International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

On the Optimal Decentralization of Data Processing in an Organization

Golovanov, S.

IIASA Working Paper

WP-79-120

December 1979

Golovanov, S. (1979) On the Optimal Decentralization of Data Processing in an Organization. IIASA Working Paper. WP-79-120 Copyright © 1979 by the author(s). http://pure.iiasa.ac.at/1063/

NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

ON THE OPTIMAL DECENTRALIZATION OF DATA PROCESSING IN AN ORGANIZATION

Sergei Golovanov

December 1979
WP-79-120

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

PREFACE

Sergei Golovanov participated in IIASA Junior Scientist Program three months during the summer 1979. This paper is one of the results of his stay at IIASA where he joined the Management and Technology Research Area. He was associated with the study of the impact of small scale computers on managerial tasks.

Decreasing computer hardware costs and increased networking possibilities mean that the system design will be based on a different balance between cost elements than has been traditional. New cost elements, hitherto secondary will play a more decisive role. Among those are different aspects of organizational consequences of computerization and its impact on individual, group and organizational behavior. Sergei Golovanov has designed a model by which the impact on computer system architecture of different weight factors for a spectrum of cost elements can be studied. The prerequisite of his study was a three-months assignment only. It is therefore quite obvious that he could not take into account more than a very limited set of cost elements in this study. It was also not possible to include a quantitative analysis of different design alternatives. It is however valuable to make his approach and methodology available in this working paper as a contribution to further work in this field.

Laxenburg, December 1979
Goran Fick

SUMMARY

Due to the developments in micro electronics, there is today a wider choice than ever of computer sizes and networks to choose from. There are many factors like investments, installation, system development, education, organizational consequences and many others to be taken into account when an organization faces such a decision. A model of a set of those costs has been developed along with a computer program to carry out the tedious numerical work to compare different system design alternatives. The report does not include quantitative simulations but focuses on qualitative aspects. It is shown that in contrast to the earliest computer technologies, there exists now a true minimum representing the optimal level of decentralization. For a given task distribution, this level is shifting in time towards greater decentralization.

CONTENTS

1. INTRODUCTION 1

2. APPROACH 3

3. MODEL 5

  3.1 Invested Costs 6
  3.2 Operational Costs 9
  3.3 Development and Dynamics Costs 15
  3.4 Optimum Decentralization Search Algorithm 19

4. CONCLUSIONS 20

Figures 24

REFERENCES 34

On the Optimal Decentralization of Data Processing in an Organization

Sergei Golovanov

1. INTRODUCTION

The rapid evolution of administrative computer applications caused on the one hand by the increasing pressure of the external environment on organizations and on the other by the nature of management in modern society, has attracted a lot of attention from managers, operational researchers and system analysts.

A better understanding of the impact that information from the external environment and internal flows have on the efficiency and stability of organizational operation has caused increasing attention to be paid to new electronic data processing opprotunities.

The introduction of a Data Processing System (DPS) into an organization produces many benefits [2,3]:

(1) improvement in operation and in the functions performed;

(2) greater flexibility and adaptability of an organization in a relatively quickly changeable external environment;

(3) better resource management and savings in some important resources consumed by an organization during an operational process.

These improvements in organization performance are the main goals in the creation of a new organizational subsystem such as DPS in an organization. At the same time this event influences all internal interconnections between other existing subsystems and causes various impacts on the operation and efficiency of the organization

This topic has been analyzed by many researchers [2-14], who have studied the impact of introducing a DPS into an organization from various points of view:

- changes in tasks and operation at every level of the organization structure from top and middle level management (management information and decision support systems, etc.) to the lower level of workers (computer-aided design, inventory and banking systems, automated office, etc.);
- changes in the function carried out by an organization (e.g., R&D, manufacturing, banking, etc.);
- a shift to more formalized creative and specialized applications and style of work;
- problem of personnel and staff adaptation to the DPS and vice versa;
- changes in the way of thinking and creating new skills;
- problems of the man-computer interface in a given application; what is easy and favorable for the user, etc.

The theoretical and empirical results obtained in these studies show that the outcome of these aspects depends very much on the approach to the system design of the DPS. Due to the rapid evolution of computer and communication technology the design must be based on very recent technological knowledge.

Recent and continuing advances in all classes of computers--from large mainframes to small microcomputers--provide many alternatives in the introduction of a DPS in an organization. Each of these alternatives influence expected benefits in different ways. One of the most important questions that arises when people try to use this tool is, "what is the best DPS alternative for a given time and state of technology and what are the consequences and trade-offs that we may expect for an organization in a limited future time period with the various alternative DPS's?"

When answering this question it is necessary to take into consideration not only the costs of the various DPS alternatives (hardware, software) but also the other aspects of costs which reflect real life expenditures.

It is particularly interesting from this point of view to consider the dynamics of DPS evolution following organizational needs in data processing. Despite inherent difficulties in obtaining some parameters of the evolution process that are rather difficult to measure as well as predicting a particular trend in the evolution, attempts to find out some answers to the question raised may provide a better understanding of the real benefits and drawbacks and provide an accurate way to proceed in the development of these systems.

2. APPROACH

When we consider very schematically the operation of an organization (as illustrated in Figure 1), in general we find that there are only two possible ways to increase the efficiency of an organization:

(1) to increase output from the organization or functions performed and consumed by the external environment;

(2) to reallocate resources from the external environment that are consumed by an organization in its operational process.

Of course it is also possible to use a combination of these two methods.

Let us assume that after some system study of an organization it has become clear that the efficiency of this organization in its classical definition (output-to-input ratio) may be improved by introducing some Data Processing System as a new subsystem, within the organization.

We may also suppose that during this study the activity of every member of the staff has been analyzed and that the tasks which could benefit from some automation have been identified (in the scale from slightly computerized support to full scale automation). Using this approach we only consider automation of already existing tasks. In real life computerization is often a reason to redesign the tasks to satisfy a wider set of criteria.

The level of possible automation is illustrated as an example in Figure 2.

In the vast majority of cases, computerization of tasks allows personnel to spend less time performing these tasks [11-13], or, in other words, to increase productivity. In this case we may see possible savings as represented in Figure 3.

The overall success of introducing more data processing of course depends on many other factors and costs such as computer expenditures, system development costs, job satisfaction, educational skills, system flexibility, etc.

Leaving aside the questions of which tasks could be automated, and to what extent, we suppose that these savings of time can be translated into saved salary expenditures by the following expression:

$$SV_i = \sum_{j=1}^{k} E_{xi} \cdot (T_{ij} - T'_{ij}) \qquad (1)$$

where $\{i\}$: $i = 1, N$ as the index of the person in organization,

N: total number of staff in organization,

$E_{xi} = \frac{Sl_i + I_{ci}}{T_w}$: average cost of given person's activity per time unit T_w,

Sl_i: mean salary of the given person per month,

I_{ci}: indirect expenditures for the person's activity in given organization per month,

T_{ij}: time necessary to perform this specific type of task with number j without DPS

T'_{ij}: time to make the j-type of job with assistance of DPS,

k: total quantity of tasks, performed by i-th person.

It is worth noticing here that the greatest benefits per capita can be achieved from the automation of the tasks (done by those members of staff who have with) the highest cost-per-time unit ratio or E_x-ratio, e.g., top managers and other qualified personnel.

These salary savings may be gained directly by increasing the level of output from a given person (if such a demand exists) and systematically going through the whole organizational structure or (if possible due to other constraints) by reducing staff to a level necessary to perform the original level of output.

Generally we may say that in both cases it is worth introducing a DPS into an organization if the expenditure can be covered by savings (in a broad sence), direct and indirect, within some limited time span.

Bearing in mind the idea of creating a simple situation for comparing the various alternative of DPS's, we suppose that, in the study of the organization mentioned above, analysts have defined the range of possible savings and conditions necessary for obtaining a level of efficiency not worse than before. The only question left in this case was, "What are the optimal key parameters of DPS and their values at a given moment of time while having some assumption about supposed dynamics of organizational evolution as well as computer technology products?"

The model described here provides a tool for a better understanding of the subject while searching for the answer to this question.

3. MODEL

The main purpose of the model is to attempt to define some key expenditures or costs involved in establishing a DPS in an organization and to correlate these costs to the given number of tasks peculiar to a given organization, making this method a tool for cost optimization. This approach provides the opportunity to calculate relative costs per single task (or job). In addition it is interesting to study some indirect costs caused by differences in the various characteristics of DPS alternatives and their influence on the operations in an organization. In the following analysis, we will consider that every DPS alternative may consist of some data processing units (DPU).

Usually there are three ways to installing a DPU into an organization: to purchase; to rent; or to rent and then purchase later [1]. To simplify the analysis we shall consider primarily the first option, mentioning however the cases where the second option may be applied. It is worth noting here that all the factors considered in any case could refer to the second option, but they must be analyzed from point of view of the DPU manufacturer, because all these costs come into the rent cost, offered then to the end users. Generally, the cost of introducing a DPS into an organization may be described as follows (Figure 4):

(1) invested costs;
(2) operational costs;
(3) development costs.

We shall consider (1)-(3) from the DPU performance point of view. The word "performance" integrates in itself may of the parameters and characteristics of a DPU. However, in this model we look at DPU alternatives as a set of "black boxes" available on the market and thus performance P refers only to classes of DPU alternatives. When thiis approach is used the different P_{i5} will be made to refer to specific computer system alternatives.

3.1 Invested Costs

Invested costs, usually paid once, may be decomposed into several subgroups:

(1) hardware and system software cost $C_{i1}(P)$;
(2) cost of installation $C_{i2}(P)$, the amount of which appears to be a rather large percentage compared to C_{i1} and is usually expressed as $C_{i2}(P) = a_1 \cdot C_{i1}(P)$, where a_i is a coefficient;
(3) cost of building and reconstruction of the room for DPS: $C_{i3}(P) = a_2 \cdot C_{i1}(P)$;

(4) cost of spare parts $C_{iy}(P)$, which should be available at local stock, usually as some percentage of C_{i1}, $C_{iy}(P) = a_4 \cdot C_{i1}(P)$;

(5) costs of maintenance equipment (testing, diagnostic, etc.), purchased simultaneously with DPU and also correlated to the C_{i1}, $C_{i5} = a_5 \cdot C_{i1}$.

