PARAMETRIC COST MODELS
OF NATIONAL DATA BANK
NETWORKS - VOL. II

McGILL STUDY

PART III : CENTRALIZATION vs. DISPERSION

1 NUMBER, LOCATION AND SIZE OF PLANTS

1.1 Introduction

1.2 Complete or partial dispersal of operations

1.3 Returns to scale

1.4 The economics of dispersion

1.4.1 Investment

1.4.2 Production

1.4.3 Interplant transactions

1.5 The form of the basic model

1.5.1 Maxima and minima of convex or concave functions

1.5.2 The relevant costs

1.5.3 The basic model

1.5.4 Nonconvexities: economies of scale

1.5.5 Pricing of data bank service: a digression

1.1 Introduction

In this section, following economic usage, the term "plant" will mean distinct productive facilities and "firm" will mean the organization which controls these productive activities. The questions to be addressed are, first, whether it is better, in terms of the firm's objective, to concentrate its entire production in a single plant, or to spread it over a number of plants at different locations. This involves the further question as to whether the dispersal of productive activities should be complete or partial, and the extent to which decision-making responsibility should be delegated to the various plants. A second set of questions, implicit in the first, concerns how production should be allocated between the various plants if multiplant operation is indicated, and how market areas should be assigned as between plants. Further, if there are transactions between the plants, how should these be priced?

For each of the above questions the economic conditions to be met will be specified, assuming that the firm's objective is profit maximization. This is the usual assumption made in economics and in most O.R. studies. It would be relatively easy to modify the analysis for the case of a non-profit organization (an objective of zero profit, or zero profit after making provision for equipment replacement and interest on debt). Similarly, though less easily, this static analysis could be adapted to the more realistic dynamic objective of maximizing the present value of the firm. It should be emphasized that the analysis seeks to provide optimal solutions to the questions addressed when viewed from the point of view of the firm.

The installation, and particularly the location, of national data banks by the Federal Government and its agencies would, presumably, be subject to macro-economic considerations such as the desire to reduce regional disparities in incomes and employment, and possibly to social and political considerations as well. We have assumed that none of these considerations fall within the present ambit of our study.

At this point it is worth explaining why we did not entitle this section "centralization vs. decentralization". The reason was that in the economic and business literature "decentralization" means the delegation of decision-making responsibility among the segments of a single-plant firm, or between the plants of a multiplant firm. In both cases it refers to existing plants. Since this is not the question we are primarily concerned with in this section (it does enter into the problem of pricing interplant transactions), it seemed desirable to avoid using the term. In terms of established economic categories, what we are concerned with is single- or multiple-plant operation, location and size of plants.

1.2 Complete or partial dispersal of operations

Setting up multiple plants could mean a number of different things in the context of a national data bank operation. Thus the data bank itself could be split up into a number of non-overlapping parts, or it could be copied, wholly or partly, from one central plant to a number of "branch" plants. The same might apply to computing facilities, if they were owned by the data bank firm. We will assume, however, that computing facilities are rented by all plants on a time-sharing basis. Conceivably, though improbably, the development of software systems could be carried out by a central plant for all plants.

There remains the possibility, equally applicable to rented as to owned computing facilities, of providing separate facilities at any given plant to service particular classes of users. That is, productive facilities may be separated in an "adjustment space" (also called "function space" or scope) rather than in geographic space [1]. For example, a central data bank may be organized to provide large-volume, regularly updated information, with access to large-scale computing facilities, while "branch" plants look after one-off retrospective searches. This kind of non-geographic separation of activities has also been referred to by W.E. Batten, Director of the U.K. Chemical Information Service, in [2], p. 284:

> ".... the so-called information centre (be it "national" or otherwise) has a further social duty. It is now the probable custodian of both disciplinary and 'mixed' data-bases. It must have organised those bases for fast and cheap searching at levels which may extend from the information manager who needs a large searchable sub-collection regularly updated, down to the individual who needs a one-off retrospective search on demand .... it may be inescapable that the larger centres will be involved in both 'wholesale' and 'retail' business for a long time to come--unless sub-centres emerge, based possibly on research associations or other cooperative bodies.
>
> What should viably subtend from the activities of the repackaging centres must depend upon a fine interplay between the forces of classification and the forces of the market. I have postulated 'large' interdisciplinary files, to be tapped by organisations and by individuals. It will always be for continuous study what degree of sub-packaging is warranted in anticipation of a volume of smaller and individual enquirers".

The Kochen and Deutsch study[1], which related to the dispersal of various kinds of services including libraries and computer systems, identified four aspects of dispersal: pluralization of facilities (e.g. service points or channels), dispersal in geographic space, specialization by function or kind of service, and adaptation to the specific requirements of each case through repeated feedback passes or negotiating queries. In their words:

> "Different functions or kinds of service are treated as being located in a function space; the distances among them correspond to the number and cost of adjustment steps which a server or a service facility needs to shift from one function to another".

With their dichotomy between geographic and functional dispersal, four forms of organization become possible: an organization centralized by service area and scope; centralized geographically but split up by scope; dispersed geographically but centralized by scope for each geographical area, and dispersed both geographically and by scope. Using a mathematical model, they go on to establish the conditions under which division of activities as described in the last three possibilities would be justified.[1]

This is not the place to comment further on the Kochen and Deutsch study. As a matter of interest, however, it may be noted that they identify ten key variables for calculating the optimal number of facilities when considering specialization in geographic space and function space in combination. This set of variables, all expressed as averages, comprise service load, geographic distance, the cost of time spent in transmitting a request and the response to it, speed of communication; the functional distance or number of functions,

---

1 Our decision to avoid use of the word "decentralization" when referring to dispersal of activities, whether in geographic of function space, receives support from the Kochen and Deutsch study. Though they clearly distinguish between organizational decentralization (delegation of responsibility) and geographical separation of activities in the text, their references to the literature are hopelessly confused as between the two.

the cost of time in adjusting to the function requested, adjustment speed, and total fixed cost per facility; number of negotiating queries per request, and an index of the value of a speedier response. Under certain assumptions (notably constant returns to scale in operations), the optimal number of facilities for a single-function system or a multi-function single-location system is found to vary approximately as the square root of the service load. For a multi-function, multi-location system the optimal number of facilities is found to vary in proportion to the two-thirds power of the service load.

Also of passing interest is their general conclusion:

> "Long term trends may be toward decentralization when service loads and the costs of service time grow faster than capital costs and transport and adjustment speeds, as seems likely for the next several decades. Where the opposite conditions prevail, cost-effectiveness should favour centralization, such as perhaps in some earlier periods, and possibly in the more distant future".

A question which immediately comes to mind concerning the Kochen and Deutsch study is whether, and to what extent, these conclusions may have been influenced by their assumption of constant returns to scale. Without further investigation, it is questionable whether this assumption was the most realistic one to make.

1.3 Returns to scale

As will become apparent later, the form of analysis of the problems stated at the beginning of this section hinges critically on the characterization of returns to scale, i.e. on what happens to the physical quantity of output when the physical quantities of all inputs are changed in the same proportion. In particular, do the costs of building and operating data bank facilities increase in strict proportion to system size, more than or less than proportionately? And how do communications costs change with changes in the volume of data transmitted? If doubling (or halving) all inputs results in exactly doubling (or halving) output the production process is said to possess constant returns to scale; if doubling (or halving) all inputs more than doubles (or halves) output, it shows increasing returns to scale; if it less than doubles (or halves) output it shows decreasing returns to scale.

