of the structure of the sub-task.

To estimate the use of system resources, the required sub-tasks for each product are examined. The predictors for each measure are determined from the planned production schedule and used to evaluate the appropriate form of the predictive equation (2). Let W_{kit} represent the predicted utilization of resources measure k, $k \in M$, by the tth sub-task in generation of product i. Its value will be estimated by using the predictive equation for measure k in sub-task $a = S_{it}$. The predictors P_{akn} will be replaced by estimates of work-load determined from the planned production schedule, $P_{akn} = U_{kitn}$, $a = S_{it}$, where U_{kitn} denotes the planned value of the nth predictor of measure k for the tth sub-task of product i. In most cases predictors will be estimated by examining source documents.

The measures are then summed for the product as a whole

$$W_{mi} = \sum_t W_{mit} \tag{3}$$

and across products, weighted by the number required in each year

$$V_{mj} = \sum_i W_{mi} Y_{ij} \tag{4}$$

to give the total resource requirements which can be compared to known capacities.

Empirical analysis

We now examine the extent to which this approach can be usefully applied to an operational GIS. In particular, the following section looks at the extent to which the conceptual level is appropriate for the definition of sub-tasks, and whether useful predictions of work-load can be made at this level. In terms of the formal model, we examine the extent to which the prediction functions f can be determined by analysis of bench-mark performance data. We then describe an implementation of the model which takes bench-mark results and combines them with planned work-loads to make estimates of resource utilization.

The Canada Land Data System (Canada Geographic Information System (CGIS)) was designed in the early 1960s as a system for input and analysis of a national land capability survey, consisting of multiple layers of polygon data. Its most significant features are the use of a scanner for data input, conversion to vector organization for storage, and a raster algorithm for polygon overlay. Other features of the system will be noted during the discussion which follows. The data to be analysed were collected during regular production as part of the everyday internal auditing process of CGIS.

The data sets were all processed as part of a larger study of land-use change in Canadian metropolitan cities. Four coverages were processed for each of six cities:

Windsor, London, Kitchener, Hamilton, Regina and Montreal. All input was obtained from complete 1:250,000 map sheets, the number of sheets varying from two in London to nine in Montreal. One sheet was shared between Hamilton and Kitchener, so its input costs were incurred only once. In total 104 sheets were input, for each of 26 map sheets and four coverages.

Three major sub-tasks have been identified in the input process for the purpose of this study, and the resource utilization is expressed in dollars. Before scanning, each input document must be copied by hand using a scribing tool, to control width of line and to insure against spurious input. The costs of scribing (SCRIBE) are largely those of labour and can be assumed to depend on the length of polygon boundaries being scribed, and also to some extent on the irregularity of the lines and on the density of features. Following scanning, the raster data are vectorized and merged with polygon attributes in process referred to as steps 0 to 4, for which cost (denoted by Z4) is primarily a function of computer use. CGIS processes its data through a service bureau, so that costs given are those billed by the bureau, as distorted by peculiarities of the billing algorithm and such factors as overnight discounts. The third cost is that of manual error correction (MEC), which occurs during input processing and consists of the labour required to identify and remove errors detected by software during vectorization and polygon building.

Only one predictor is available for the three sub-tasks, in the form of a count of the number of polygons on each sheet. Although many more sensitive predictors might be obtained from the data after input, such as counts of coordinate pairs or line lengths, it is relatively easy to estimate polygon counts for typical map sheets in advance. The four coverages used in the study are shown in Table 1.

Table 1. Coverages used in the study.

Code	Theme	Mean polygon count
040E, F	Study area outline	3.2
100E	Recreation capability	59.7
200E	Agricultural capability	238.5
760X	Land-use change	1142.4

The theme of each sheet accounts for a large amount of the variance in input costs: 40.1 per cent of SCRIBE, 45.3 per cent of Z4 and 28.2 per cent of MEC. But almost all of this is because of variation in polygon counts; although each type of coverage has different conditions of shape of polygon and contortedness of line, disaggregating by coverage produces no significant improvement in the ability to predict costs once allowance has been made for polygon counts.

