

Capacity planning, benchmarking and evaluation of small computer systems

A. ARBEL and A. SEIDMANN

Department of Industrial Engineering, Faculty of Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract: A decision framework for implementing small computer systems with particular emphasis on business is presented. The analysis presented here is applicable to systems involving a few microcomputers, but the time required of decision makers is especially justifiable in the case of acquisition problems involving a large number of units. Such problems are typical to large organizations where the intensive deployment of standard microcomputers has to be a carefully planned process. The decision framework integrates relevant decision elements in a hierarchical model that permits their systematic analysis. This analysis is concerned with capacity determination and prioritizing all decision elements in both cost and benefit categories. The priority structure permits direct evaluation of proposed systems with respect to users' needs and criteria. The decision process described here has been put to use in a few *real* acquisition problems.

Keywords: Decision theory, analytic hierarchy process, microcomputers

1. Introduction

The continued reduction in cost/benefit ratio have created a decision problem for many organizations examining the proper way to implement microcomputer systems for internal use. This is especially true when a large number of microcomputers are concerned. First, there is the financial outlay that far exceeds the price tag of the single unit and, more importantly, such an introduction should be a carefully planned process taking into consideration the specific organization implementing the system [2].

There exists a large body of knowledge developed for performance evaluation of mainframe computers, (e.g., [1,4,12]). While these methods proved useful for mainframe computers, alternative approaches have to be developed for the microcomputer decision problems. The main thrust in evaluating mainframe computers is concerned with its data processing and computational capabilities for users with significant computer skills. In contrast, microcomputers are judged by their

direct contribution to 'human effectiveness' that relates machine performance to individual users capabilities. In a recent paper McFarlan and McKenney [9] studied the major factors that encourage the fast deployment of stand-alone mini- or microcomputer systems and the resulting managerial issues. As the 'micros' become more powerful, it is desirable to establish a decision framework for the introduction of microcomputers that examines users needs, determines operational requirements and integrates all decision elements in a performance model supporting the acquisition decision. The structure of the paper is as follows. First, a problem statement is provided, followed by a description of the determination of operational requirements for a specific application, a generic performance model capable of handling diverse applications and, finally, it concludes with a performance analysis process leading to the selection of the desired microcomputer system.

2. Problem statement

In striving for 'human effectiveness' in the implementation of microcomputer-based systems, one faces a multistage decision problem. In the first

Received August 1984; revised April 1985

stage the organization considers the question of continuing the current practices (non-computerized operations) as opposed to the introduction of the computerized system to support the daily activities. The issue here is affected not only by the direct economic costs but rather by the impact such an introduction may have on the effectiveness of operations. Once such a commitment has been made, desired performance levels for the prospective systems are established. These are based on the operational and information requirements identified in the first stage. These parameters may include such items as main memory size, disk capacity, printer capability, required organizational changes and gross budget allocation to support this endeavour. All these activities make up the *system definition* phase.

The next decision stage is concerned with *system evaluation*. This stage considers proposals from various vendors responding to the Organization's RFP issued based on the results of the system

definition effort. The evaluation effort is concerned both with the 'cost' and the 'benefit' aspects of the decision problem. The result of the overall effort is a particular system to be selected and installed on-site. The overall description of the decision process is summarized in Figure 1.

The methodology outlined in this paper has been applied in several cases where large organizations (e.g., insurance companies or industrial corporations) evaluated their 'standard' micro (or mini) computer systems. These systems were installed for decentralized or distributed applications.

3. Operational requirements

The first step in the acquisition process leading to a computerized mode of operation is that of system definition (cf. Figure 1). This step determines the performance levels to be met by the system to be procured. In determining such performance levels, operational requirements supporting specific applications are defined. These performance levels consider specific organizational needs and capabilities, and their realization through hardware and software elements have to anticipate future growth. Performance levels are generally concerned with both software and hardware aspects; it is assumed however, that basic software requirements have already been established and, therefore, are not described in the analysis outlined here. Determination of performance levels is concerned here mainly with the operational load on the system, so that a microcomputer system with its limited capabilities will accommodate the anticipated load. The exact determination of computers' processing capabilities is a complex issue and the extensive effort undertaken in this determination is unwarranted in the case of small computers. The approach taken in this paper—and described next—provides an implementable scheme for workload analysis of microcomputer systems. Readers unfamiliar with basic contemporary computer science terminology may refer to [3,4].

Initial load calculations are based on the concepts of *Software Physics* [8]. The basic measurement unit is the Software Unit Work (SUW) representing the amount of work done by the processor in transferring one byte (eight bits) to a