Therefore the invested costs considered above

$$C_{IN} = \sum_{j=1}^{5} C_{Ij} = C_{ij} \cdot \sum_{j=1}^{5} a_j \quad .$$

Some indirect costs may be analyzed here too. Considering the existing trends in computer evolution, it is easy to show that since the first appearance of DPUs, the cost of hardware has permanently decreased. Many curves looking like the one described in Figure 5 have been referred to and discussed in different studies [4,7].

If we consider the period of time for DPU delivery we learn that the more complex and sophisticated a system is the more time it takes between ordering it and putting it into operation. In the worse cases, for example, for very large mainframes, the time gap between these two events was about a year or a year and a half. As a good illustration here, a comparison should be made between the time and cost necessary to assemble, transport, mount, install and test a large mainframe system (which also requires double floors, special power supply, airconditioning, etc.) and that necessary for a system consisting of several interconnected microcomputers. The reduced complexity of microcomputers will result in a shorter time delay only if the additional complexity added by the networking aspects will be relatively low. With the present advanced state of networking technology this is a probable assumption.

Taking into consideration also the time required for information about new commercially available products to become known to the end users (collecting the information, comparing the information from different sources, making benchmarks, etc.) before they make the order, we get a certain delay time which for some computers can be equal to one-third or one-half of its rather short economic life. Bearing in mind the rate at which new systems are developed we can suppose that at the time a system becomes operational a new DPU should appear on the market with the same level of performance but for a lower price. At Figure 5 situation is represented by the ideal or maximum possible losses ΔC, which in real life of course are less because for a new system there also exists a similar time delay. This delay however is expected to become loss significant due to the progress in technology, etc. one could see that in this case a more favorable situation is for microcomputers, for which it takes much less time to become operational:

$$\Delta t_1 < \Delta t_2 < \Delta t_3 \quad .$$

In this way the cost of a DPU may be approximated by some time- and performance-dependent function from statistical data in references and case studies as

$$C_{i1} = f(t,P) \tag{2}$$

and

$$\Delta C_1 = f(T_{or}, P_i) - f(T_{or} + \Delta t_i, P_i) \tag{3}$$

where i: index of performance class P

Δt: delay between the order of a system and its full scale operations.

ΔC_1: indirect losses caused by Δt.

To make indirect costs explicit for future comparisons of the various DPS alternatives we make the following artificial expression

$$C_{IN} = C_1 + C_{ij} \cdot \sum_{j=1}^{5} a_{Ij} \tag{4}$$

In the case of rent, indirect losses C_i still exist; however, other invested costs besides C_3, transfer to the rent cost.

3.2 Operational Costs

The main parts of the operational costs are (Figure 6):

(1) maintenance of hardware and system software;
(2) resources consumed in operation;
(3) cost of unreliability.

Let us consider the components of each of these parts, usually described as monthly costs.

3.2.1 Maintenance costs

Day-to-day maintenance costs consist, in the purchase case, of (Figure 7):

(1) Cost of hardware and software personnel salaries and corresponding indirect expenditures;

$$C_{01} = N_{HW}(P) \cdot E_{xHW} + N_{SW}(P) \cdot E_{xSW} \qquad (5)$$

where $N_{HW}(P)$: number of hardware maintenance personnel required for given DPU with performance P,

$N_{SW}(P)$: corresponding number of software personnel;

(2) cost of renewing the testing and repairing equipment C_{02} which may be described as a variable dependent on cost of DPS hardware C_{i1}

$$C_{02} = a_6 \cdot C_{i1} \qquad (6)$$

(3) cost of computer time spent for maintenance purposes C_{03}.

The last item (3) may need some additional explanation. In some cases computer time spent for maintenance takes a relatively large percentage of the total computer time resource available for end users. The total expenditure for running a DPS can be correlated with the computer's time when we consider it as a resource. This correlation can be used for cost estimations of computer time spent for maintenance and software purposes.

These estimations, however, can only be made at the second iteration after all the costs have been calculated.

3.2.2 Educational Costs

Another important cost factor which usually does not draw much attention in other cost analyses is educational costs.

We try to make this more explicit in the following way.

Let us consider the steady working process of a person in an organization. Assuming that the level of his qualifications and knowledge is approximately equal to the level of his functions, we may measure his activity in cost as salary S_i plus indirect expenditures I_i. When this person leaves our organization for some reason, a new-comer arrives.

Whatever qualification he has, he should follow some education and learning process to gain knowledge and understanding for the necessary specific course of action.

In this sense we have some transitional process and within time period of this process the output of the person is not equal to the cost paid for his activity

$$(S_i + I_i) = E_{xi} \qquad (7)$$

Meanwhile, some results achieved in the study of the mental and intellectual activities of man [17] shows that the speed at which new knowledge and information is gained is presumed to be exponential (Figure 8). In simple words this may be expressed as, the more a person has to remember and know the less quick this process is.

In our case we would suppose that a new-comer gains the qualifications necessary for his job exponentially in time, beginning from his original level of qualification Q_0 (Figure 10).

Analytically it may be formulated as follows:

$$Q(t) = Q - q_0 \cdot e^{-\eta t}$$

where $Q(t)$: current level of qualification

q_0: original level of new-comer's qualification measured in the range [0,Q]

η: rate of qualification gain.

Using our assumption about the measure of qualifications as (7), we determine the cost of education as the area of the triangle between a constant level of E_x and a variable level of output j for a given person (Figure 9).

The original level of qualification q_0 may be measured in scale $(0,E_x)$.

The area of the triangle in Figure 9 is in this case the following expression

$$C_e(t) = E_x \cdot t - \int_0^t (E_x - E_x(1 - q_0) \cdot e^{-\eta t} dt$$

which reduces to

$$C_e = \frac{E_x(1 - q_0)}{\eta} (e^{-\eta t} - 1)$$

and finally, when t → ∞

$$C_e = \frac{E_x(1 - q_0)}{\mu} \tag{8}$$

where q_0 is within the range $(0,1)$.

The value of q_0 is very much dependent on the given external labor environment and in general is a stochastic parameter with a certain distribution. For our purposes we may limit our consideration of this model taking some mean value of q_0 specific for the external environment.

Most importantly for this consideration is an assumption that η depends on the complexity of the DPU, its architecture, and the system and application software. The more sophisticated a DPU is, the more time one needs to learn it (Figure 10).

This assumption reflects real life but does not eliminate certain difficulties in the measurements.

However, some case studies and analysis available from references provide some ground for such an approach [17,18].

Now we may direct our attention to the personnel handling maintenance and system software.

Let N_{HW} and N_{SW} be the numbers of maintenance hardware and software personnel, and δ_{HW} and δ_{SW} be the rates of arrival of new-comers per person into hardware and software groups respectively. We then set up the following formulae for the educational costs of both groups:

$$C_{eHW} = N_{HW}(P) \cdot \delta_{HW} \cdot E_{xHW} \cdot \frac{1 - q_{0HW}}{\eta_{HW}(P)} \tag{9}$$

$$C_{eSW} = N_{SW}(P) \cdot \delta_{SW} \cdot E_{xSW} \cdot \frac{1 - q_{0SW}}{\eta_{SW}(P)} \tag{10}$$

This way, educational costs for maintenance are as follows:

$$C_{04} = C_{eHW} + C_{eSW} \tag{11}$$

3.2.3 Resource Consuming Costs

For the cost of material resources consumed by DPS while running, we may limit our analysis to two main categories--spare parts C_{05} and power supply $C_{06}(P)$. However in a more detailed analysis it is possible to include other resources (paper, etc.). The cost of spare parts depends on the level of reliability; this will be considered later. For our purposes we assume that $C_{05} = \sigma_{HW}(P) \cdot a_7 \cdot C_{i1}$, where $\sigma_{HW}(P)$ is the average fault rate for hardware. The cost of the power supply is obviously a function of given hardware power consumption E, working time t and power prices g: $C_{06} = g \cdot E(P) \cdot t$.

3.2.4 Unreliability Costs

One of the most complex cost characteristics is the cost paid for the non-100% reliability of a DPS. We can define the price paid for some unreliability as a loss caused by interruption or delay of personnel activities which integrates later into the main organizational output. It is rather difficult to evaluate the combined cost of delays and interruptions related to functions consumed from the given organization by the external environment. Because of this, we can express these losses as money spent on personnel salaries and indirect expenditures while the data processing service is not operating. This situation is usually caused by some rate at which faults occur in hardware and software as well as specific situations, such as the interdependent task flows, processed simultaneously at the same resource (lockouts, deadlocks, quiescings, hang-ups, etc.).

Assuming that certain data about the reliability of the different DPS alternatives can be learned from an analysis of statistics or in some other way, we may introduce such aggregated variables as

$\sigma_{HW}(P)$: hardware rate of faults per month, which periodically occur in some type of DPU;

$T_{HW}(P)$: average non-scheduled maintenance time, necessary to recover the hardware;

$\sigma_{SW}(P)$: software rate of faults per month,

$T_{SW}(P)$: average non-scheduled software maintenance time, necessary to recover the system software and make it run.

$\sigma_{int}(P)$: faults, caused by interference of different kinds of tasks, running simultaneously at the same processing resource.

$T_{int}(P)$: average time to recover the system.

The third group of faults should need more explanation. Various studies, appearing during the last 10 years, have shown that when there is an increasing quantity of job streams, performed at the same processing resources (cpu, memory, channels, etc.), there is more probability of faults from deadlocks, etc.

Another reason for this separation of the interference factors from software faults is the increasing probability of coming across a "bug" in complex system software, which has been unknown only because this particular multi-stream job situation has not occurred before.

So far the main topic that is underlined by the introduction of the third group of reliability variables is that the wider the range of simultaneously processed task is, the more often can faults caused by interference of these tasks occur and, conversely, the more homogenous a processed task flow is the less probable these kinds of faults are.

Reflecting the fact that when any of these failures occur it influences or stops (e.g., in the case of the highly automated activity of personnel) the working process of those members of staff who use the system, we may roughly evaluate the price paid for unreliability as follows.