If a constant returns to scale technology in the activities referred to describes, to a reasonable approximation, the relationship between changes in cost with changes in output (all input prices being assumed constant), modelling of the first set of problems referred to earlier takes a relatively simple form: that of a linear programming problem or a mixed integer programming problem of the transportation type. In the case of non-constant returns to scale in respect to any of the system costs separately identified, resulting in nonlinear terms in the objective function to be minimized, certain difficulties may arise in falling back on piecewise linear approximations of the cost functions.[2] Under certain conditions (viz. that both the objective function and constraints are separable nonlinear functions[3]),the original problem can be replaced by an approximating problem, and if further conditions on the functions are met (viz. that they have the appropriate convexity

---

2 Hadley refers to this problem briefly in [4], section 12.6 et seq., and in detail in [3], chapter 4.

3 If they are not, it is often possible to convert them to this form by transformations of variables. This enlarges the problem, however, by imposing at least one additional constraint for each new variable.

or concavity properties) a local minimum can be obtained which will also be a global minimum for the transformed problem, and hence an approximate optimal solution to the original problem. In the cases in which these conditions are met there is no great difficulty in extending the analysis to deal with the nonlinearities in one of the ways outlined by Hadley in [3]. But in certain classes of problems encountered in practice considerable computational difficulty results. Heuristic procedures have been developed for dealing with some of these classes of problems. We take up this question again in sections 1.5.1, 1.5.3, and 1.5.4.

At this stage we merely draw attention to the question of returns to scale. Their nature is fairly crucial to the outcome of the set of problems we are addressing, and evidently we should attempt to investigate their nature as closely as possible. As far as the solution method to be employed is concerned, much depends on the relative magnitude of cost changes with changes in scale of operation. If the cost changes were relatively small it would be permissible to use the device of treating them as fixed start-up costs, thus effectively converting the problem into a mixed integer programming problem (of the fixed-charge discrete transportation type). This device has been followed in [5], [6] and [7]. It is, of course, not admissible where economies of scale are expected to persist over the entire range of sizes of facilities considered.

In a recent (1973) study on computing facilities, alluded to elsewhere in this report, Streeter [8] observes that the frequently acknowledged economies of scale attaching to computing equipment, which he expresses in the relationship $E = KC^2$, where E denotes system effectiveness, C system costs and K is a constant proportionality factor, are becoming more complex: "the observed effect of scale may [now] be somewhat greater or somewhat less than quadratic". Streeter leaves us in no doubt, however, that there are substantial economies of scale in computing equipment, the principal sources of these now being larger and faster storage and data channels rather than the computer itself, together with large economies in personnel costs which are also assuming a growing proportion of operating costs. The subject is also reviewed by Sharpe [9], pp. 314-322, who presents some evidence in relation to third-generation equipment.

What is true of equipment and labour costs is also, according to Streeter, true of a number of other cost components, such as floor space and number of software packages to be maintained. Inter-installation communication charges exhibit diseconomies of scale according to him, and thus also favour centralized computer operations.

The forces at present favouring geographical dispersion of computing are, according to Streeter's account, less tangible, the most obvious advantage being in lower user-computer communication costs. Streeter's solution to the problem is to propose a strategy for reaping the chief advantages of both centralized and dispersed computing services. Essentially, this strategy involves a geographic separation of operations resting upon a partitioning of the function space referred to by Kochen and Deutsch, and Batten: certain standard, large-volume services are to be provided centrally, while "locally anomalous personalized or evolving services" may be better provided on site.

1.4 The economics of dispersion

Whether we are concerned with geographical or functional dispersion of data bank activities or some combination of the two, we can say very broadly that it will pay to partition operations geographically and/or functionally if (i) this results in lower costs in the long run than centralized operation, or if (ii) it increases revenues more than costs (e.g. by expanding the market or by raising the quality of service provided), or if (iii) it reduces risks, <u>ceteris paribus</u>. This third condition might apply to certain "classified" or sensitive government information, particularly that relating to defence and international relations.

While not losing sight of the last two conditions, for most practical purposes we may safely concentrate upon the first.

What specific form does the centralization vs. dispersion decision take? From an economic point of view it consists of a set of decisions, if we include certain decisions to which a dispersed mode of operation (whether geographical or functional) would give rise.

1.4.1 <u>Investment</u>

We first note that any decision to disperse or partition a data bank, whether already existing or only in contemplation, will involve some degree of investment. That is, some expenditures will have to be incurred which will only yield up their benefits over a number of years. These capital expenditures (or start-up costs) will, depending on the form of dispersion and method of operation, include the costs of removal or duplication of the existing data base (in whole or in part) or, where the system has yet to be set up, any costs of acquiring the right to reproduce data. With complete duplication each facility would have the same capacity to handle the total volume of service demanded as if there were only one centralized facility. As Kochen and Deutsch note [1, pp. 841-2], if dispersion to n facilities is indicated on cost grounds, it would be even more favoured if a lesser degree of redundancy were permitted. Other items of capital expenditure might include the cost of acquiring a building (or a long-term lease on office space), acquiring computing facilities (or long-term leasing of same), and the cost of communication lines if they must be provided by the facility.

Two conditions must be satisfied for investment to be justified. In stating these we will assume simply that the firm's objective for investment is to maximize the net present value of cash flows and, for production decisions, to maximize profits.[4]

---

4 More consistency between decision rules and objectives would require:

	Static	Dynamic
Corporate objective	Profit maxn.	Max. the present value of the firm
Production objective	Max. profits s.t. a single-period production function:	Max. profits period by period in a way which is consistent with maximizing over the firm's planning horizon, s.t. a multiperiod production function. For the conditions, see [13], pp. 263-4
decision rule	MR = MC TR > TC	
Investment objective	Max. the utility of the consumption stream provided by future dividends paid to owners of firm.	The same, but treated dynamically
decision rule	If capital markets are perfect and there is no capital rationing: Max. the NPV of cash flows over the set of independent projects considered; NPV > 0 for each independent project (no explicit allowance for uncertainty)	

The two conditions are as follows:

(i) (a) <u>If a central data bank already exists</u>:

The marginal cost (MC) of centralized operation must exceed the MC of operating central plus branch facilities, or the MC of a fully dispersed series of branch facilities.

The MC of centralized operation will consist of variable operating costs; the MC of branch operations will include, in addition, the discounted and annualized capital cost of the branch facilities.

If K_j is the capital cost associated with establishing plant j, n years its estimated economic life and S_j its value at the end of this period, the annual equivalent capital cost in discrete terms is $(K_j - S_j v^n) a^{-1}_{\overline{n}|i}$ measured at the cost of capital rate i, where $v = (1+i)^{-1}$

and $a^{-1}_{\overline{n}|} = 1/(v + v^2 + v^3 + \ldots\ldots + v^n) = i/(1 - v^n)$.

The corresponding expression with continuous discounting is

$$(K_j - S_j e^{-n\delta}) \cdot \frac{\delta}{1 - e^{-n\delta}},$$

where $\delta = \ln(1+i)$ is the continuous rate of interest.