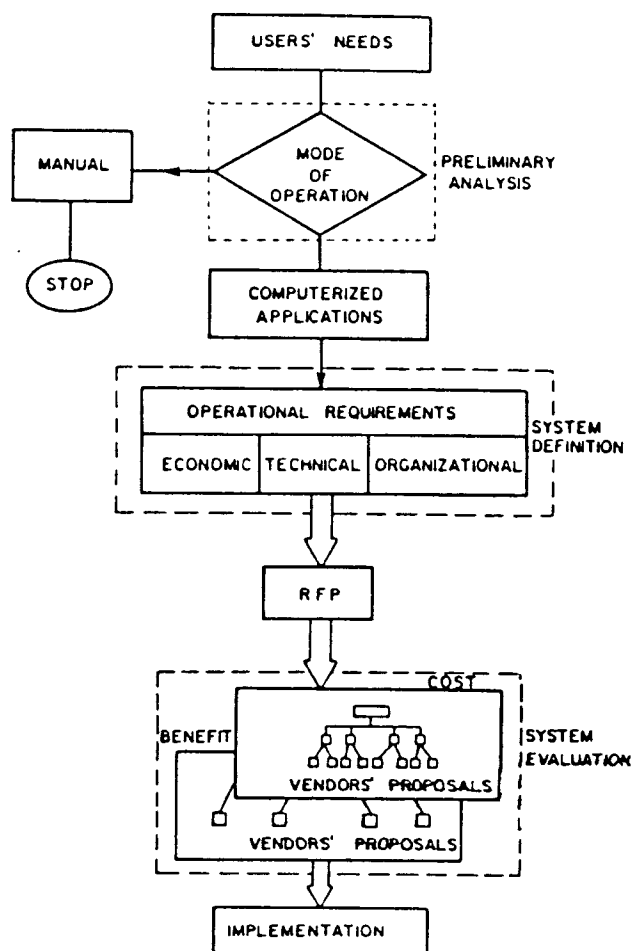


Figure 1. The decision process

given storage device. The storage device may be the RAM, disk driver, a CPU register or a printer. The work done by a given microcomputer system can be characterized according to several equipment class components. These components comprise the *Software Work Vector* (SWV). The basic structure of a Software Work Vector having V components is:

$$SWV = \begin{bmatrix} 1. \text{ CPU} & \text{work} \\ 2. \text{ Disk} & \text{work} \\ 3. \text{ Hard disk} & \text{work} \\ \vdots & \\ V. \text{ Printer} & \text{work} \end{bmatrix}$$

The estimation of these components can be based on software monitoring routines or on direct computations. For example, in peripherals, the software work is equal to the number of I/O transfers times the average block size in bytes. Given that a job does α_1 units of work on equipment class 1 (say, CPU), α_2 units of work on equipment class 2 and so on—its total work is the sum of $(\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_V)$. Denoting the total work units $\sum \alpha_i = W$ then the SWV is

$$SWV = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_V \end{bmatrix} = W \cdot \begin{bmatrix} \alpha_1/W \\ \alpha_2/W \\ \vdots \\ \alpha_V/W \end{bmatrix} = W \cdot \begin{bmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_V \end{bmatrix}$$

Clearly $\sum \hat{\alpha}_i = 1$, and the vector of these components is called the *Software Unit Vector* (SUV). This vector is (roughly) constant for a given application regardless of the total work performed, or the specific microcomputer system in use. Given the amount of total work units in a time period (W) one may retrieve the SWV vector by multiplying it by the Software Unit Vector. The result characterizes the application workload by equipment class.

In order to prepare forecasts of future work load one needs to estimate the projected total works unit (W). In many applications, however, it may be difficult to provide direct estimates for W . One common way to overcome this difficulty is to find the number of work units per (natural) business transaction. For example, dividing the total software work by the number of transactions it may be estimated that one transaction (such as

inventory withdrawals or sale-slip processing) requires 500 units. Given the projected activity requirements for the application in question one may derive from them the future value of W and the resulting SWV.

Dividing software work by time leads to the *throughput power*. If T_c denotes the elapsed time for the entire microcomputer system, then the resulting throughput power vector is:

$$\text{Throughput power} = \frac{\text{Software power}}{T_c} = \begin{bmatrix} \alpha_1/T_c \\ \alpha_2/T_c \\ \vdots \\ \alpha_V/T_c \end{bmatrix}$$

The discussion presented here focuses, for brevity, on systems having minimal multiprocessing capabilities. The mathematical derivations for multiprocessing systems can be found in [4] and [8]. With no multiprocessing, the ratio α_i/T_c traces the average power usage of equipment class i . Its absolute power usage is given by $AP(i) = \alpha_i/T_i$, where T_i is the execution time of equipment class i , ($T_i \leq T_c$).

The applicability of Software Physics concepts to feasibility studies of microcomputers' configuration is illustrated next through an example representing inventory control in a small company. The mean number of inventory items in storage is 640 with standard deviation of 90. Each item is associated, on the average, with 11 orders a month with a standard deviation of 4. Thus, the total monthly transaction rate is 7040. Assume that this application requires 7500 work units per transaction and assume that the SUV for inventory control application is given by:

SUV

$$= [\alpha_1 = \text{CPU}(0.8908), \alpha_2 = \text{Keyboard}(0.0012), \alpha_3 = \text{Hard disk}(0.026), \alpha_4 = \text{Diskettes}(0.031), \alpha_5 = \text{Printer}(0.051)]$$

With the given workload, it is now possible to assess the feasibility of *realizing* the required application through a *specific* configuration. A sample microcomputer system is characterized by the following average absolute power (Bytes/second) of its components:

$$\begin{aligned} AP_1(\text{CPU}) &= 9300; & AP_2(\text{Keyboard}) &= 0.28; \\ AP_3(\text{Hard disk}) &= 312; & AP_4(\text{Diskettes}) &= 90; \\ AP_5(\text{Printer}) &= 28. \end{aligned}$$

(For detailed computations of the average absolute power the reader may refer to [10] and to the data sheets supplied by the vendors.)

It is well known [12] that the nested random variable of the form

$$X(s) = N[B(s)]$$

has mean

$$\mu_X = \mu_N \cdot \mu_B$$

and variance

$$\sigma_X^2 = \mu_N \sigma_B^2 + \mu_B^2 \sigma_N^2$$

(random sum of random variables).

Therefore, the total work units per month in the inventory control application amounts to

$$W = (11)(640)(7500) = (528)10^5$$

with

$$\begin{aligned} \sigma(W) &= [(11)90^2 + (640^2)4^2]^{1/2}7500 \\ &= (193.3)10^5. \end{aligned}$$

The total time required to process the monthly workload is

$$[(528)10^5(0.8908, 0.0012, 0.026, 0.031, 0.051)]$$

$$\begin{bmatrix} 1/9300 \\ 1/0.28 \\ 1/312 \\ 1/90 \\ 1/28 \end{bmatrix} (1/3600) = 97.25 \quad (\text{hours/mo.}).$$

Assuming 180 working hours a month, the microcomputer will be utilized, on the average, 54 per cent of the time; adding two standard deviations to the average load as a safety factor, the computed utilization amount to 93.5 per cent of the time. Sensitivity analysis may reveal the desirability of using various microcomputer components with different absolute power values. For example, one may wish to assess the impact of a faster printer on the expected system utilization.