For centralized system:

$$C_{05} = T\cdot(\sigma_{HW}\cdot T_{HW} + \sigma_{SW}\cdot T_{SW} + \sigma_{int}\, T_{int})\cdot \sum_{i=1}^{n} E_{xi} \qquad (12)$$

and for decentralized system with personal dapa processing units:

$$C_{05} = T \sum_{i=1}^{n} (\sigma_{HWi}\cdot T_{HWi} + \sigma_{SWi}\cdot T_{SWi}) E_{xi} \qquad (13)$$

where T is considered period of DPS economic life (years);

n is mean number of staff using DPS in their work simulnateously.

N is total number of personnel in organization;

$E_{xi} = \dfrac{S_i + I_i}{T_m}$: the relative cost of i-th person working time per working month T_m;

S_i is salary of i-th person;

I_i is indirect expenditures for his activity.

The absence of interference variables in the decentralized case may be explained as $T_{int} = 0$ owing to practically full homonegeity of tasks flows generated by a person who uses a data processing unit without sharing it with other users.

In the centralized case the system is used by n working members of staff simultaneously and all the service interruptions cause parallel stops and delays in their working activity.

3.3 Development and Dynamics Costs

It appears that there is widespread understanding of the fact that introducing a DPS into an organization awakes the creative abilities of personnel [2,12,13] and causes a constantly increasing number of tasks performed with DPS assistance.

Users who have become acquainted with the system and had good experience usually increase not only the density of the task flows but also the numbers in the task environment, generating more and more new tasks while automating their everyday activity.

In its turn these circumstances create a highly dynamic load-to-process with time span (Figure 11).

To understand this problem it is necessary to consider the attributes of load generated in organization.

3.3.1 Model of the Load

Considering the question of the load to be processed at a DPS and generated in an organization, we assume that the organization has a hierarchical structure (Figure 12) or that it can be brought to the latter.

Every element of this organizational structure (DP Department is considered separately) is assumed to use a DPS in its activity. One may enumerate all the tasks which are executed with DPS assistance.

Let us suppose that a given person i in an organization structure which has N members $(i = I,N)$ generates jobs in each class of tasks with density $\{\overline{\lambda_i}\}$, $j = \overline{1,k_i}$, where k_i is the total number of tasks solved by the i-th member of staff. For λ measurements we may choose any appropriate scale-number of j-th tasks per hour, day, week, etc. Examples of the tasks within the framework of the given analysis are as follows: text editing, internal communication and electronic mail within an organization, personal and common file and data base handling, information updating, decision support and management information, etc.

For every task from set $\{k_i\}$, $i = \overline{1,N}$, someone or the person himself can formulate a certain set of requirements and constraints, e.g., approximate predictable task increase rate $\lambda_i = f(t)$, some months or years in advance (Figure 13): time response or turnaround time necessary to implement the task--t_{ij}, the necessary accuracy A_j in executing the task and results; security constraints; if the task should be processed locally at his working place or internally in an organization or external data processing allowed, etc.

Passing round all the elements in the branches of a hierarchcal structure we can in this way define the general requirements of a DPS to be met by any of the DPS alternatives.

The total flow of tasks to be processed

$$\Lambda(t) = \sum_{i=1}^{N} \sum_{j=1}^{K} \lambda_{ij}(t) \qquad (14)$$

where K is the total set of tasks, $K = \bigcup_{i=1}^{N} k_i$ (15)

and N is the total number of personnel in the organization. The task flows obtained reflect to some extent the functional specialization characteristic of the given organization.

What are the DPS alternatives to meet these requirements?

3.3.2 Data Processing System Alternatives

Managers who are responsible for making a decision about introducing a DPS into an organization have to choose an appropriate system among those available on the market.

Every time they do this they have to analyze a huge number of hardware and software characteristics, e.g., cpu cycle time, memory size and cycle and access time, buts speed, disk capacity and access time, etc. All these data do not necessarily represent information about the real processing capabilities of the system to the specific load requirements of the given organization and often only increase the probability of making the wrong decision [2,16].

In addition to this, different operating systems and software packages running with the same hardware produce quite different processing capabilities and applications of the computer system as a whole.

The absence of a unified theoretical approach in this field, and which is unlikely to be developed in the near future, gives no opportunity to make a reliable connection between organizational processing needs and hardware/software characteristics [3,16].

To find some way to handle this problem we can make the rather realistic assumption that every computer system, as a data processing unit (DPU), can be measured to obtain data about processing capabilities in every task class among those from K characteristic for our organization.

Using queuing theory notation, we suppose that it is possible to obtain service rates $\{\mu_i\}$, $i = \overline{1,K}$, of every DPU, among a finite set of computers available on the market at a given moment in time.

One of the possible ways to obtain these data is to use some computer-based task generator to produce the stream of jobs necessary for the measurements, as shown in Figure 14.

However, the question of the measurements and approach toward it needs special consideration. We can notice that in general, when such tools as an emulation and a portable application software are used, there are not many obsticles to this approach.

No other constraints are imposed on these measurements and the set of DPU's, except the homogenuity of the task stream generated. In the set of DPU's, for example, we may include random subsets of computer families varying in memory size and options, external memory devices, different software packages, etc.

In this case, we can get the information about DPU performance that represents real performance characteristics of the system as a whole--hardware plus software. It makes this information independent of any subjective judgments and methods, as it is in the benchmarking.

After this measurement, made on a finite set of m data processing units, among which we of course included all commercially available microcomputers, we would get a matrix

$$M = \{\mu_{ij}\},\ i = \overline{1,M};\ j = \overline{1,K} \qquad (16)$$

The matric $M = \mu_{ij}$ provides almost all the necessary information about data processing technology for the given field of tasks K that exists at the moment.

Some additional data may be taken into consideration, which are concerned with the problem of overheads and interference of tasks.

It is well-known that the majority of present-day computers run software aimed at a wide range of applications and use hardware cpu resource for user tasks as well as for system software. The great difference in the various task requirements--and particularly time response--make it necessary to create complex dynamic scheduling algorithms, priority systems, etc., which deteriorate the level of resource utilization.

In contrast, unfavorable task interference, e.g., the well-known simultaneous calculation of large scientific jobs and data base operations, which require a high level of Input/ Output operations, cases a considerable increase of overheads and, as a result, delays in time response and decreasing service rates for all executed tasks (thrashing, swapping, etc.).

There were several indications, e.g., [2,3] have shown that for large mainframes overheads may vary from 20% to 60% of the total computer system utilization.

Of course this situation comes into conflict with the main goal of DPS in organizations--to provide the necessary processing facilities which satisfy all the load requirements at minimum cost per task calculated.

We may try to take overhead and interference into account by introducing an overhead function that is dependent on the numbers of simultaneously processed tasks (Figure 15).

$$M_{ij} = f_{ij}, \; e = i,K \tag{17}$$

This information may be also obtained from the same type of measurements as described in Figure 14.

It should be noted here that, when we consider the personal DPU case, all the load generated consists of the sequence of different tasks--everyone who works at the microcomputer generates only one type of task flow at any moment. In this case we would have overheads and interdependences at minimum levels.

3.4 Optimum Decentralization Search Algorithm

Some studies of DPS developments and evolution made in organizations with very active DPS use have shown that a process towards decentralization of DPS exists [8,11]. Different departments try to establish their own DPS. In the studies mentioned we can also find explanations describing the different driving forces of the process. In our approach, however, we suppose that there is a decentralization optimum, which depends on a given state of DP technology $\{\mu_{ij}\}$, an organizational structure and a set of tasks Λ. What does the approach consist of?

We vary the level of decentralization, -D, which we define as a number of DPUs within a DPS.

We make these variations from the level where every member of staff has his own DPU, to the level where all the data processing is carried out by one central DPU. In all cases, DPS should satisfy all the load requirements Λ and the solution for every DPS alternative should be found from a finite set of available DPUs. We describe all these data here, as input to the model.

(1) Load requirements:

$$\Lambda(t) = \{\lambda_{ij}(t)\}; \quad i = \overline{1,N}; \quad j = \overline{1,K} \qquad (18)$$

where N is the total number of personnel in an organization and K is the total number of task classes.

(2) Task classes constraints:

-- response time $T_r \leq T^*_r$,

$$T_r = \{t_{rij}\}; \quad T^*_r = \{t^*_{rij}\}; \quad i = \overline{1,N}; \quad j = \overline{1,K} \qquad (19)$$

-- security $S \leq S^*$,

$$S = \quad s_{ij} \ ; \quad S^* = \quad s^*_{ij} \ ; \quad i = \overline{1,N}; \quad j = \overline{1,K} \qquad (20)$$

S_{ij} and S^*_{ij} may vary in the integer set of sequrity levels $S_{ij} = \overline{0,4}$; $S_{ij} = 0$ is the permission to process the task j from a member of staff i at external data processing resources; $S_{ij} = 1$ -- within an organization; $S_{ij} = 2$ --e.g., at the level of a department, etc. When $S_{ij} = L$ --the tasks must be processed locally.

-- accuracy $A \geq A^*$,

$$A = \{a_{ij}\}; \quad A^* = \{a^*_{ij}\}; \quad i = \overline{1,N}; \quad j = 1,K \qquad (21)$$

several other task constraints may be added if necessary.

(3) Set of DPUs (or computers) available on the market M and their operational characteristics:

$$M = \{\mu_{ij}\}; \quad i = \overline{1,M}; \quad j = \overline{1,K} \qquad (22)$$

(4) Vector of economic life costs inherent to every DPU:

$$V = \{v_i\}; \quad i = \overline{1,M} \qquad (23)$$

$$v_i = \frac{C_{INi} + C_{Ii} + T_{ei}C_{opi}}{T_{ei}} \qquad (24)$$

where C_{INi} is the installation cost (4);
ΔC_{Ii} are the timing losses (3);
C_{opi} is the operational cost;
T_{ei} is the period of DPU economic life.

We note here that some operational cost parameters, e.g., C_{03} and C_{05}, may not be put into expression (24), because of difficulties in its definition at this stage, or may be taken as a certain percentage of other operational costs.