(b) <u>If no data bank yet exists</u>, the comparison will be between variable operating costs and capital costs of both centralized operation and dispersed operation, measured at the margin.

This is the <u>necessary</u> condition for investment to be justified.

(ii) The <u>sufficient</u> condition is that the investment must justify itself at the firm's cost of capital, the opportunity cost of investing measured at the margin, and in competition with all other investment opportunities under consideration by the firm at the time. With no budget or resource constraints, this simply means that the investment must satisfy the NPV criterion. This criterion does not explicitly allow for uncertainty. This second condition is not mentioned in any of the literature we have seen.

1.4.2 Production

If dispersal of data bank facilities is indicated by the above conditions, a further condition is needed for determining how to operate them. The necessary condition here is that each facility be operated at that level at which its MC (here no longer including the cost of capital inputs as in the investment decision, but instead economic depreciation, representing the fall in value of the facilities due to use) equals the MR of the firm as a whole. The sufficient condition would require that MC in each plant should be increasing more rapidly than MR of the firm as a whole at the optimum point.

1.4.3 Inter-plant transactions

Dispersal of operations also raises the possibility that some transactions may develop between facilities. For example, one facility may communicate information to one or more other facilities to update or modify their data bases, or the development of a particular kind of software by one facility may be made available to other facilities within the system. In cases such as these we are presented with the problem of transfer pricing, i.e. of determining the appropriate prices to govern these inter-facility transactions. Like the sufficient condition for investment, none of the literature on plant and warehouse location acknowledges this problem.

In practice a great variety of different methods is to be found among industrial firms. Transfers are sometimes based on outside market prices (if an outside market exists), made at standard cost, actual cost (in each case it may be direct cost or direct cost plus some overhead allocation), actual cost plus return on investment, or by free negotiation between the departments or divisions concerned. Most of these methods are inconsistent with a production objective of profit maximization, and none of them is economically appropriate in all circumstances. The whole purpose of dispersal of activities is to increase the efficiency of the firm in terms of its objective. Besides affecting the efficiency of internal resource allocation, the prices which govern these internal transactions will affect the level of operation of the activities concerned, the performance measure by which each activity is judged, and the profitability of the firm as a whole.

The "transfer price problem" is a problem in the adaptation of price mechanisms to the internal environment of the firm. When optimally determined, the transfer price should measure the opportunity cost of the product or service transferred to the firm as a whole, measured at the margin. Only then will the transferred good or service be used at the optimal level relative to all alternative uses and to all constraints upon optimization of the firm's objective. The neoclassical theory of the firm makes the implicit assumption that all internal allocations of resources are made under perfectly competitive conditions. This ignores a number of external (market) and internal (organizational) factors, and would not in general lead to an optimal pricing rule.

In the model we shall develop it will be assumed that prices are determined optimally for all internal transfers, after taking into account all the costs of implementing the system. The theory of optimal transfer pricing is set out in references [10] and [11] and is summarized in [12], pp. 132-5 and Appendix V, and in [13], pp. 256-9, 529-530 and 576-7.

Before leaving this subject it may be observed that operation of such an internal pricing system runs up against some substantial practical difficulties. Optimal transfer prices can only be determined after information relating to the intermediate product market (if one exists) and the final product market has been obtained. Goal congruence between separate activities and the firm as a whole can be achieved by the prices being determined centrally by central management for all activities (as they are under economic socialism) or locally, by the activities concerned (as are market prices under a free enterprise system). The question is whether an optimal pricing system to ensure goal congruence is justified under the latter method if the activities are not separable (in view of the increased transmission of information which is then required between them), and, more importantly, consistent (or possible, when there are significant interdependencies between the activities) with the desired degree of decentralization of authority within the firm. Centralized setting of transfer prices, by reducing local autonomy, may have serious disincentive effects in the activities, and it may be necessary to introduce incentives to make the prices effective. It would also result in an increased transmission of information within the firm; and any pricing system is worth while only if its estimated benefits through greater efficiency exceed the costs of operating it. Moreover, the optimal transfer prices cannot be estimated with certainty, and hence frequent changes may be necessary if resource allocation is not to be distorted. Finally, the prices so determined are meant to govern marginal adjustments in output.

Performance measures for the separate facilities struck after pricing internal transfers optimally are not in general the appropriate data on which a decision to continue or abandon the facilities should be taken, because this involves non-marginal considerations.

1.5. The form of the basic model

1.5.1 Maxima and minima of convex or concave functions[1]

Before beginning the technical discussion concerning the form of the basic models it is necessary to define a number of terms relating to nonlinear functions.

Definitions:

Convex set: A set X is convex if, for any points $\underline{x}_1$ and $\underline{x}_2 \in X$, every convex combination of $\underline{x}_1$, $\underline{x}_2$ is also in the set, i.e. the line segment joining $\underline{x}_1$, $\underline{x}_2$ is also in the set. A set consisting of a single point is convex.

Convex combination: The line passing through two different points $\underline{x}_1$ and $\underline{x}_2$ in R^n is defined as the set of points $X = \{\underline{x} | \underline{x} = \lambda\underline{x}_2 + (1 - \lambda)\underline{x}_1,\ \text{all}\ \lambda\}$. If $0 \leq \lambda \leq 1$, the set represents the line segment joining $\underline{x}_1$ and $\underline{x}_2$. For a specified λ in this range, the point $\underline{x} = \lambda\underline{x}_2 + (1 - \lambda)\underline{x}_1$ is called the convex combination of $\underline{x}_1$ and $\underline{x}_2$.

Closed set: is a set which contains all its boundary points. A set need not possess boundary points. It is also possible for a set to be neither open nor closed.

Convex function: A function $f(\underline{x})$ is convex over the convex set $X \subset R^n$ if, for any two points $\underline{x}_1$ and $x_2 \in X$,

$$f[\lambda\underline{x}_1 + (1 - \lambda)\underline{x}_2] \leq \lambda f(\underline{x}_1) + (1 - \lambda)f(\underline{x}_2)\ , \qquad 0 \leq \lambda \leq 1$$

---

[1] This section may be skipped without loss.

$f(\underline{x})$ is strictly convex if the strict inequality holds for all λ such that $0 < \lambda < 1$ and $\underline{x}_1 \neq \underline{x}_2$.

Concave function: A function $f(\underline{x})$ is concave or strictly concave if $[-f(\underline{x})]$ is convex or strictly convex, respectively.

Alternative definitions: $f(\underline{x})$ is convex over the convex set X if and only if, for all $\underline{x}$, $\underline{x}^* \in X$

$$f(\underline{x}) - f(\underline{x}^*) \geq \nabla f^*(\underline{x} - \underline{x}^*),$$

with the inequality reversed for concavity and strict for strict convexity or concavity.

A function is locally convex or concave at $\underline{x}^*$ if the set X is the neighbourhood of $\underline{x}^*$.

(It is essential in the definition that X be a convex set, since we require that $\lambda\underline{x}_1 + (1 - \lambda)\underline{x}_2$ be in X if $\underline{x}_1$, $\underline{x}_2$ are. X may be R^n, in which case the function is globally convex or concave.)

Linear function: A linear function is both convex and concave, but not strictly convex or concave.