Storage requirements for disks and diskettes follow the standard estimation procedures for file size and operating system overhead calculations

for sequential and indexed sequential files [3]. Consider, for example, the inventory system discussed here. Assuming 31 characters per transaction and 7040 transactions a month one requires storage space for 218240 characters in order to store the volume of the monthly transactions files, or 2618880 on an annual basis. While a diskette may be sufficient for the monthly transactions file, a disk is required to store the annual transactions file. In practice, there is a need to multiply the computed volume by a safety factor of 1.5 to 2.5 in order to accommodate for the DOS (Disk Operating System) requirements and for periodic volume fluctuations [5]. Following the System Definition phase the RFPs are issued to the vendors. The next decision step is, therefore, system selection through comparative performance evaluation.

4. Performance model

The system evaluation phase is based on comparing candidate systems with respect to a number of evaluation parameters. These parameters are grouped into two major categories: 'cost' and 'benefit'. The evaluation effort is directed at establishing priorities of cost and benefit parameters associated with each system. A number of cost parameters can be identified as relevant in a microcomputer selection problem. These may include such items as: local vendor, operating cost, and required customizing to name a few. Each of these parameters is, in turn, subdivided into more specific parameters. The importance of these parameters may shift depending on the point of view (criteria) used. Criteria relevant to the selection problem, include the set-up cost, daily operations aspects and dependability issues. These criteria and parameters are summarized in the hierarchy shown in Figure 2.

Observing the cost hierarchy one notes that not every parameter is relevant to every criteria. For example, maintainability is relevant to the daily operations aspects and to the dependability issues but irrelevant to the setup cost. The hierarchy provides a list of criteria and parameters whose prioritization facilitates the selection process. This prioritization will be carried out using the Analytic Hierarchy Process (AHP) developed recently by Saaty [11].