In the case of rent,

$$v_i = r_i + \frac{C_I + C_I + (C_{01} + C_{esw_i} + C_{05})\ T_e}{T_e} \qquad (25)$$

where r_i is rent cost, and into C_{01} should be taken only that part which concerns software personnel salaries.

These data form the initial input into the optimal decentralezation search algorithm shown in Figure 16.

The goal function of the B1 block is to go round all sources of tasks at a given level of orgstructure and tries to find those DPUs which have the maximum number of tasks which requirements can be met at a minimum cost v_s.

At the lowest level, the search is made among a subset of M DPU's, consisted only of microcomputers. In this case we possibly does not find a DPu that meet all the requirements for all the tasks; however, by sieving a maximum number of them we eliminate the load for the next level of processing (Figure 17).

For the load- and time-response requirements analysis we may use any model which reflects the processing characteristic of the DPU, e.g., M/M/1 queueing model, which has considerable accuracy for our purposes (Figure 18). Comparing the different DPUs, we consider task flows homogenous if it is a personal processing unit, otherwise we use function (17) of service rate dependency on the number of simultaneously processed tasks.

In both cases, the level of DPU utilization should not exceed a certain level, e.g.,

$$\sum_{j=1}^{k} \rho_j \leq \rho^* = 0.75,$$

and all theother constraints must be met.

A more detailed description of the algorithm is given in Figure 19.

By varying the initial levels of the organizational structure and correspondingly the structure of the load we can get at the output of the model described and the changing values of DPS cost: $C_s = F(D)$.

The behavior of this function is shown in Figure 20. It can be seen that an optimum level of decentralization exists for some given level of load and processing technology. If we use data about the processing characteristics of the computers in the past for the comparison, we presumably may expect the variation of the decentralization optimum as illustrated in Figure 21. In the early years when only large tube-mainframes were used no cost-efficient optimum existed.

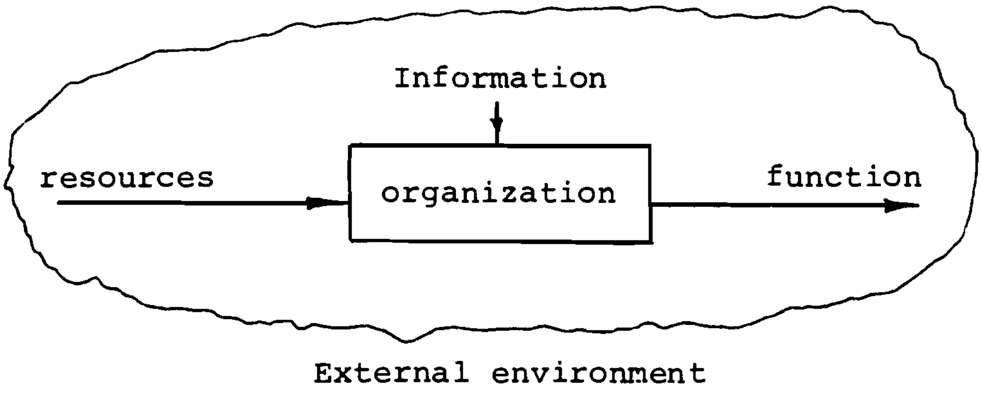
A permanent decrease in hardware and software costs and improved data-processing characteristics have chanbed this situation. It can be seen that the optimum level of decentralization D* is a function of time.

4. CONCLUSIONS

A certain deficiency of centralized data-processing systems, which appears in the decentralization trends throughout various organizations, has caused not only partial dissatisfaction of some personnel (impersonality, unreliability, lack of flexibility, etc), but also has some economic grounds.

The rather rough analysis made in this paper allows us to determine the economies of introducing a decentralized DPS into an organization. Several additional factors can be induced in the model when a more thorough study can be provided, e.g., scaling factors when the simultaneous purchase of a large quantity of microcomputers causes a reduction of its prices; psychological factors of introducing DPSs of different structures into an organization (problems of the man-computer interface); various other ways of introducing DPSs into an organization--external data processing resources and problems of optimal functional specialization of DPU's in a local network.

However, bearing in mind the desire to reflect main cost parameters, we may also include in the model certain memory requirements.



1. Improved operation and functions
2. Better resource management
3. Consumed resourse reallocation & savings
4. Greater flexibility and adaptability toward changeable environment.

Figure 1. Possible positive consequences caused by introducing DPS in an organization