The sum of nonlinear functions: Consider the sum $f(\underline{x}) = \Sigma f_j(\underline{x})$ of a number of convex functions, defined over the same convex set X. We have:

$$\begin{aligned} f[\lambda\underline{x}_1 + (1 - \lambda)\underline{x}_2] &= \Sigma f_j[\lambda\underline{x}_1 + (1 - \lambda)\underline{x}_2] \\ &\leq \Sigma[\lambda f_j(\underline{x}_1) + (1 - \lambda)f_j(\underline{x}_2)] \\ &\leq \lambda f(\underline{x}_1) + (1 - \lambda)f(\underline{x}_2). \end{aligned}$$

Hence the sum of convex functions is convex (and the sum of concave functions is concave).

Since $cf(\underline{x})$ is obviously convex if $f(\underline{x})$ is convex and $c > 0$, any positive linear combination of convex (concave) functions is convex (concave).

Maxima and minima

Consider the problem of determining the maximum or minimum of $f(\underline{x})$ over the closed convex set $X \subset R^n$, subject to $g_i(\underline{x}) = b_i$. It is assumed that f and g_i are both separable and are everywhere $\varepsilon\, C^1$ (where C^1 indicates that f and g_i and their first derivatives are continuous over some subset of R^n).

(1) Linear case: If $f(\underline{x}) = \Sigma c_j x_j$ and $g_i(\underline{x}) = \Sigma a_{ij} x_j$, determination of the optimal values is a linear programming problem:

$$\begin{aligned} &\text{max or min} \quad \Sigma c_j\, x_j \\ &\qquad \text{s.t.} \quad \Sigma a_{ij}\, x_j = b_i\,, \qquad i = 1, \ldots, m \\ &\qquad\qquad\qquad x_j \geq 0\,. \end{aligned}$$

(2) $f(\underline{x})$ convex: The problem is now of the form:

$$\begin{aligned} &\text{max or min} \quad \Sigma f_j(x_j) \\ &\qquad \text{s.t.} \quad \Sigma a_{ij}(x_j) \quad \{\leq, =, \geq\}\; b_i \\ &\qquad\qquad\qquad x_j \qquad \geq 0 \end{aligned}$$

where $f(\underline{x}) = \Sigma f_j(x_j)$. The objective function is convex if each f_j is convex. If there is a feasible solution to the problem, the set of feasible solutions will be convex if the $a_{ij}(x_j)$ are:

concave whenever the i'th constraint has a $\geq$ sign;

convex whenever it has a $\leq$ sign

linear whenever it has an = sign.

(these are what were referred to in section 3 as the "appropriate convexity or concavity properties.")

If the set of feasible solutions is convex, and the f_j are all convex, a local minimum of the objective function over the set of feasible solutions is the global minimum. If the set X is bounded from below and the global maximum of $f(\underline{x})$ is finite, the global maximum will occur at one or more extreme points of X. If the f_j are all strictly convex, the global optimum will be unique, but not otherwise.

(3) $f(\underline{x})$ concave: The problem has the same form as in (2). If the set of feasible solutions is convex and the f_j are all concave, a local maximum of $f(\underline{x})$ is also a global maximum. If X is bounded from below and the global minimum of $f(\underline{x})$ finite, it will occur at one or more extreme points of X. If all the f_j are strictly concave, the global optimum is unique.

Approximating problem

By making use of the device of piecewise linearizations (polygonal approximations) of the f_j and a_{ij} approximating problems may be formulated and used to solve the above nonlinear programming problems. If the original problem has a unique optimal solution and its objective function is strictly convex or strictly concave, the solution of the approximating problem will be an approximation to the global optimum for the original problem. Note, however, that it is not necessarily true even then that the approximating problem will have a unique optimal solution, because its objective function will not be strictly convex (or strictly concave).

1.5.2 The relevant costs

Essentially, we shall be concerned in our modelling of the centralization vs. dispersal problem with the necessary conditions for setting up one or more plants, with the siting and size of those plants, and with which market or markets each plant should serve. It is intuitively easier to pose the problem in terms of geographical dispersion than of functional specialization, though the same principles apply in both cases. The principal variables with which we shall be concerned are variable operating costs at each plant, the costs of communication between all combinations of plants and markets, and the capital costs of establishing each plant, expressed as annual equivalents. The cost of inter-plant communications will be assumed to have been included in plant operating costs. It is further assumed that operating and capital costs for all plants are known, and that prices are the same (for an equivalent service) in all markets, an assumption never made explicit in any of the models we have seen in the literature. We return to this last point later. Certain other refinements will be held over at this stage, e.g. the importance of speed of response and communication (a factor included in the Kochen and Deutsch analysis [1]), or the fact that in periods of full employment the siting of plants may be considerably influenced by availability of labour of the requisite type. A finite number of facilities is also assumed.

1.5.3 The basic model

It will be assumed for simplicity that there are $i = 1, \ldots, m$ possible plant locations to serve $j = 1, \ldots, n$ market areas. Each plant supplies the same single service and holds no inventories[5] (so that output = sales). The basic model is known in the literature as a fixed charge transportation type model. With slight modification it may be stated as follows:

$$\min \quad \sum_{i,j} c_{ij}\, x_{ij} + \sum A_i \delta_i$$

$$\text{s.t} \quad \sum_{j=1}^{n} x_{ij} = a_i\,, \qquad a_i > 0\,, \quad i = 1, \ldots, m$$

$$\sum_{i} x_{ij} = b_j\,, \qquad b_j > 0\,, \quad j = 1, \ldots, n$$

$$x_{ij} \geq 0\,, \quad \text{all } i, j$$

$$\text{where } \delta_i = \begin{cases} 0 & \text{if } \sum_j x_{ij} = 0 \\ 1 & \text{otherwise} \end{cases}$$

and the first constraint is a plant capacity constraint. The A_i are called fixed charges because they are incurred only if $\sum_j x_{ij} > 0$. The charge is not a function of the output of plant i ; in terms of our problem, the cost of installing and operating a plant at location i is treated as a fixed cost, invariant with the size of plant i.

But for this A_i term, the problem would be a straightforward linear programming problem. (If all the A_i are equal and the problem is not degenerate, an optimal solution to the linear programming problem when the fixed charges are ignored is also an optimal solution to the fixed charge problem [14].) The fixed charge makes the problem nonlinear; the objective function becomes concave over the range of values of x_{ij} considered. An optimal solution to this problem occurs at an extreme point of the convex set of feasible solutions. With a fixed charge

---

5 Of course one of the distinguishing features of a data bank is that it does hold inventories; in effect it stores the negatives of all photographic prints it sells over some predetermined period. Our present formulation in effect assumes that the variable portion of this carrying cost is included in variable operating costs. A more accurate formulation would show it as a separate term. There is also the prior problem of determining the optimal holding period.

associated with each $\sum_j x_{ij}$, every extreme point is a local optimum. Some of these local optima, however, may differ from the global optimum (may, in fact, be far from the global optimum). Approximation techniques can be used to establish a local optimum, but the procedure is not computationally efficient if the objective function is concave. Finally, if, as is likely, the number of markets or demands (j) is large relative to the number of plants (i), an optimal solution will only very rarely permit a given demand to be supplied by more than one plant: the amount sold to market j from plant i will usually be $\min(b_j, a_i) = m_{ij}$, [3], p. 139.