The best fit was obtained by a double logarithmic or power law model of the form

$$m = ap^b \tag{5}$$

where a and b are constants, calibrated by regressing the log of each measure against the log of the predicator, in this case log(cost) against log(polygon count). Logarithms are to base 10. The results from the 104 cases available are shown in Table 2.

The manual operation of scribing has the most predictable costs in terms of variance explained. Assuming no variance in shape, on purely dimensional grounds it would be expected that the total length of polygon boundaries on a map sheet would be proportional to the square root of the number of polygons. However, the regression shows that the costs of scribing rise with the 0.69 power, indicating that a higher density of polygons requires more effort per unit length of line than the added line length would suggest, presumably owing to the added complexity of working with high densities.

Table 2. Regression analysis of cost against polygon count.

Sub-task	Variance explained (per cent)	b	Standard error of estimate
SCRIBE	84	0.69	0.30
Z4	72	0.31	0.19
MEC	68	0.53	0.25

We expect the vectorization steps to be relatively insensitive to the number of polygons, and indeed the calibrated power is the lowest at 0.31, indicating that a doubling of cost will permit the processing of a sheet with approximately eight times as many polygons. The costs of manual error correction rise with the 0.53 power, suggesting either that the probability of error is dependent on length of line, or that the difficulty of correction is approximately twice as great for a sheet with four times as many polygons.

The standard errors of estimates are given above for each of the three sets of costs. Since the regression was performed on the logs of the costs, a standard error of e must be interpreted as meaning that the error of prediction from the model is typically a factor of 10^e. In the case of SCRIBE, which has the largest standard error, the typical error factor is therefore 2.0, meaning that we will commonly observe actual scribing costs which are half or twice the predicted value. Although this is a substantial uncertainty, it is very much less than the range of costs of scribing map sheets, which vary from a low of $2 to a high of over $2,000.

This method of computing prediction error must be treated as conservative for a number of reasons. Firstly, it assumes that the parameters in the model are estimated correctly. In reality, both a and b are subject to uncertainty, which in turn increases the uncertainty in predictions. Secondly, if we assume that residuals from the model are normally distributed, then the transformation which must be applied to allow for the use of logs will give a disproportionately large influence to large residuals. Thus, although 10^e may be typical of error factors, the mean error factor may be substantially higher.

After completion of the input steps, including edgematching of adjacent sheets, the data were merged into six databases, each with four coverages. The coverages were then overlaid using CGIS polygon overlay algorithm which employs raster techniques to superimpose vector data structures. Both c.p.u. time and billed cost were available as measures for each overlay, the relation between them being proprietary to the computer service bureau and compounded by CGIS decisions about job scheduling. Linear regression of overlay cost on overlay time showed that only 74 per cent of variance in cost is accounted for by variance in c.p.u. time for execution. Total input costs for each city's data were also available, but gave results which added little to those already obtained for the map sheet data: since the largest component of input cost is scribing, regression of total cost on polygon count gave results very similar to those shown above for SCRIBE.

The results of regressing log (overlay cost) and log (overlay time) on the logs of various polygon counts are shown in Table 3 in terms of the variance explained. The increase in uncertainty introduced by the billing algorithm is clear in all cases. Not unexpectedly, given the nature of the overlay algorithm, the best predictor is total output polygon count, reflecting the cost of revectorizing the image after overlay and building attribute tables for the new polygons. The estimated power is 0.44, which compares well with the power of the Z4 vectorization above. The standard error of estimate is 0.14, or an error factor of 1.4. Although counts of output polygons would not be available as a prior predictor of system work-load, they are linearly related to total input counts for these data; each input polygon generates, on average, 2.54 output polygons, the input count explaining 85 per cent of the variance in output count. The standard error of estimate if log (input count) is used to predict overlay time rather than log (output count) is 0.16 rather than 0.14.