Next, one constructs the benefit model to be

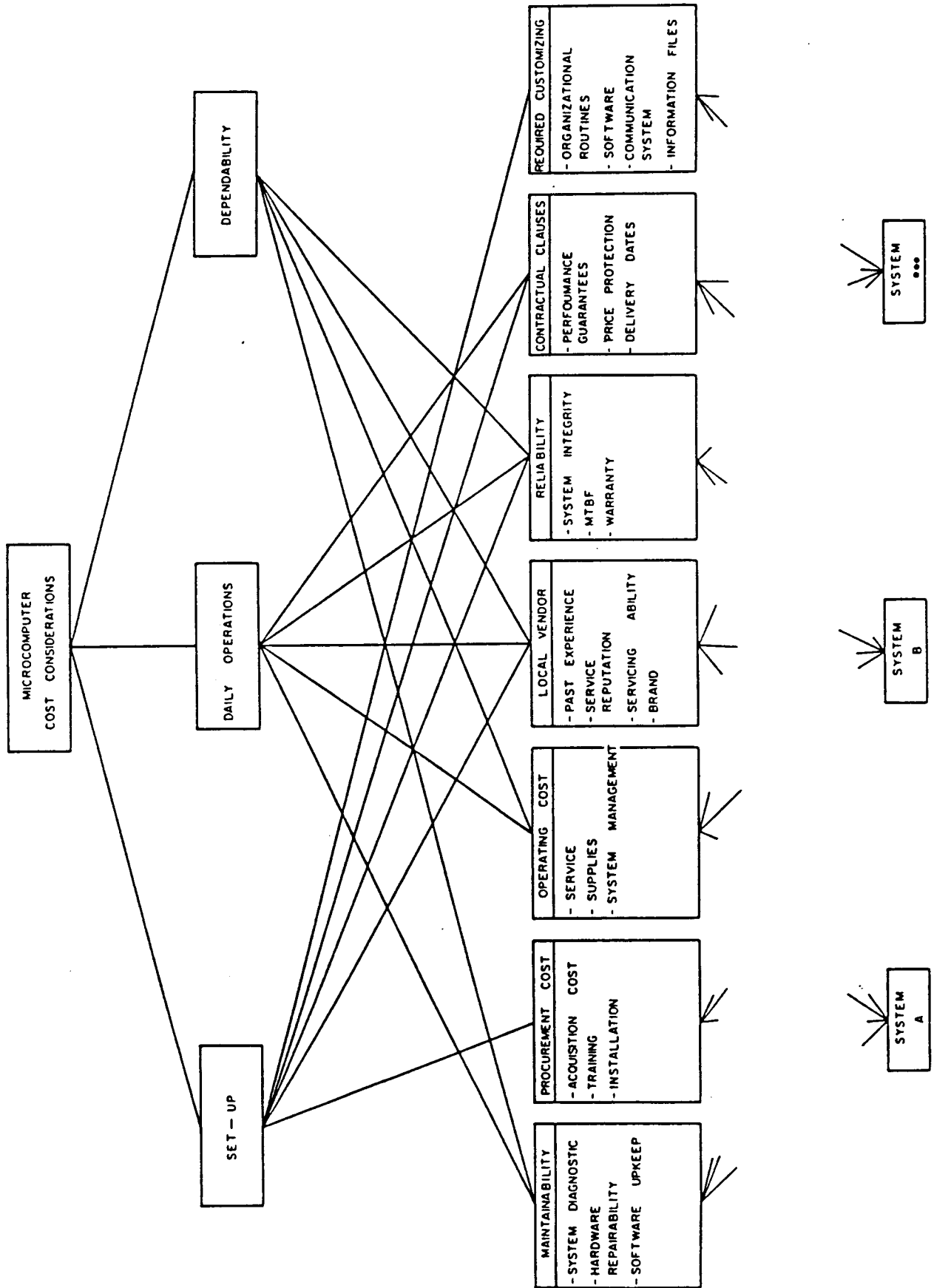


Figure 2. Microcomputer system cost hierarchy

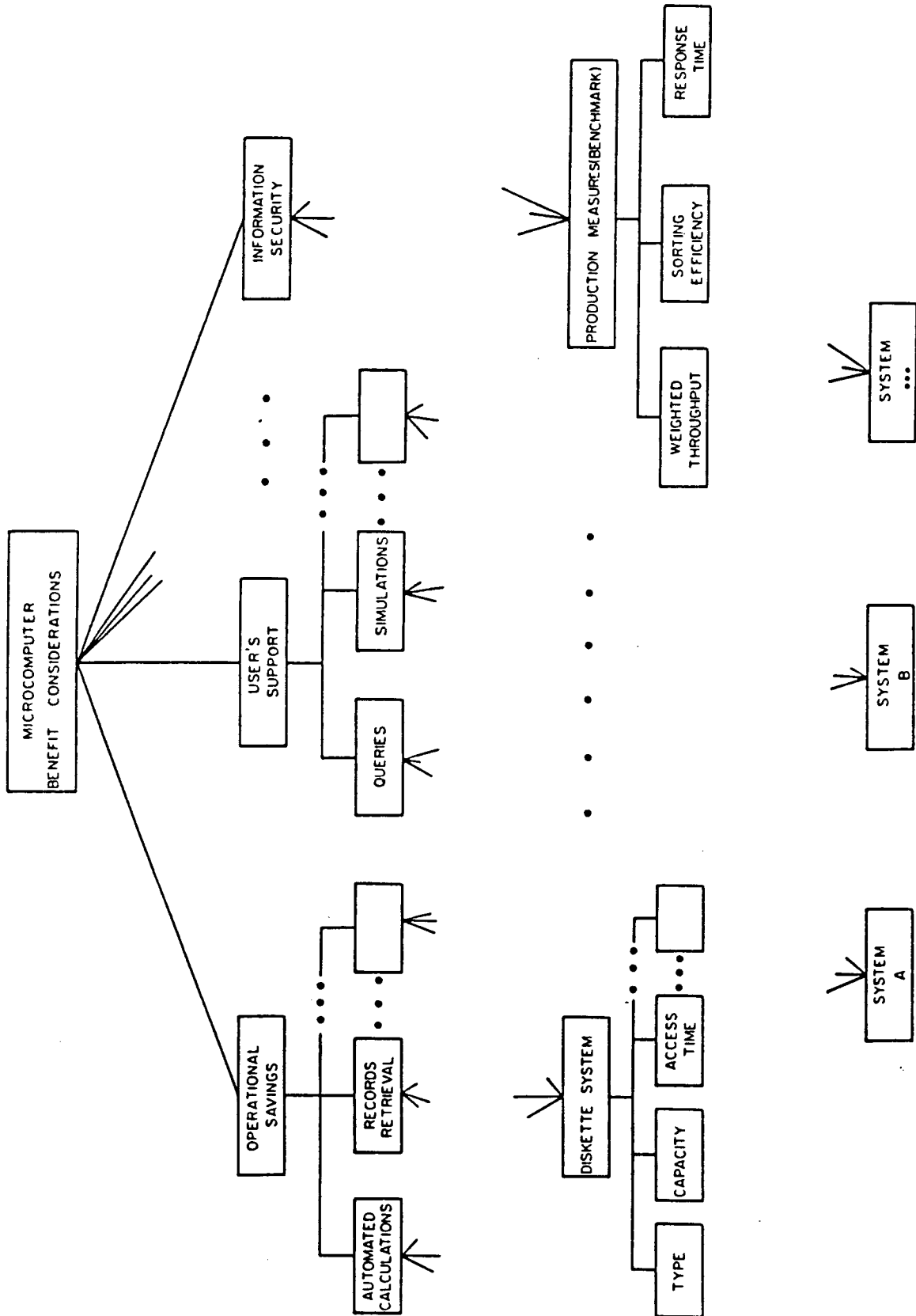


Figure 3. A sketch of microcomputer system benefit hierarchy

Table 1
Benefit criteria

- *Operational Savings*
- Automated Calculations; Records Retrieval;
- Updating Capability; Data Aggregation
- *User's Support*
- Queries; Simulations; Management Reports;
- Text (Word) Processing; Communication; Programmability
- *User Friendliness (Ergonomics)*
- Hardware Aspects; Software Aspects
- *Data Conversion*
- Analog-Digital; Parallel-Serial; ASCII - Binary
- *Record Keeping*
- Transactions Validation; Editing and Balancing;
- Controls and Auditability; Capacity
- *Information Security*
- Access Control; User Identification; Functional Integrity

used in the selection process. Similar to the cost model, the benefit model is also constructed as a hierarchy whose uppermost level describes relevant criteria and where the next level describes parameters. Criteria and subcriteria relevant to *general* microcomputer selection problems are summarized in Table 1. The criteria shown in this table provide a guideline for determining the operational criteria to be used in a particular problem.