The fixed charge problem can alternatively be formulated as a mixed integer-continuous variable linear programming problem:

$$
\begin{aligned}
\min \quad & \sum_{i,j} c_{ij} x_{ij} + \sum A_i \delta_i \\
\text{s.t.} \quad & \sum_i x_{ij} = b_j \\
& \sum_j x_{ij} - d_i \delta_i \leq 0 , \qquad \delta_i \text{ integer} \\
& 0 \leq \delta_i \leq 1 \\
& x_{ij} \geq 0 , \quad \text{all } i, j
\end{aligned}
$$

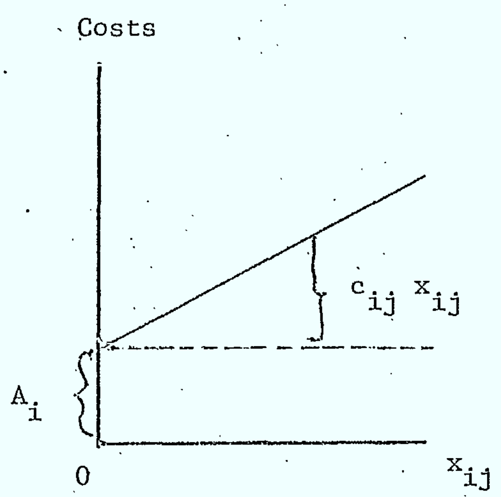
where d_i is the upper bound (assumed to have been determined) on the total output (sales) of plant i, $\sum_j x_{ij}$. This approach is still incomplete in that it can yield only a local optimum. Integer programming algorithms exist for this type of problem. They do not, however, have a high degree of computational efficiency.

1.5.4 Nonconvexities: economies of scale

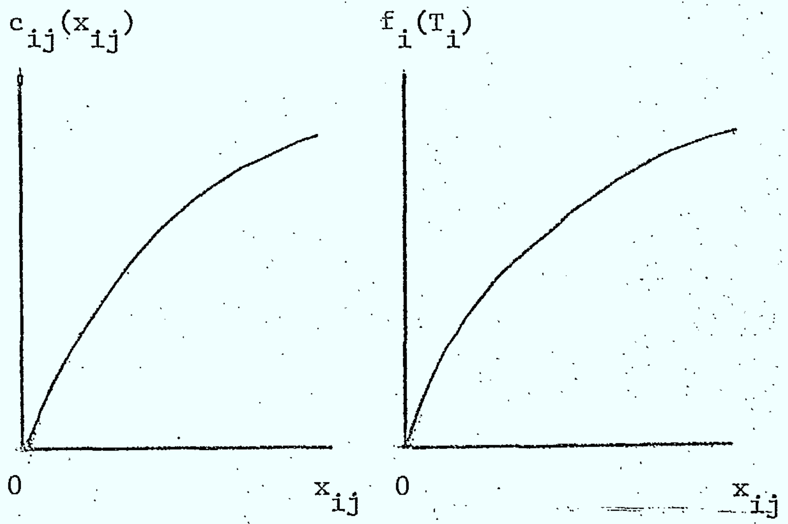
The above formulation is unlikely to be satisfactory as a representation of the problem of determining the number, location and size of data banks for two reasons. First, the first term in the objective function, $\sum_{i,j} c_{ij} x_{ij}$, which in our problem will represent communications costs between data bank and user, may be nonlinear - may, in fact, be a concave function $\sum_{i,j} c_{ij}(x_{ij})$. Whether this is so can be determined empirically, and we return to this point later. Secondly, it is most unlikely that the second term can be represented simply as a fixed cost. This cost comprises the variable operating cost and the annual equivalent capital cost of each of the plants which the model considers for inclusion in the data bank network. Feldman et al. [15] concluded in their study of the warehouse location problem that "optimal sizing and locating of facilities are very sensitive to the shapes of the warehousing cost functions" (comprising the cost elements just described).

Their model assumed that the second term in the objective function was concave (the first linear), due to the existence of economies of scale ("big warehouses are more 'efficient' than small ones"). Economies of scale in the operation of large plant units is a fairly general phenomenon in many industries (though as the firm grows larger these operating economies tend to be lost to some extent by a counter-tendency for overhead costs to rise more than proportionately: the firm develops "organizational slack" [16]) In the present problem, however, we are concerned only with the costs that vary as a result of establishing (or dispersing) and operating the data banks.

To be more precise, a reasonable initial assumption would be to expect variable operating costs to increase less than proportionally with changes in scale of output, and the capital cost component to be at worst linear, and probably concave also.[6] The objective function would hence be concave in either event. These initial assumptions are contrasted with those of the fixed charge problem in the diagrams below for a single plant:



1. Fixed charge problem



2. Concave cost functions

(i) Communications costs

(ii) Variable plant operating plus capital costs[7]

This would mean that we would have (in either case) a mixed integer linear programming problem, after carrying out the necessary piecewise linear approximations, leading to multiple optima.

The form of the model which follows is a modified version of the Feldman <u>et al</u>. model [15]. The assumption is continued of a single service being supplied by each plant, the same for all plants, and no inventory holdings.

---

6 Even if the capital cost element were a convex function of output, the total cost function would still be likely to be concave, as annualized capital costs are likely to be small relative to variable operating costs. For the present we follow earlier work in not showing capital costs as functionally related to output, but as a given constant which may vary between plants.

7 The functions $f_i(T_i)$ are defined on the next page.

$$\min \sum_{i,j} c_{ij}(x_{ij}) + \Sigma f_i(T_i)$$

$$\text{s.t.} \quad (1) \quad \sum_i x_{ij} = D_j \quad , \qquad j = 1, \ldots, n$$

$$(2) \quad x_{ij} \geq 0 \quad ,$$

where: $f_i(T_i) = \begin{cases} r_i T_i + w_i & \text{if } T_i \neq 0 \\ 0 & \text{otherwise} \end{cases}$

x_{ij} = the flow of services from plant i to market j

c_{ij} = unit communications cost of flow x_{ij}

D_j = demand in market j, expressed in units commensurate with the units of x_{ij}

f_i = cost function for plant i, made up of variable operating costs and installation costs

$f_i(T_i) = \Sigma(r_i T_i + w_i)$

$T_i = \sum_j x_{ij}$ = total sales of plant i

r_i = unit variable operating cost of plant i

$w_i = \dfrac{\delta}{1 - e^{-\delta L_i}} K_i$ = annual equivalent capital cost of establishing plant i

K_i = installation cost of plant i

δ = the continuous rate of interest[8]

L_i=L = estimated economic life of plant i, here assumed equal for all plants

The modifications introduced into the Feldman model consist of (i) making the communications cost function nonlinear and (ii) giving a

---

[8] Not to be confused with the zero-one variable in earlier models.

better representation of the capital cost component, which appears in the Feldman model without any indication of how its value is to be obtained. Feldman et al. developed an heuristic for solving this problem; it is described in their paper. Tests of the heuristic against mixed integer analytical solutions are presented.[9]

As noted by Feldman et al. [15], given the (assumed) concavity of the objective function and the absence of capacity constraints on plants, in the optimal solution to this problem no market will be supplied from more than one plant.