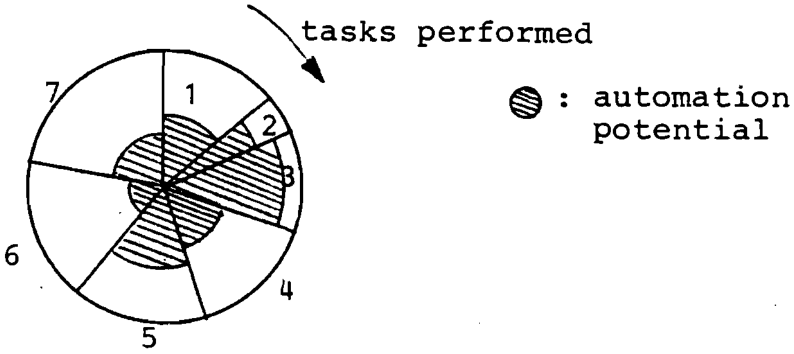


Figure 2. Analysis of' Possible Computer Assistance in Everyday Tasks Handling

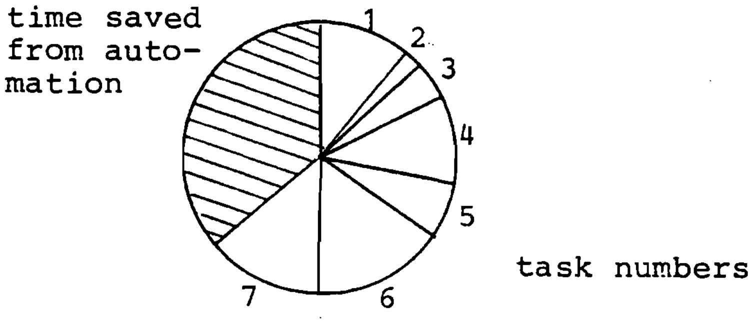


Figure 3. Time Savings from Possible Automation

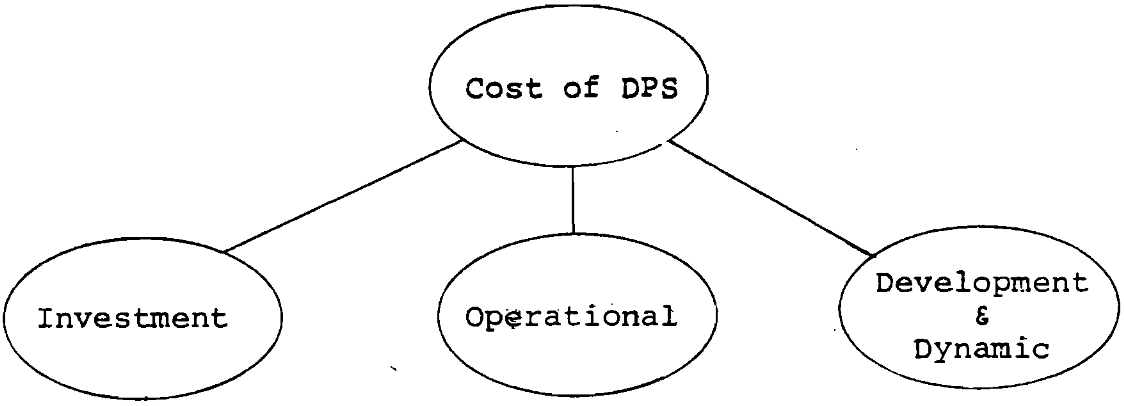


Figure 4. Costs Attributes

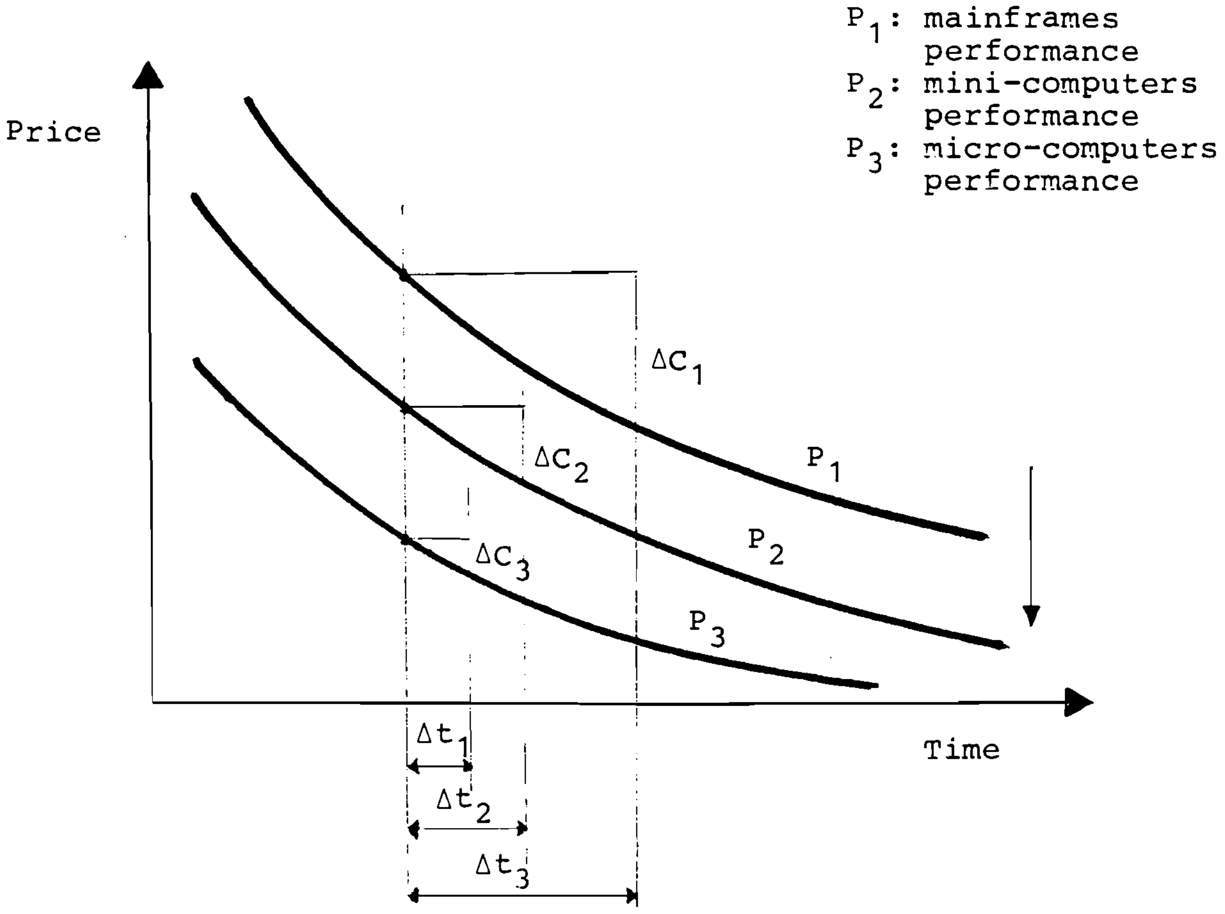


Figure 5. Trends in hardware costs

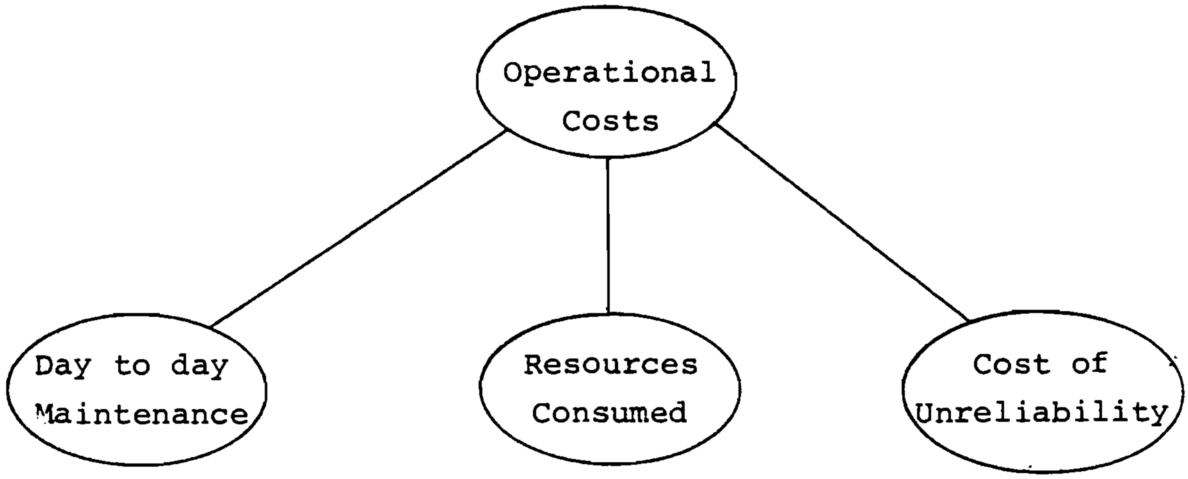


Figure 6. Components of Operational Costs

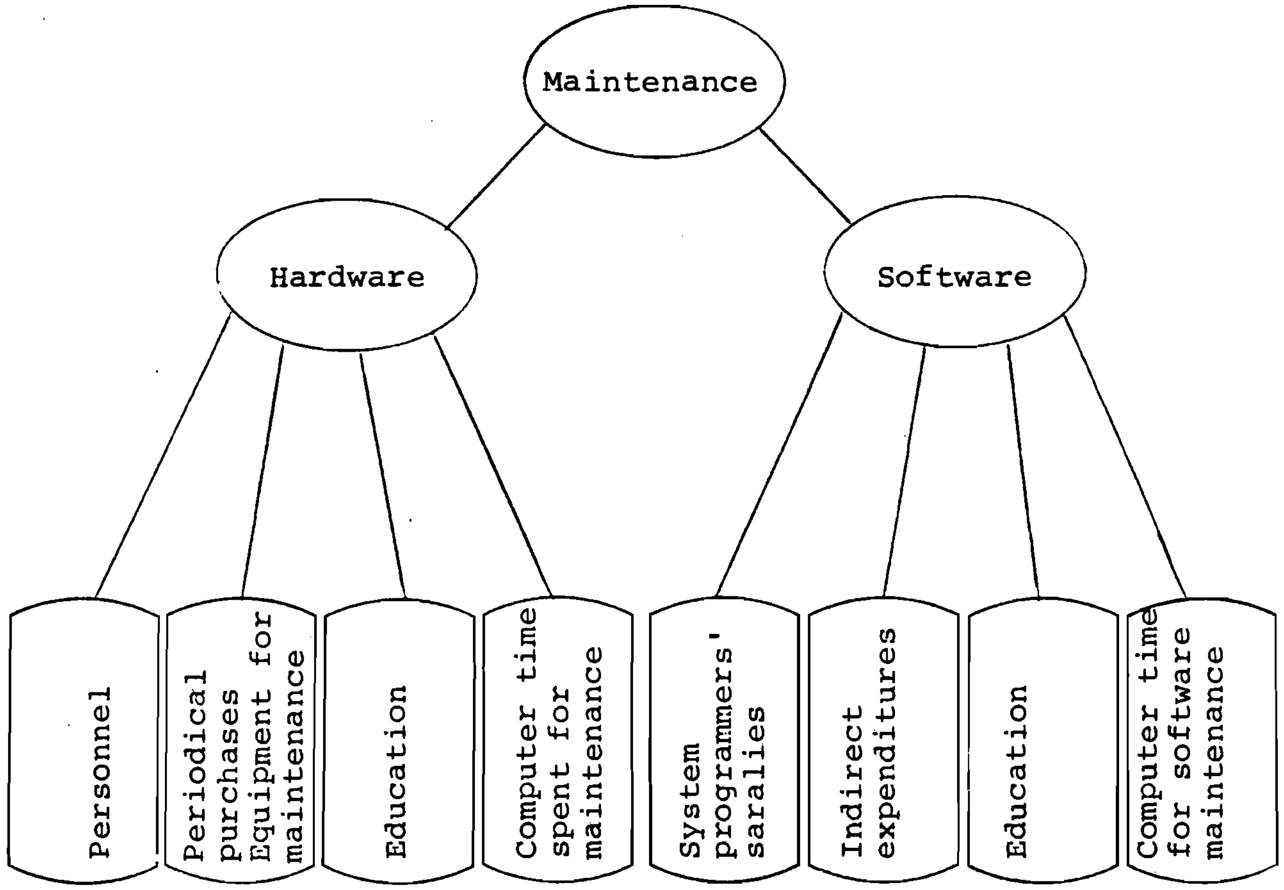


Figure 7. Components of Maintenance Costs

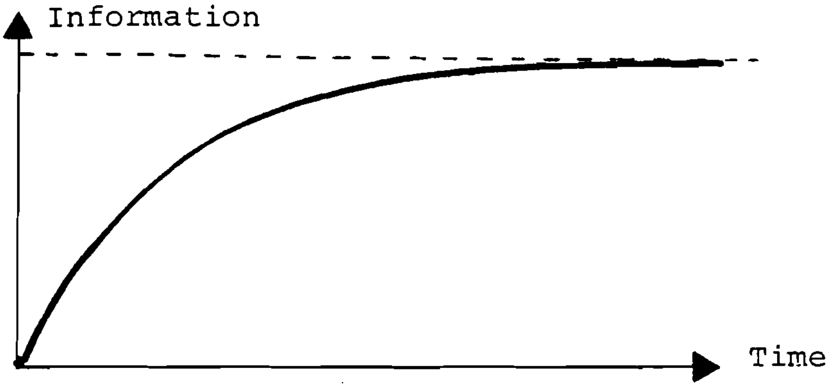


Figure 8. Human Abilities in Learning New Knowledge and Information

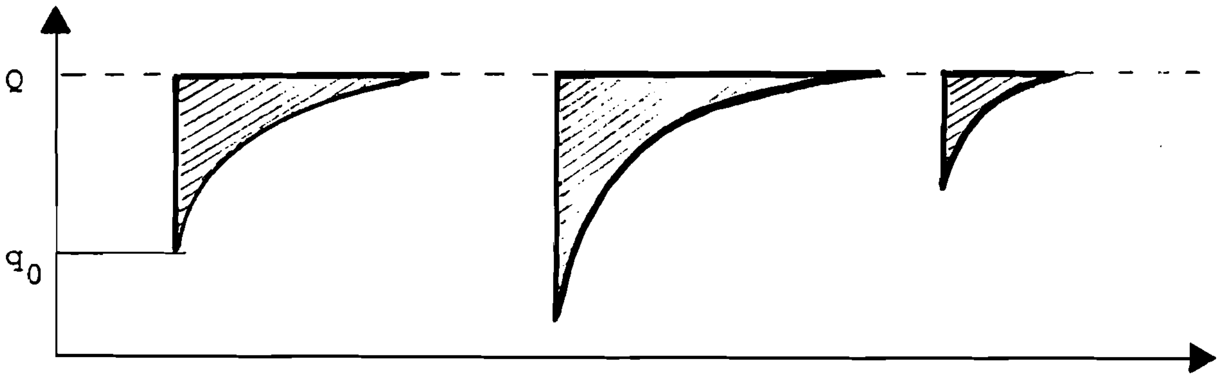


Figure 9. Educational Costs for One Working Place in Organization

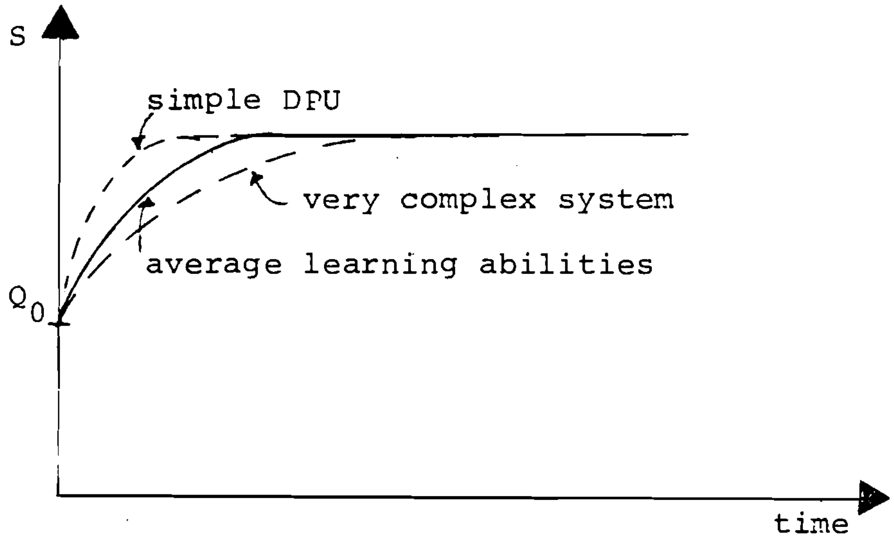


Figure 10. Illustration of the way complexity of the system influences educational cost.