For example, in selecting a financial analysis system one may employ such criteria as Controls and Auditability and disregard other criteria such

Table 2
Benefit parameters - Diskette System

- Type: Capacity (M-Bytes); Access Time (Millisecond/record);
- Data Transfer Rate (M-Bits/second); Diagonistics;
- Number of Drives
- *Printer*
- Technology: Speed; Noise; Print Columns;
- Font Flexibility; Print Controls
- *Files Management*
- Catalog: Automatic Indexing; Backup; Recovery;
- Protection; Retrieval
- *CPU & Memory*
- Word size; RAM; Operating Systems; Clock Frequency;
- Interrupt Handling; Addressing Modes
- *Systems Utilities*
- Editors & Loaders; General Purpose Routines; Debug Tools
- *Software Packages*
- Type: Relative Capability, Compatibility;
- Special Features; Suitability; Documentation
- *Expandability*
- Multi-user; Storage Upgrade; Software Features
- *Production Measures (Benchmark)*
- Weighted Throughput; Data Entry Rate; Peak Load Handling

as Data Conversion or Word Processing capabilities.

The next level in the benefit hierarchy is concerned with parameters based on which a direct comparison of candidate systems can be made. Such a list of parameters is shown in Table 2 which does not claim to be exhaustive. Again, the reader is reminded that only parameters relevant in a particular setting are selected from this list to form the parameters' level of the benefit hierarchy. Once relevant criteria and parameters are selected a benefit hierarchy can be constructed similar to the one shown in Figure 1. A sketch of such a hierarchy is shown in Figure 3; more elements can be added to it with the aid of Tables 1 and 2.

When criteria and parameters, in both the cost and benefit categories, have been identified for a particular application, one enters the quantitative phase of the analysis where priorities are assessed. This is done in the next section.

5. Performance analysis

Once minimal requirements for the system have been determined, RFP's issued, and vendors' proposals received, one enters the evaluation phase (cf. Figure 1). The evaluation phase employs the relevant cost and benefit hierarchies (cf. Figures 2 and 3) and starts by prioritizing its elements. This prioritization allows the comparison of vendors' proposals for arriving at an overall merit figures. The analytic methodology, employed for this phase is provided by the Analytic Hierarchy Process (AHP). The adoption of the AHP methodology to microcomputer selection is illustrated next.

The analysis starts with the top level of the cost hierarchy and aimed at deriving priorities associated with members of the criteria level. The derivation of priorities is obtained as a solution of an eigenvector problem of a certain comparison matrix. The comparison matrix summarizes pairwise comparison of elements in a particular level. Namely, if a level has n elements, than the $n \times n$ comparison matrix will have its elements given as $a_{ij} = w_i/w_j$. The w is the (normalized) vector of priorities solving $Aw = \lambda_{\max} w$ where A is the comparison matrix and λ_{\max} is its largest eigenvalue. The elements of the matrix A are estimated by the decision maker using entries taken from a 1-9 ratio scale [11]. For example, referring to the cost

Table 3
Comparison of cost criteria from Figure 2

Cost criteria	(1)	(2)	(3)	Priority
(1) Set up	1	$\frac{1}{4}$	$\frac{1}{6}$	0.09
(2) Daily operation	4	1	$\frac{1}{2}$	0.32
(3) Dependability	6	2	1	0.59
		$\lambda_{max} = 3.0092$		1.00

hierarchy shown in Figure 2, the criteria level contains three specific elements, whose pairwise comparison is summarized in Table 3. The entries of this matrix represent estimates of ratios of weights (priorities), e.g. 'daily operation' is estimated to have priority four times larger than that of 'setup' cost. Solving for the eigenvector associated with the largest eigenvalue yields the required vector of priorities given by the last column of Table 3. The largest eigenvalue provides a measure of consistency through its deviation from the dimension of the matrix. More on this topic can be found in [11].

The prioritization is continued by considering the elements in the next level. This is done by assessing the importance of cost parameters to specific criteria. Not all cost parameters are relevant to every criteria as is evident by observing the interconnection of elements in the cost hierarchy depicted in Figure 2. Five groups of parameters are important in assessing the setup cost; these are shown in Table 4. This group of parameters represents a partial set of the general parameters list and its selection was done by the decision maker who deemed them relevant. Once the list of relevant parameters is available one proceeds to prioritize them. This is done in a pairwise comparison manner, as was done earlier, and the re-

Table 5
Comparisons with respect to daily operations

Daily operations	(1)	(2)	(3)	(4)	(5)	Local priority
(1) Maintainability	1	$\frac{1}{3}$	1	4	4	0.24
(2) Operating cost	3	1	$\frac{1}{3}$	2	2	0.25
(3) Local vendor	1	2	1	4	5	0.35
(4) Reliability	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	1	1	0.08
(5) Contractual clauses	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{3}$	1	1	0.08
			$\lambda_{max} = 5.4711$			1.00

Table 4
Comparisons with respect to set-up

Set-up	(1)	(2)	(3)	(4)	(5)	Local Priority
(1) Procurement cost	1	2	3	2	1	0.29
(2) Local vendor	$\frac{1}{2}$	1	3	1	$\frac{1}{2}$	0.17
(3) Reliability	$\frac{1}{3}$	$\frac{1}{3}$	1	$\frac{1}{2}$	$\frac{1}{4}$	0.08
(4) Contractual clauses	$\frac{1}{2}$	1	2	1	$\frac{1}{2}$	0.15
(5) Required customizing	1	2	4	2	1	0.31
			$\lambda_{max} = 5.0303$			1.00

sulting matrix is shown in Table 4 with its resulting priority eigenvector.

Next, one considers the parameters relevant to the daily operations and to the dependability cost criteria. This relevant lists, their pairwise comparisons, and their priority vectors, are shown in Tables 5 and 6, respectively.