Essentially, the problem is one of striking the right balance between communications costs and plant costs (operating plus capital), which is equivalent to minimizing their sum, subject to the constraint that all demands are exactly met. The capital costs term would include the annual equivalent of some or all of the following:

acquisition cost of office building or of a long lease

acquisition cost or long lease of computing facilities

cost of acquiring the rights to reproduce data

computing costs of assembling the data at the plant

development costs in establishing initial (minimum) range of software programs, and

any other costs the benefits of which will be spread over a number of years.

1.5.5 Pricing of data bank services: a digression

Once the assumption of common prices in all markets is relaxed the problem becomes one of maximizing net receipts. There is no reason, other than convenience, of course, to suppose that each plant supplies the same single service as every other; a number of services may be

---

[9] An account of an improved heuristic procedure for solving warehouse location problems with concave costs, seen after this section was written, is given in [18]. The procedure is shown to converge rapidly to a "good" solution.

offered by each plant, their composition differing from plant to plant. This is easily dealt with by adding another subscript, k, to the x_{ij}. It is conceivable that different 'ex-works' prices might be set upon identical services by different plants, or that services which are close substitutes (say communication of the same information at different speeds from different plants) might show price differentials after allowance has been made (if it can be made) to bring the services to equivalence.

Differential pricing as between plants would introduce a bias into our problem, influencing in particular the size of the market areas to be served by each plant. It is for this reason considered to be worth investigating.

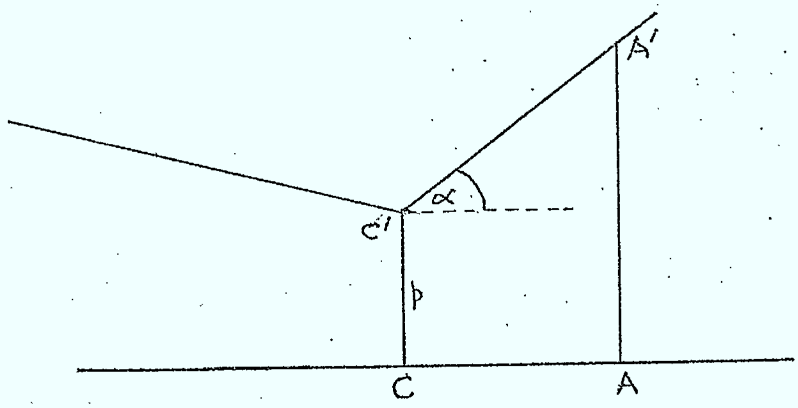
Long before the days of linear programming, the German literature contained some notable work on location of production and the delineation of the market areas of different plants. It was assumed in this work that price to the buyer consisted of the ex-works price plus transport costs; no attention was paid to speed of delivery.

A simple but useful way of analysing the problem was developed by the German economist Launhardt [17]. If p denotes the ex-works price and p_e the local price to a buyer at a distance e from the works, and transport costs are proportional to distance,

$$p_e = p + fe$$

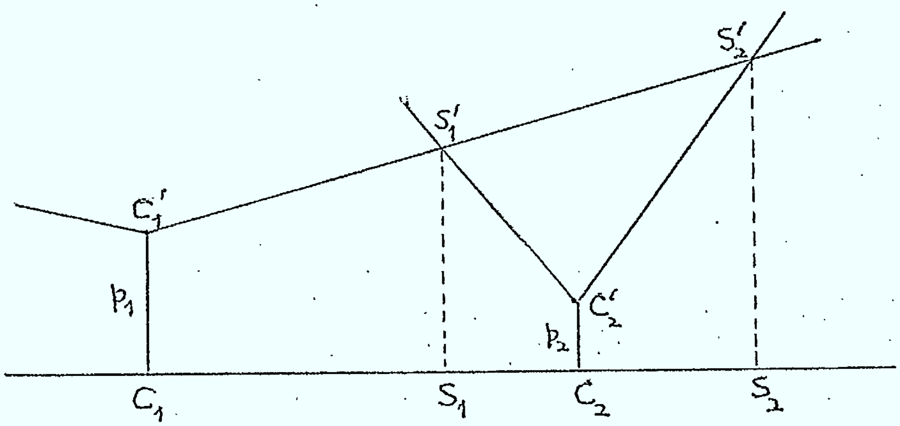
where f is the freight rate per physical unit per mile. On the assumption that deliveries go by the shortest geometrical route, all points of sale having the same p_e will lie on a circle of radius e centred on the production centre, C. If a perpendicular is erected above every point of sale, its height representing the local price, we obtain an inverted cone (known as Launhardt's funnel) with apex C' at distance p vertically above the centre of production. The slant edges of the cone which ascend in every direction from C' all have a slope of $\tan \alpha = f$.

Consider the section which results when the inverted cone is cut by a plane through CC':



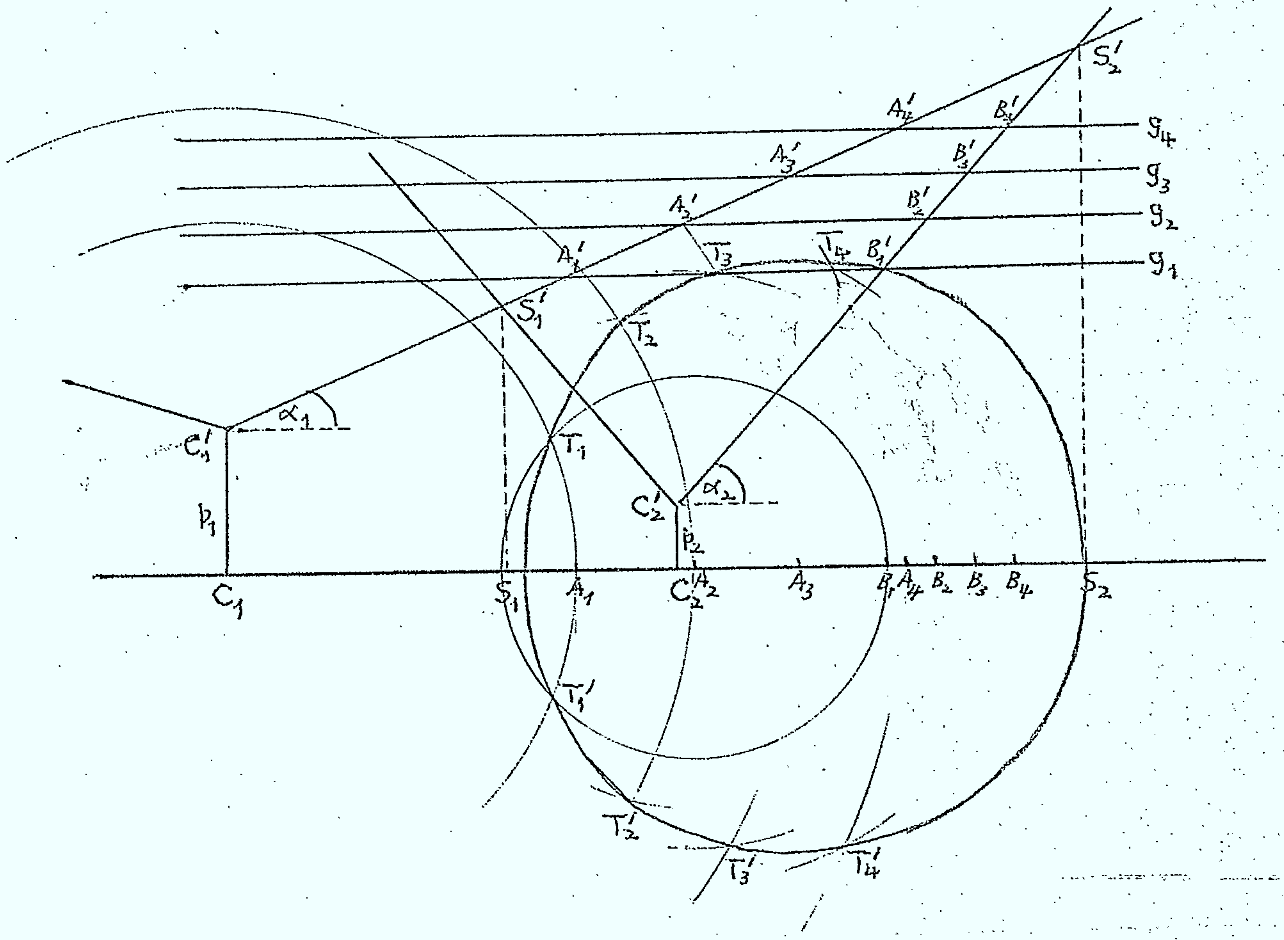
Suppose the maximum price at which all demand ceases is AA'. Then the sales area of plant C is bounded by the circle with centre C and radius CA.