Table 3. Variance explained when overlay cost and overlay time are regressed against polygon counts.

Polygon count	Time (per cent)	Cost (per cent)
Total output	85	31
Total input	79	27
040E/F	80	53
100E	59	44
200E	81	46
760X	73	21

Also shown are the results of using polygon counts from each of the four coverages individually as predictors. Although none is as successful as total output, it is interesting that the counts of polygons on three of the input coverages explain almost as much variance. The results confirm an expectation that c.p.u. time in a polygon overlay would be closely related to polygon counts on the most dense input coverages, but also suggest that polygon counts on the least dense coverages are also useful predictors.

From this analysis it appears to be possible, given stable software and hardware and sufficient data, to model the performance of a GIS at the level of the conceptual GIS sub-task, and to obtain reasonably accurate predictions of resource use. As was noted above, there is no possibility of perfectly accurate modelling; on the other hand, any reduction in uncertainty is presumably better than pure guesswork in system planning. The same basic approach of curve fitting seems to be equally as suitable for machine use as for purely manual and mixed manual and machine operations. The next section describes the operationalization of the complete model, including calibration steps and work-load estimation for a set of planned products, in an interactive package.

Implementation

The first author and Tomlinson Associates have implemented the formal model and the calibration procedures discussed above in a package for MS-DOS systems identified as SPM. It is structured in eight interdependent modules linked by a master menu, as shown in Table 4. Module 6 allows the user to choose from a wide range of possible models, including additive and multiplicative combinations of predictors and various transformations of variables. The values of constants can be obtained either by ordinary least squares, or by direct input by the user.

A recent test of the approach used data obtained by Tomlinson Associates from a study of the GIS requirements of a US National Forest. Forest Service staff had previously identified a total of 55 GIS products which they planned to use in their resource management activities in the first 6 years of operating a GIS. The combined

Table 4. Master menu showing modules of the SPM system.

Module	Function
1	Build, edit or retrieve the library of sub-tasks L.
2	Input ordinal performance scores for each sub-task from the results of a qualitative bench-mark test.
3	Input definitions for a set of required products $R_1, R_2, \ldots, R_i, \ldots$, including required processing steps.
4	Generate a statistical report based on the ability of the system to produce the required products, given the input performance score.
5	Input values of suitable performance measures chosen from M, and predictors chosen from P, from the results of a quantitative bench-mark test.
6	Construct and calibrate suitable models f of each sub-task from the data input in the previous step.
7	Input predictor values U measuring intended system workload for each product.
8	Compute and generate a statistical report giving cumulative estimates W of resource use for the intended work-load.

production task required a total of 65 coverages or data types to be input to the system, and a total of 51 different GIS functions or sub-tasks to perform the required manipulations. The number of sub-task steps required for each product ranged from five to 24.

Because of the effort involved, bench-mark performance models were constructed using SPM only for the eight most resource-intensive sub-tasks, including polygon overlay, generation of buffer zones and edgematching. Four measures were used: c.p.u. time, personnel time, plotter time and disk storage bytes. The predictive models relied on a total of 11 different measures, including polygon, line and point counts as appropriate to each sub-task. The final results were expressed in terms of total resource requirements for each product in each year of production, given the bench-marked hardware and software configuration.

Discussion

Agencies acquiring GIS have had to contend with considerable uncertainty, firstly over whether the system being acquired could indeed perform the necessary manipulations of spatial data, and secondly over whether the computing resources of the system were sufficient to meet required production schedules. GIS software has now reached a stage of development where much of the first form of anxiety has been removed: functions such as polygon overlay and the generation of buffer zones now perform with reasonable efficiency in most systems. However, not many models of system performance required to reduce uncertainty of the second type yet exist.