Finally one has to deal with the *global* priorities of the cost parameters; i.e., one is not interested in the importance of cost parameters with respect to specific criteria but rather, to the overall, global, cost issue. Table 7 summarizes the local priorities (i.e. with respect to specific criteria) of all seven parameter groups. Observe, for example, that the maintainability parameter was judged irrelevant with respect to the setup cost criteria, but of relatively high priority with respect to the daily operations (0.24) and dependability (0.31) criteria. Similar observations can be made with respect to all other parameter groups.

If one denotes by u_{ij} the local priority of the i -th parameter group with respect to the j -th criteria then the matrix U will have columns representing local priorities of parameters with respect to a particular criteria. In the case under consideration this matrix is of dimension 7×3 . If $w(i)$ represents the vector of global priorities of elements in level i , and there are m levels in the

Table 6
Comparisons with respect to dependability

Dependability	(1)	(2)	(3)	(4)	Local priority
(1) Maintainability	1	4	3	$\frac{1}{2}$	0.31
(2) Operating cost	$\frac{1}{4}$	1	$\frac{1}{2}$	$\frac{1}{3}$	0.08
(3) Local vendor	$\frac{1}{3}$	2	1	$\frac{1}{3}$	0.14
(4) Reliability	2	5	3	1	0.47
			$\lambda_{max} = 4.0634$		1.00

hierarchy, then the following holds:

$$w(i + 1) = U w(i), \quad i = 1, \dots, m. \quad (1)$$

In this particular case one has

$$w(2) = U w(1). \quad (2)$$

Recalling that the elements of U are the three first columns of Table 7, and that $w(i) = [0.09, 0.32, 0.59]$, then (2) yields the last column of Table 7, namely the global priorities of the parameters. Note, for instance, that the reliability parameter was judged to be of a relatively low priority (0.08) with respect to both setup and daily operation cost criteria, but of high priority (0.47) with respect to the dependability criteria. This shift of local priorities is demonstrated in Figure 4. The application of (2) yielded the global priority of the reliability parameter to be the highest (0.310) of all parameter groups in this decision problem. (Figure 5).

Before direct comparison of proposed systems with respect to their cost parameters is done, one has to prioritize single elements in each parameter groups. For example, the 'local vendor' parameter group having a global priority of 0.210 consists of four specific elements as shown in Figure 2. Their pairwise comparison and the derived priority vector are shown in Table 8. The global priorities of these four elements are obtained simply by multiplying each local priority with the global priority of the 'host' group (0.210).

Once all the parameters of the cost hierarchy are prioritized, the benefit hierarchy is dealt with

Table 7
Parameter priorities

Parameters	Criteria			Global priorities
	Set-up (0.09)	Daily operations (0.32)	Dependability (0.59)	
Maintability	0	0.24	0.31	0.260
Procurement cost	0.29	0	0	0.026
Operating cost	0	0.25	0.08	0.127
Local vendor	0.17	0.35	0.14	0.210
Reliability	0.08	0.08	0.47	0.310
Contractual clauses	0.15	0.08	0	0.039
Required customizing	0.31	0	0	0.028
				1.00

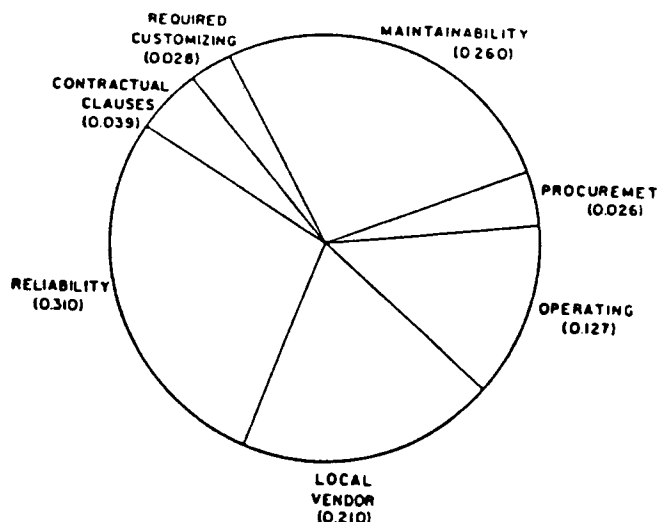


Figure 4. Local priorities of the cost parameters

by prioritizing its elements. The benefit hierarchy is constructed by selecting elements from Tables 1 and 2 that are relevant to the particular problem at hand, (cf. Figure 3). An example of a specific benefit hierarchy designed for an office automation problem can be found in [13]. The general structure of such a benefit hierarchy follows the same guidelines set for the cost hierarchy. That is, one starts with major criteria followed by sets of technical parameters and ending by vendor's proposals.

Once the benefit and the cost hierarchies are prioritized one enters the final phase of the analysis in which direct comparisons of proposals are made. Such comparisons rely on data that is either directly available through the vendors proposals (e.g., purchase price, lead time, or disk capacity) or has to be assessed by the decision maker. The comparative evaluation process will be highlighted through two comparisons. The first set of comparisons involves a group of *descriptive* parameters analyzed by the decision maker based on system's merits. Table 8 describes the global priorities of four elements that form a subset of the overall 'local vendor' parameter set in the cost hierarchy. In comparing vendors proposals with respect to each of these four parameters no 'hard numbers' can be used to describe the qualifications of vendor's proposals with respect to these parameters. It represents a case where the decision maker's judgement has to be employed. In comparing three proposals with respect to, say, 'service reputation'