Consider now <u>two</u> suppliers of goods which are substitutes (e.g. different grades of ore). Reducing them to quantities regarded as equivalent by buyers means considering different weights, and hence different freight rates. Prices p_1 and p_2 will refer to a unit of good No. 1 and the equivalent quantity of good No. 2.



Seller 1 has works at C_1 and seller 2 at C_2. It is assumed that C_1 and C_2 are sufficiently close for the sales areas to overlap. The vertical sections through the two inverted cones are shown above. All points between S_1 and S_2 belong to the sales area of C_2; those to the left of S_1 and to the right of S_2 belong to C_1. The frontier of competition (containing all points where the local prices for equivalent quantities from the two works are the same) between the two works passes through S_1 and S_2. This frontier will be the projection of the curve formed by the intersection of the two cones. To determine the entire frontier, draw a family of horizontal straight lines g_1, g_2, ... parallel to C_1C_2, cutting the cone above C_1 in A_1', A_2', ... and the cone above C_2 in B_1', B_2', ... A_1, A_2, ... and B_1, B_2, ... are the projections of A_1', A_2', ..., B_1', B_2', ..., respectively upon C_1C_2. The circles around C_1 with radii C_1A_1, C_2A_2, ... and about C_2 with radii C_2B_1, C_2B_2, ... are the loci of points at which prices of equivalent quantities from the two works are equal; i.e. $A_1A_1' = B_1B_1'$; $A_2A_2' = B_2B_2'$; ..... The points of intersection, T_1 and T_1', T_2 and T_2', ... of such pairs of circles are points on the competition frontier. Any point on this frontier satisfies the condition

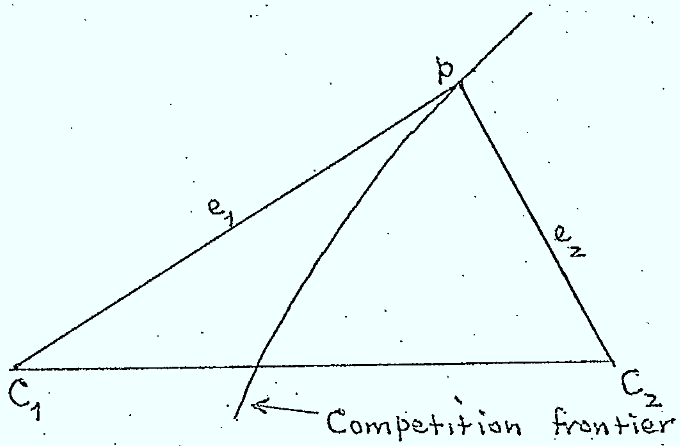
$$p_1 + f_1e_1 = p_2 + f_2e_2 .$$



We can now list possible cases:

(i) If, as in the last diagram, $p_{.} \neq p_2$, and $f_1 \neq f_2$, the competition frontier is a closed curve (in fact an ellipse of the fourth order) around the plant with the lower ex-works price.

(ii) If $p_1 \neq p_2$ (and $p_1 > p_2$), and $f_1 = f_2 = f$, then $e_2 - e_1 = \dfrac{p_1 - p_2}{f}$. The frontier is that portion of a hyperbola which is concave to the dearer plant C_1; it is no longer a closed curve:



(iii) If $p_1 = p_2 = p$ and $f_1 \neq f_2$, $e_2 : e_1 = f_1 : f_2$. The frontier is the circle which divides C_1C_2 in the proportion $f_2 : f_1$ (Appolonius' circle)

(iv) If $p_1 = p_2 = p$ and $f_1 = f_2 = f$, we have $e_1 = e_2$, and the frontier is the perpendicular bisector of C_1C_2

(v) If there are more than two plants, the sales areas of each will be polygons bounded by curve segments

(vi) Any change in relative ex-works prices or freight rates will cause a shift in the competition frontier.

REFERENCES

[1] Kochen, M. and K.W. Deutsch, Decentralization by Function and Location, Management Science, April 1973, pp. 841-856

[2] Batten, W.E., The Impact of Economics on System Design, Proceedings of ISLIC International Conference on Information Science, Vol. I, Tel Aviv, 1971

[3] Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley, 1964

[4] Hadley, G., Linear Programming, Addison-Wesley, 1962

[5] Balinski, M.L., On finding integer solutions to linear programs, Mathematica, Princeton, May 1964

[6] Kuehn, A.A. and M.J. Hamburger, A heuristic program for locating warehouses, Management Science, 9, July 1963, pp. 643-666

[7] Manne, A.S., Plant location under economies-of-scale - Decentralization and computation, Management Science, 11, November 1964, pp. 213-235

[8] Streeter, D.N., Centralization or dispersion of computing facilities, IBM Systems Journal, Vol. 12 No. 3, 1973, pp. 283-301

[9] Sharpe, W.F., The Economics of Computers, Columbia University Press, 1964

[10] Hirshleifer, J., On the Economics of Transfer Pricing, Journal of Business (Chicago), XXIX, 1956, pp. 172-184

[11] Whinston, A., Price guides in decentralized organizations, in New Perspectives in Organization Research, ed. W.W. Cooper, H.J. Leavitt and M.W. Shelly, John Wiley, 1964, pp. 405-448

[12] Amey, L.R., The Efficiency of Business Enterprises, George Allen & Unwin, London, 1969; Augustus M. Kelley, New York, 1970

[13] Amey, L.R. and D.A. Egginton, Management Accounting: A Conceptual Approach, Longman, 1973

[14] Hirsch, W.M. and G.B. Dantzig, The Fixed Charge Problem, RM-1383, The RAND Corp., 1954

[15] Feldman, E., F.A. Lehrer and T.L. Ray, Warehouse location under continuous economies of scale, Management Science, 12, May 1966, pp. 670-684

[16] Cyert, R.M. and J.G. March, A Behavioral Theory of the Firm, Prentice-Hall Inc., 1963

[17] Launhardt, W., Mathematische Begründung der Volkwirtschaftslehre, Leipzig, 1885. Reference to this work is made in Schneider, E., Pricing and Equilibrium (English version of 2nd edn. by Esra Bennathan), Allen & Unwin, 1962. A more widely quoted German contribution is Lösch, A., The Economics of Location, Yale Univ. Press, 1954 (translated from revised German edition, 1944).