The most critical step in modelling performance is the definition of sub-tasks. The conceptual level of defining sub-tasks used in this paper matches the level used for most user interfaces in GIS, and is readily understood by agency staff not otherwise familiar with GIS operations and concepts. The empirical section of this paper has shown that it is possible to model performance at this level, even though sub-tasks may include substantial manual components and may have to allow for unpredictable events such as the failure of hardware.

It was noted earlier that any successful attempts at modelling must not simply approach a system as black box, but use knowledge of the complexity of sub-tasks and GIS algorithms to anticipate appropriate predictor variables and their role in the form of predictive models. This point also applies to the design of bench-marks, since the same arguments can be used to make suitable choices of measures and predictors, and to design appropriate variations of the key parameters. The number of independent runs required to obtain a reliable calibration of a given model is also determined by the number of variables and constants appearing in the model; conversely, the choice of possible models is constrained by the number of independent bench-mark tests made of each sub-task.

In this paper it has been assumed that the hardware and software configuration bench-marked is also the one proposed for production: no attempt has been made to develop models valid across configurations. To do so would add a new level of difficulty to the modelling which is outside the context of the present study. On the other hand, the choice of the conceptual level for defining sub-tasks allows the same general strategy to be followed whatever the configuration.

This last point restricts the applicability of this approach to the context defined in the introduction, that of a vendor or agency wishing to make a reliable estimate of resource use for a given work-load and a given system. It is not useful for an agency wishing to make a comparison between alternative systems, except as a means of developing information which might later form the basis of the comparison.

Acknowledgement

We wish to acknowledge the assistance of the Canada Land Data Systems Division, Environment Canada, in providing data for this study.

References

Bard, Y., and K.V. Suryanarayana, 1972, Quantitative methods for evaluating computer system performance: a review and proposals. In *Statistical Computer Performance Evaluation*, edited by W. Freiberger (New York: Academic Press), p. 329.

Beilner, H., and E. Gelende, (editors), 1977, *Measuring, Modelling and Evaluating Computer Systems* (New York: North Holland).

Chandy, K.M. and M. Reiser, (editors), 1977, *Computer Performance* (New York: North Holland).

Ferrari, D., 1978, *Computer Systems Performance Evaluation* (Englewood Cliffs: Prentice Hall).

Gomaa, H., 1976, A modelling approach to the evaluation of computer system performance. In *Modelling and Performance Evaluation of Computer Systems*, edited by E. Gelende (New York: North Holland), p. 171.

Grenander, U., and Tsao, R.F., 1972, Quantitative methods for evaluating computer system performance: a review and proposals. In *Statistical Computer Performance Evaluation*, edited by W. Freiberger (New York: Academic Press), p. 3.

Hellerman, H., and T.F. Conroy, 1975, *Computer System Performance* (New York: McGraw Hill).

Jones, R., 1975, A survey of bench-marking: the state of the art. In *Bench-marking: Computer Evaluation and Measurement*, edited by N. Benwell (New York: Wiley), p. 15.

Lehman, M.M., 1977, Performance evaluation, phenomenology, computer science and installation management. In *Computer Performance*, edited by K. M.

Chandy and M. Reiser (New York: North Holland), p. 1.

Nelder, J.A., 1979, Experimental design and statistical evaluation, In *Performance Evaluation of Numerical Software*, edited by L.D. Fosdick (New York: North Holland), p. 309.

Racite, M.P., 1972, The use of pure and modified regression techniques for developing systems of performance algorithms. In *Statistical Computer Performance Evaluation*, edited by W. Freiberger (New York: Academic Press), p. 347.

Wegner, P., 1972, Discussion of Section V. In *Statistical Computer Performance Evaluation*, edited by W. Freiberger (New York: Academic Press), p. 372.

Yeh, A.C., 1972, An application of statistical methodology in the study of computer system performance. In *Statistical Computer Performance Evaluation*, edited by W. Freiberger (New York: Academic Press), p. 287.