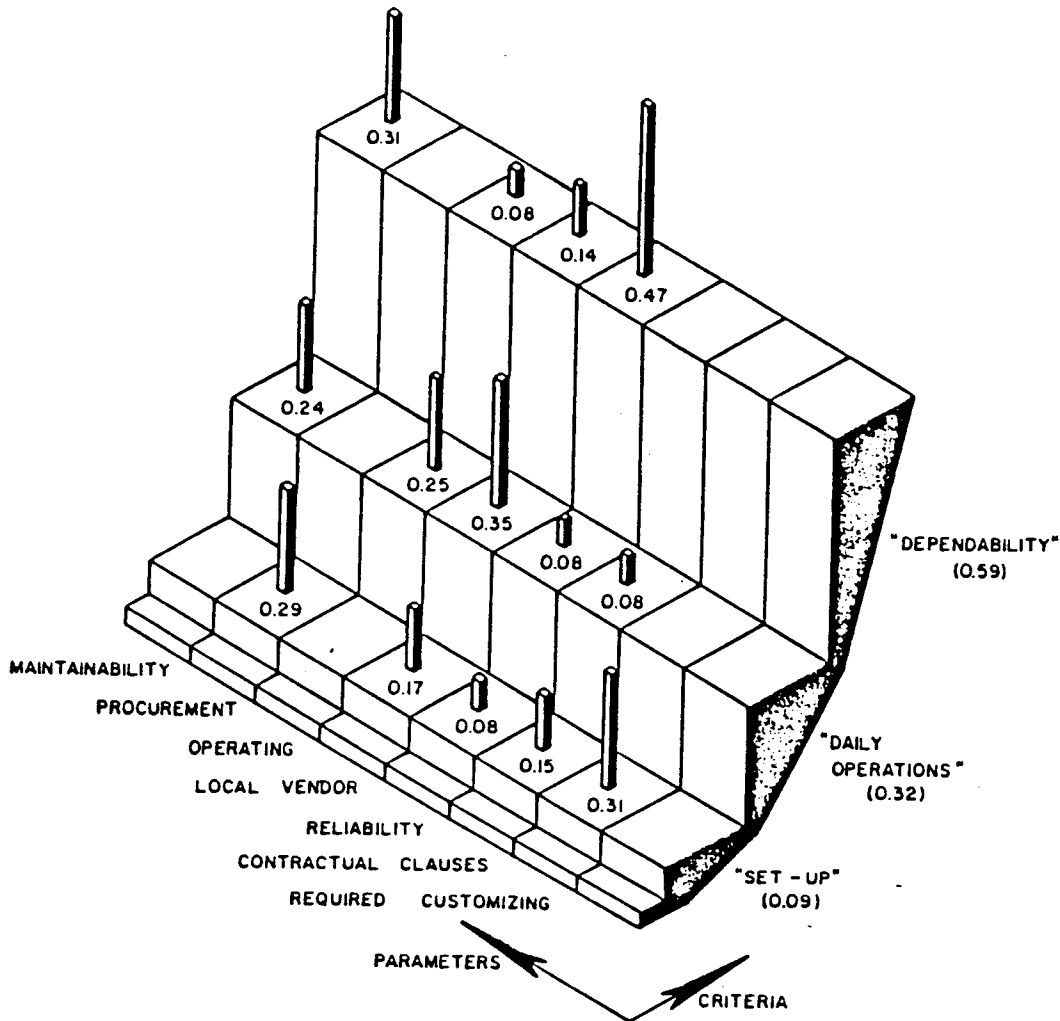


Figure 5. Global priorities of the cost parameters

one may arrive at the comparison shown in Table 9.

Examining Table 9 one observes that the service reputation of the vendor supplying System A was judged to be slightly better than that of System B. ($a_{12} = 2$); B slightly better than C ($a_{23} = 2$) and A moderately better than C ($a_{13} = 3$). This judge-

ment matrix represents the point of view taken by the decision maker in judging the proposals. As mentioned earlier these comparisons refer to data judged qualitatively. Other types of data, however, may require more elaborate efforts in determining the relative assessments. For example, vendors proposals describing microcomputer systems will

Table 8
Comparisons with respect to local vendor

Local vendor	(1)	(2)	(3)	(4)	Local priority	Global priority
(1) Past experience	1	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{4}$	0.10	0.021
(2) Service reputation	3	1	2	$\frac{1}{2}$	0.29	0.061
(3) Servicing ability	2	$\frac{1}{2}$	1	$\frac{1}{2}$	0.18	0.038
(4) Brand	4	2	2	1	0.43	0.090
	$\lambda_{max} = 4.0539$				1.00	0.210

Table 9
Service's comparisons

Service Reputation	A	B	C	Local priority
System A	1	2	3	0.54
System B	1	1	2	0.30
System C	1	1	1	0.16
	$\lambda_{max} = 3.0092$			1.00

invariably specify the type of microprocessor used in their particular installation (e.g., Z 8000, Intel 20860, or Motorola 68020). The listing of this type of information, coupled with such items as clock frequency, data bus width and other CPU parameters does not convey a clear measure of efficiency to the end user of such system. In order to develop such a measure, a special effort has to be undertaken that will examine hardware, firmware, and the software features in a real working simulation of the system.

This examination provides a composite measure (benchmark results) that is then used in assessing the suitability of the complete system in performing its tasks. Benchmark results have been used extensively in the last two decades in performance evaluation of mini and mainframe computer systems [15]. While the extensive effort required of benchmark studies in mini-and-mainframe computer is quite justified, their automatic extension to the microcomputer area should be limited in scope. This is so because of the difference in price and in the relative load of data processing activities. While a few man-months effort may be justified for mainframe benchmark studies, such an extensive effort may be economically unwarranted for a microcomputer acquisition. This does not mean, however, that benchmark studies should not be performed at all for microcomputers but rather, that they should be scaled down efforts. An illustration of this idea is provided next.