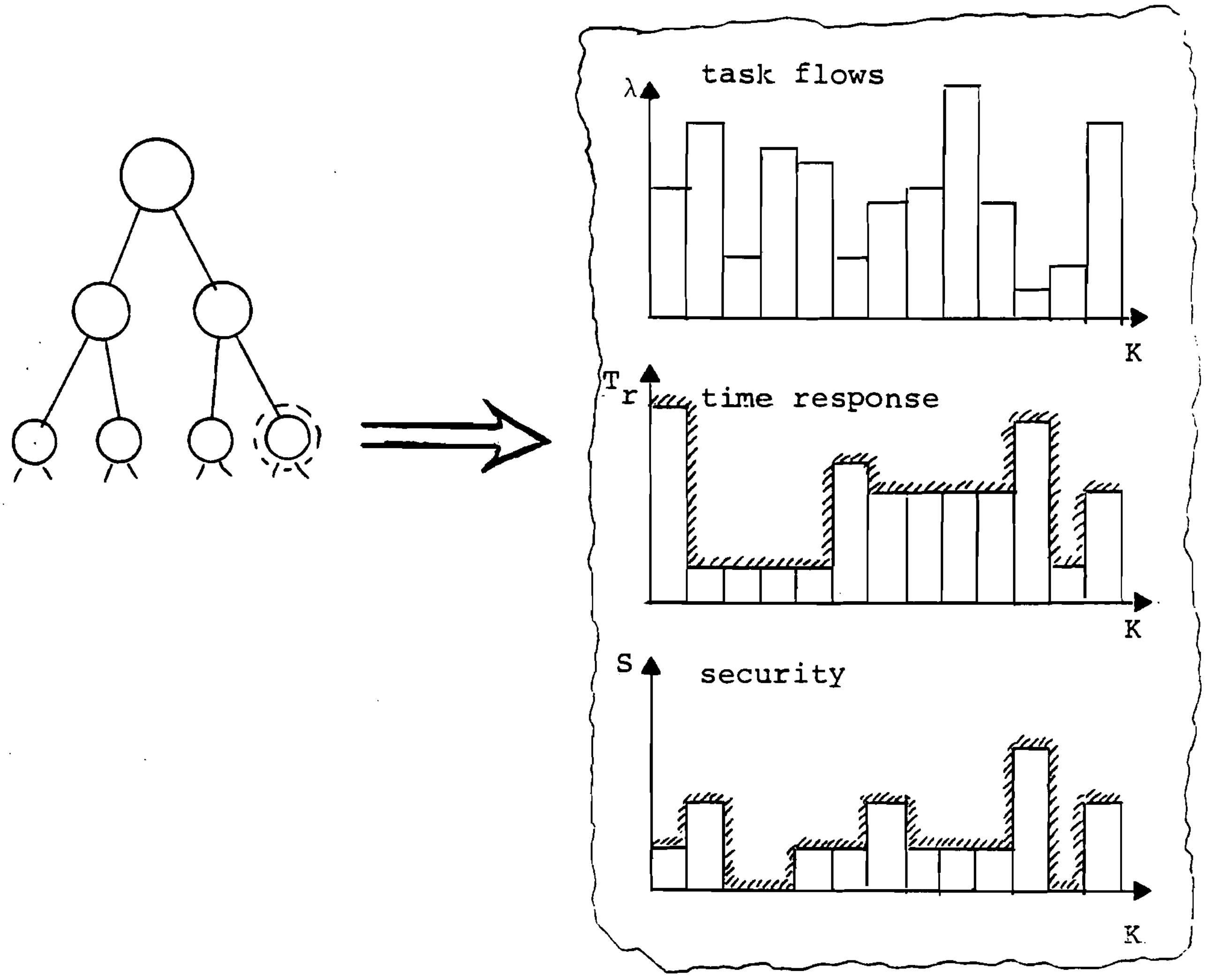


Figure 12. Correlation between load, processing constraints and organizational structure.

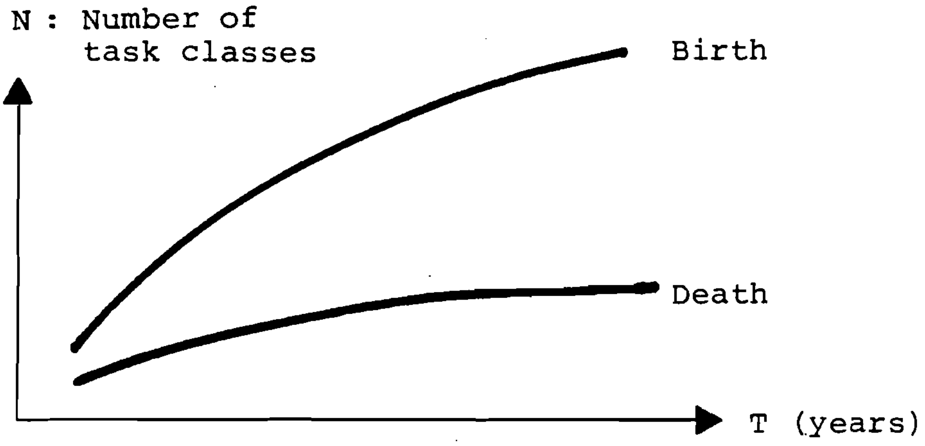
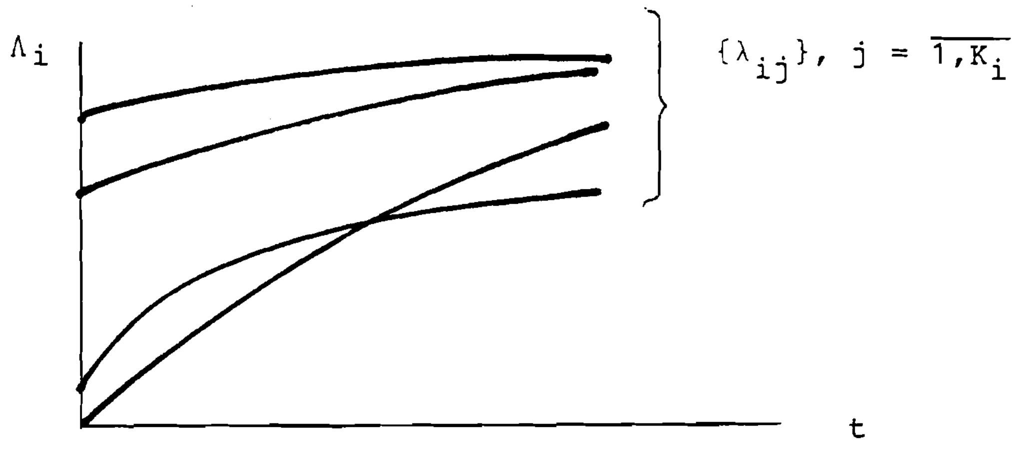


Figure 11: Increase in the Quantity of Task Classes, Generated by Average User



Fiture 13. Expected Increase of Task Flows Generated at i-th Working Place in Organization

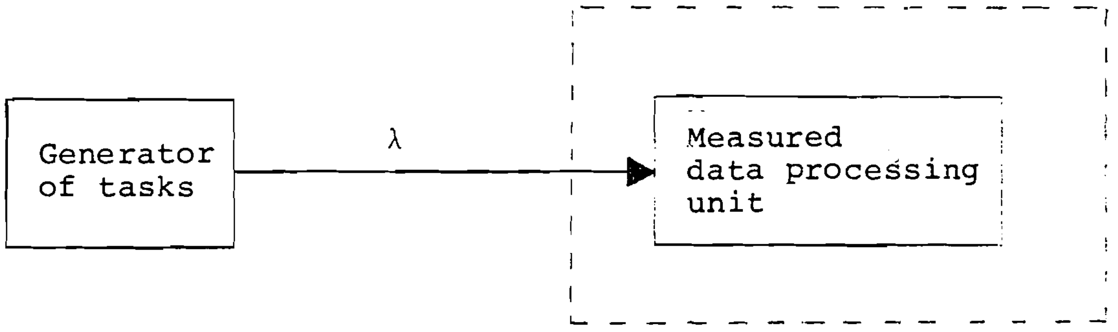


Figure 14. Measurement of Specific Tasks Data Processing Capabilities

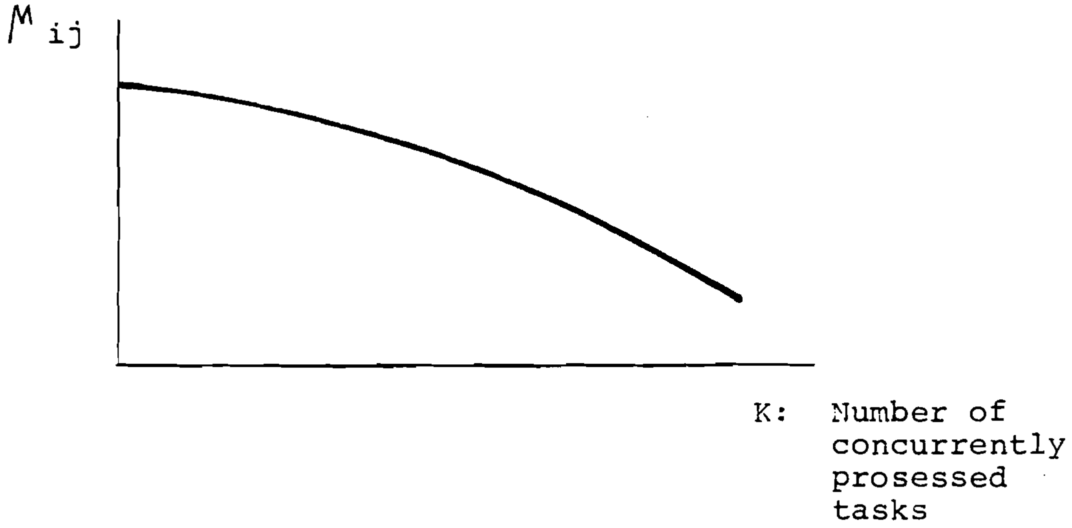


Figure 15. Decrease in Service Rate Caused by Interdependence of Task Flows and Overhead

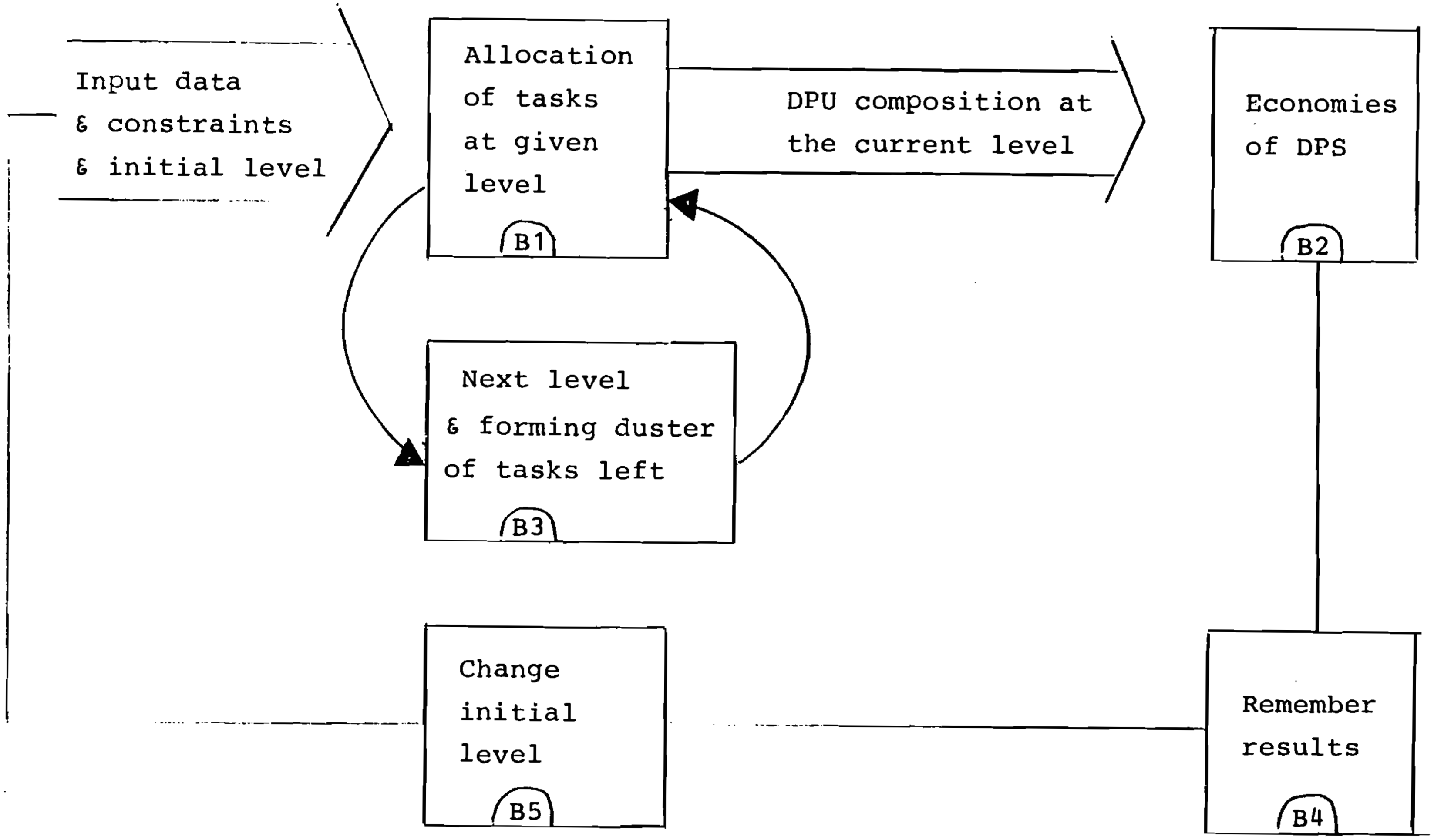


Figure 16. General structure of optimum decentralization search algorithm

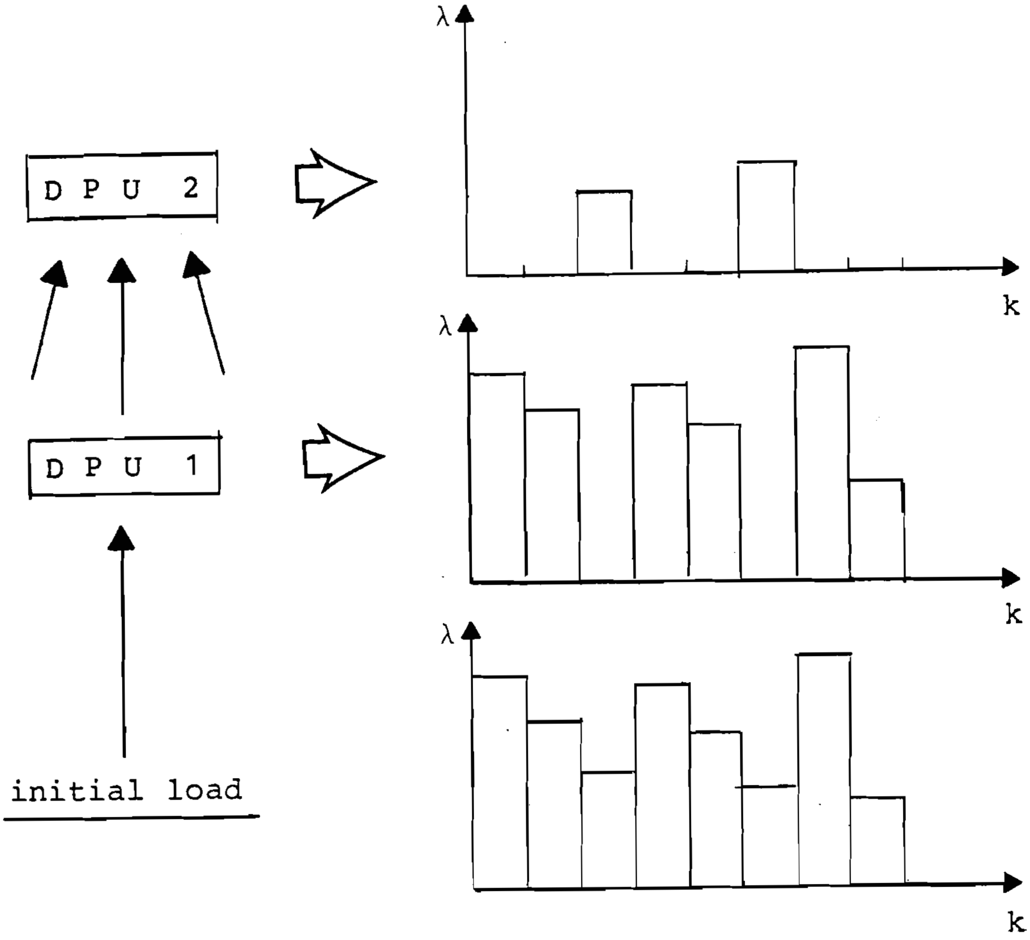
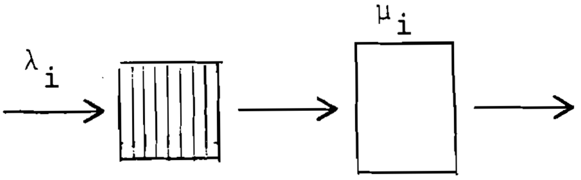