A commonly used, and easily implementable type of benchmark test is the one involving file sorting. This involves the sorting of sequential data files of various lengths in the main memory (RAM). When data files consists of randomly generated numbers, the sorting effort is directed at arranging these records in a pre-determined order. The benchmark study is directed here at determin-

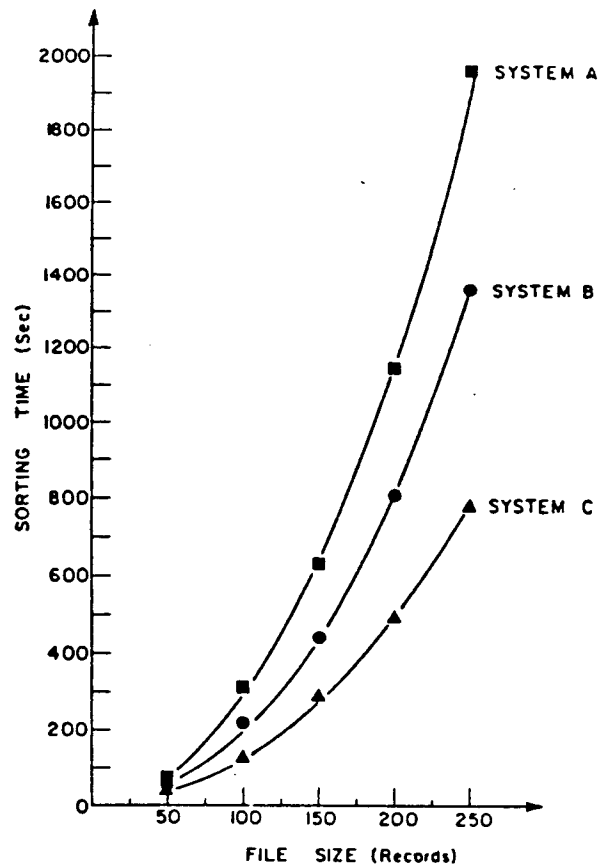


Figure 6. Benchmark results of three microcomputer systems

ing the time required by the microcomputer system to perform the sorting task.

Figure 6 depicts the results of such a benchmark study conducted for three commercially available microcomputer systems, running the same sorting program.

Each comparison (like the ones shown in Tables 8 and 9) results in an m -dimensional local priority vector, denoted by w_i , where i is the index related to the particular subject of comparison, $1 \leq i \leq n$, n is the number of technical parameter sets in the bottom level of the hierarchy and m is the number of vendor's proposals. Performing all these comparisons for all parameters will result in having n m -dimensional vectors. These vectors can be arranged as columns of an $m \times n$ matrix, and the global priorities of vendors' proposals are then obtained through

$$w = [w_1, w_2, \dots, w_n]v,$$

$$w, w_i \in \mathbb{R}^m, v \in \mathbb{R}^n, 1 \leq i \leq n,$$

where v is the global priority vector of all the technical parameters.

These global priorities for all the vendors proposals are derived for both the benefit and the cost hierarchies. It should be emphasized, however, that one seeks to choose a system whose benefit priority is the *largest* while at the same time its cost priority is the *lowest*. When such an ideal case is not available, cost-benefit methodology is applied to the cost and benefit priorities derived thus far [7]. This is done by arranging the alternatives in ascending order of cost priorities and then evaluating the *marginal* benefit to cost ratio. In the process one prefers a more costly alternative if its added benefit justifies the added cost.

6. Concluding remarks

The framework presented in this paper has been used in a number of real acquisition problems. Even though these problems were quite different in nature they all followed the same analysis states as described above. Each case had to identify its own decision elements out of a large set which is shared by all problems of this nature. This selection has led to the hierarchical models used at each particular application and whose prioritization was applied in evaluating and selecting the microcomputer systems. The approach proved to be flexible and efficient in the sense that it could accommodate various operating environments and outlooks; the final decision was facilitated by integrating opinions solicited from participants having different responsibility and technical expertise levels.

References

- [1] Borovits, I., and Ein-Dor, P., "Cost/utilization: A measure of system performance", *Communication of the ACM* 20 (3) (1977) 185-191.
- [2] Boucher, G.A., "Communications options for the integrated office", *Words* 10 (6) (1983) 35-37.
- [3] Edwards, C., *Developing Microcomputer Based Business Systems*, Prentice-Hall, London, 1982.
- [4] Ferrari, D., *Computer Systems Performance Evaluation* Prentice-Hall, Englewood Cliffs, NJ, 1978.
- [5] Frankenhuis, J.P., "How to get a good mini", *Harvard Business Review* 60 (3) (1982) 139-149.
- [6] Giffin, W.C., *Introduction to Operations Engineering*, Irwin, Homewood, IL, 1972.
- [7] King, J.L., and Schrems, E.L., "Cost-benefit analysis in information systems development and operation *ACM Computing Surveys* 10 (1) (1978) 19-34.
- [8] Kolence, K.W., *Introduction to Software Physics*, Institute for Software Engineering, Palo Alto, CA, 1976.
- [9] McFarlen, F.W., and McKenney J.L., "The information archipelago—Governing the new world", *Harvard Business Review* 61 (4) (1983) 91-99.
- [10] Morris, M.F., and Roth, P.F., *Computer Performance Evaluation*, Van Norstrand Reinhold, New York, 1982.
- [11] Saaty, T.L., *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1981.
- [12] Sauer, C., and Chandy, K.M., *Computer Systems Performance Modeling*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [13] Seidmann A., and Arbel, A., "An analytic approach for planning computerized office systems, *Omega* 11 (6) (1983) 607-617.
- [14] Walsh, M.E., *Understanding Computers*, Wiley, New York, 1981.
- [15] Wyrick, T.F., "Benchmarking distributed systems: Objectives and techniques", in: D. Ferrari (ed.), *Performance of Computer Installations*, North-Holland, Amsterdam, 1978, pp. 83-101.