Figure 17. Filtering the load



$$\rho = \frac{\lambda}{\mu}$$

$$T_r^* \geq \frac{1}{\mu - \lambda}$$

Figure 18. M/M/1 queueing model

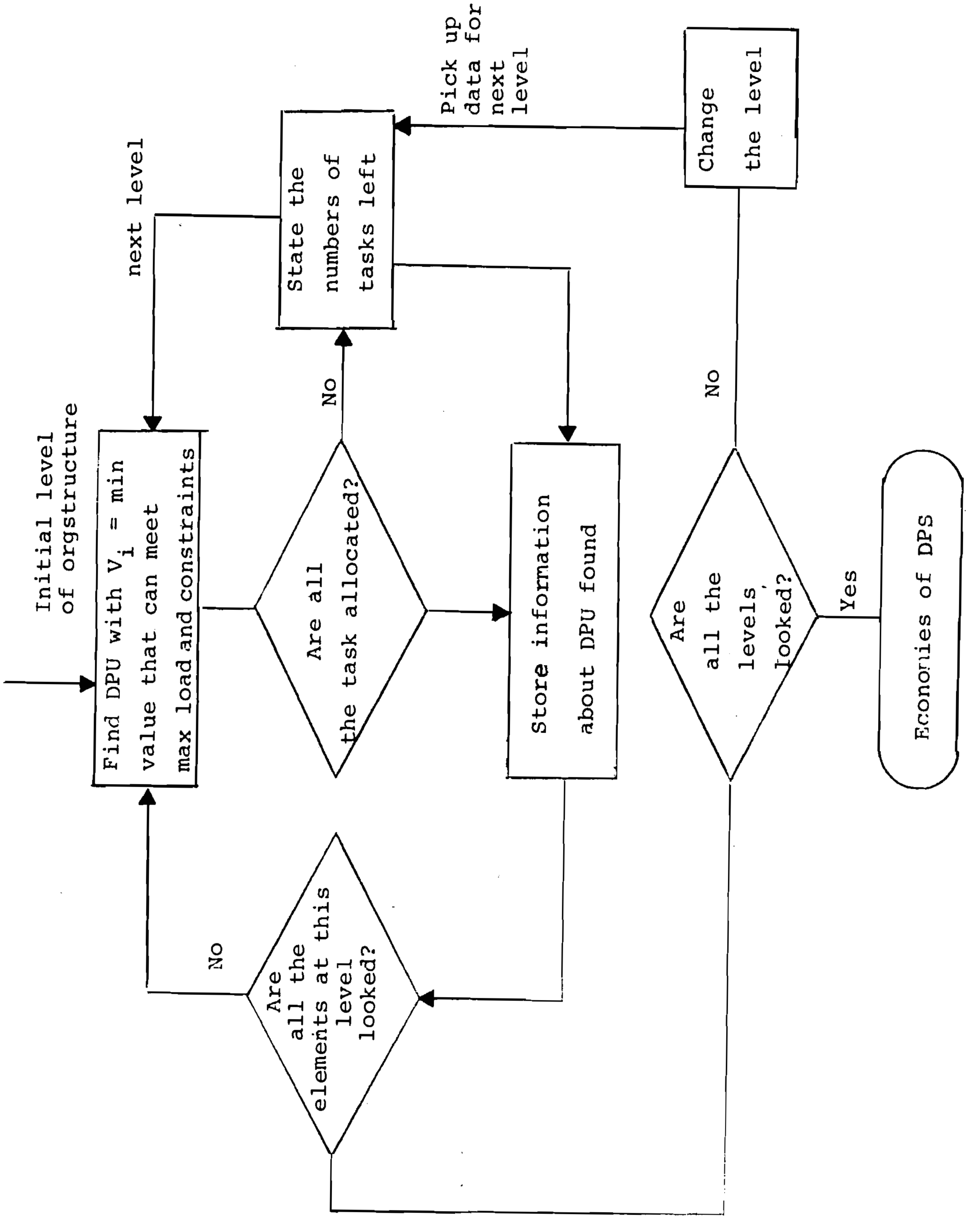


Figure 19. Algorithm for load allocation at decentralized DPS

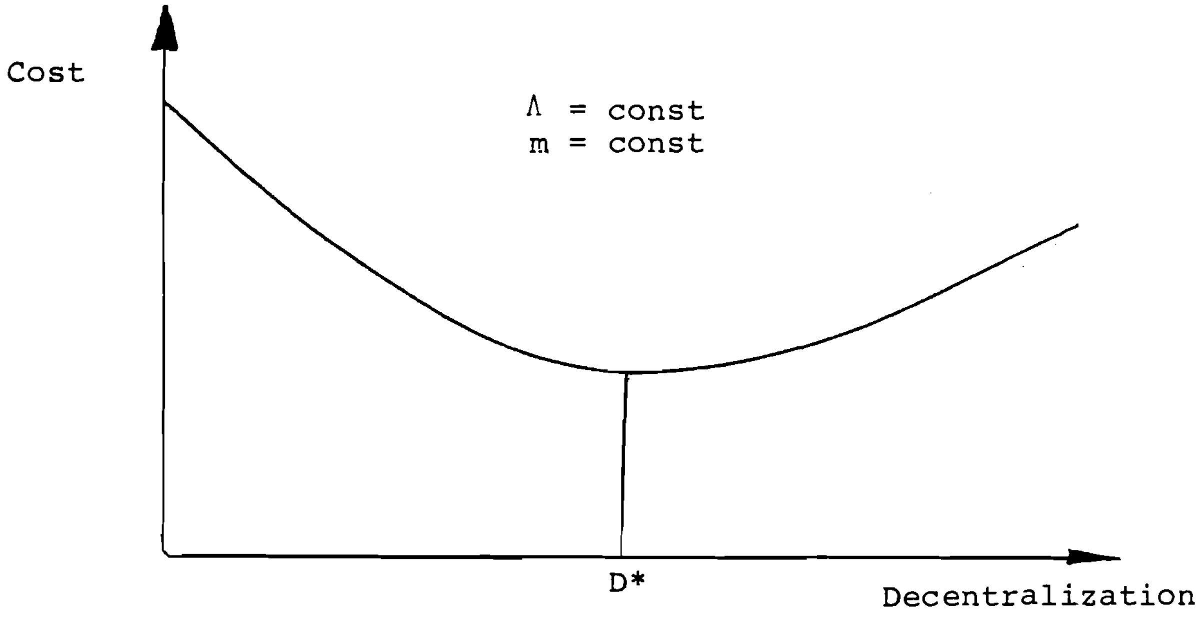


Figure 20. Optimality of the data processing system decentralization

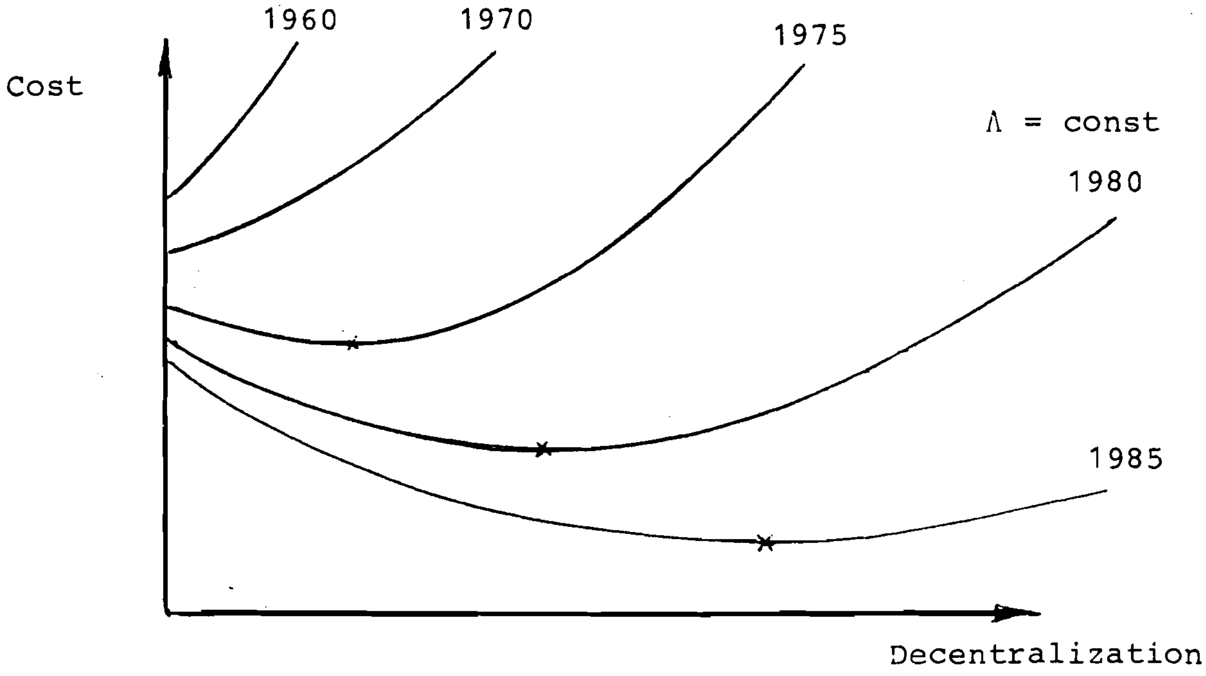


Figure 21. Decentralization optimum dependence on the level of DP technology

REFERENCES

[1] Ahituv, N., Borovitz, I. (1978) Analysis of the purchase option of computers. The Computer Journal 21(2): 105-109.

[2] EDP Performancd Management Handbook. Vol.1. Applied Computer Research. 1978. Phoenix, U.S.A.

[3] EDP Performance Data Processing Handbook. Vol.2: Tools and Techniques. Applied Computer Research. 1979.

[4] Aurmsrong, J.S. (1978) Long range forecasting: From crystal ball to computer. Wharton Econometric Forecasting Assoc. N.Y.: Wiley & Sons.

[5] Nolan, R. (1977) Restructuring the data processing organization for data resource management. Information Processing 77, IFIP. pp.261-165.

[6] Cotton, I.W. (1077) Cost-benefit Analysis of Interactive Systems. Computer Networks 1:311-324.

[7] Joslin, E.O. (1971) Costing the system design alternatives. Data Management. April:23-27.

[8] Streeter, D.N. (1972) Cost-benefit evaluation of scientific computing services. IBM Systems J. 3:219-233.

[9] Diehr, D.M. (1975) Putting a price on EDP services. J. of System Management. 26(3):16-19.

[10] King, J.L., E.L. Schremss (1978) Cost-benefit analysis in information system development and operation. Computer Surveys 10(1), March:19-34.

[11] Bair, J.H. (1978) Communication in the office-of-the-Future: where the real payoff may be. Evolution in computers communications. Proceedings of IV International Conference on Computer's Communications. Kyoto. pp.733-739.

[12] Carlisle, J.H., (1976) Evaluating the impact of office automation on top management communication. AFIPS Conference Proceedings. NCC. pp.611-616.

[13] Konkel, G.Y., and P.Y. Peck (1976) Traditional secretariat cost compared to word processing. The Office. February:67-68.

[14] Neidleman, L.D. (1979) Computer usage by small and medium sized European firms: An empirical study. Information and Management. pp.67-77.

[15] Kleinrock, L. (1976) Queueing systems. John Wiley & Sons.

[16] Timmerech, E.M. (1973) Computer selection methodology. ACM Computing Surveys. 5(4).

[17] Pask, G. (1975) The cybernetics of human learning and performance. Hutchinson & Co., Ltd.:London.

[18] Krantz, J. ed. (1974) Learning, Memory and Thinking, Contemporary Developments in Mathematical Psychology. Vol.1, Freeman and Company: San Francisco.