[18] Khumawala, B.M. and Kelly, D.L., Warehouse location with concave costs, Infor, 12, Feb. 1974, pp. 55-65.

2. A Review of Past Approaches

   2.1 Sparks et al.: a simulation exercise

   2.2 Kochen and Deutsch: a generic model

   2.3 Streeter: the optimal number of computer installations

   2.4 Casey: the optimal allocation of files in a network

3. A data-base assignment problem and solution

   3.1 Assumptions

   3.2 The Model

   3.3 Results and Limitations

   3.4 The Computer Running Time

The question of centralization arises because of the sudden concern for consolidating files which have evolved autonomously in many computer centres. The question of dispersing arises because a particular centre has developed a file which is becoming of greater interest to remote users and computer centres.

The problem is thus a general policy problem:given a present situation of redundancy and/or unavailability of files, how to reallocate the data-bases in the best cost-effective way, or, if this is not possible, how far from the optimum is the present situation.

This chapter addresses to these questions.

2.1 Sparks et al.: A Simulation Exercise

Sparks, Chodrow and Walsh (hereafter SCW) have developed a large scale mathematical model whose purpose is to serve as a management tool (similar to PERT or CPM) for system designers, to evaluate design alternatives. The evaluation of the alternatives is made in terms of their impact on total costs, costs breakdown, and average user cost.

The basic concepts developed in SCW's model are:[1]

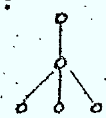
(i) the disciplines: the information is categorized into subject disciplines: mathematics, mechanical engineering, etc. . .

(ii) the information packages: information is then categorized by its form or mode of occurrence: serials, monograph, etc. . .

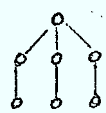
(iii) the users: they belong to users community serviced by a service centre.

(iv) the structure: the network structure is one of the key factors in the model, since this is the control variable. Three levels of decentralization typify most structures:

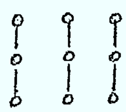
(a) centralization of acquisition and input processing



(b) centralization of acquisition processes alone



(c) decentralization



---

1 Information Dynamics Corporation, " A methodology for the Analysis of of Information Systems ", Final Report to the National Science Foundation, NSF C-370, 1965.

(v) the organization: the functional specialization is also investigated by considering three levels of specialization:

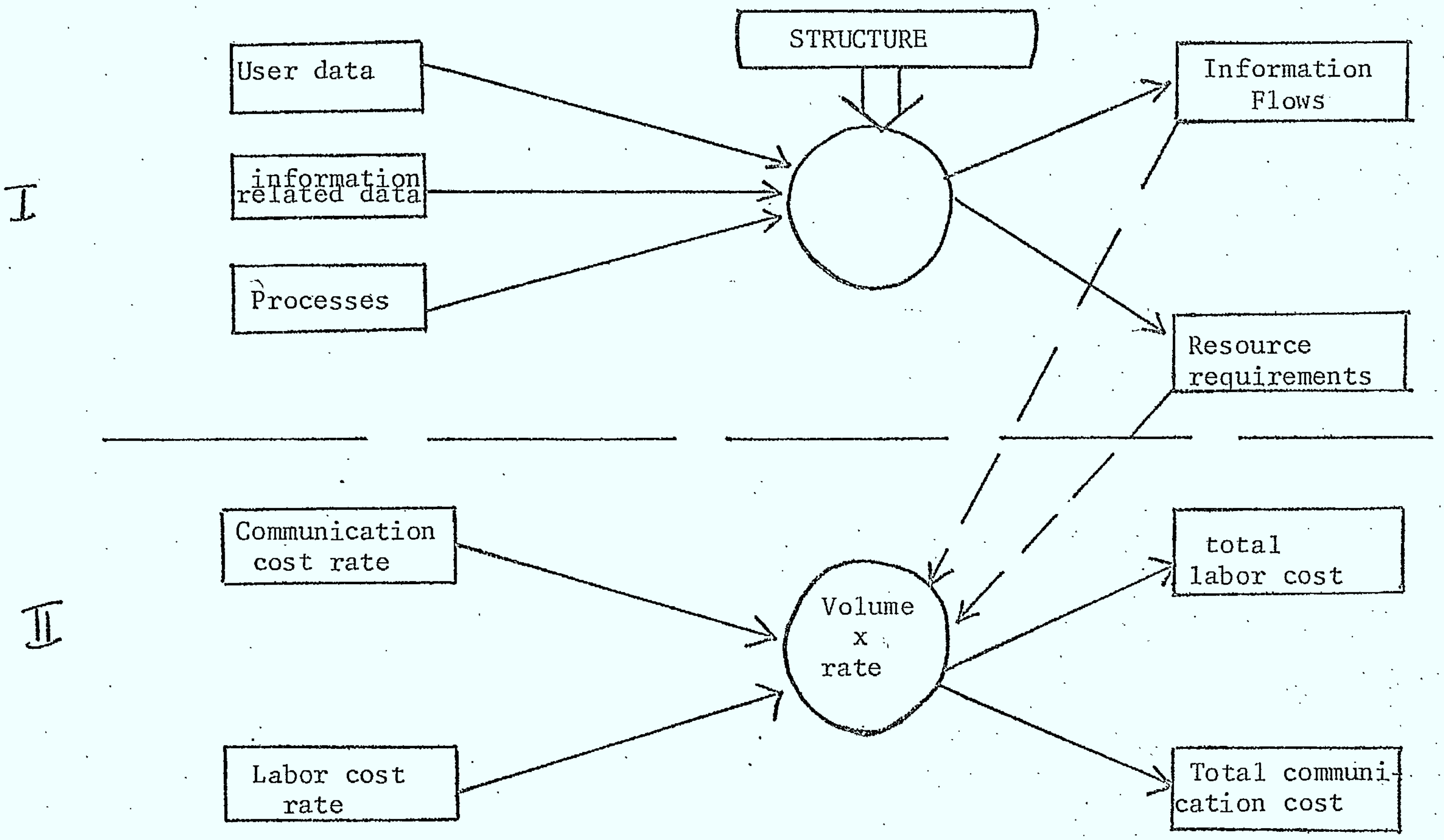
(a) discipline-oriented service centres

(b) project-oriented service centres.

(c) regional orientation

(vi) the information flows: categorized into discipline areas and physical forms, the information flow volumes crossing the system determine the resource requirements in terms of manpower, communication links and capital equipment

The methodology is essentially heuristic: the model translates the network design alternatives (the structure and the organization above) into costs in a two-stage simulation:



The user distributions, geographical and by discipline, the rates of information flows, by discipline and by form, the mathematical description of the processes of information transformation are fed into the descriptive part of the model, which puts out the network of information flows and the resource requirements according to the structure (see earlier comment) of the system. The heart of this first part of the model is a function which yields the resource requirements per volume unit of information flow. The second part of the model takes up both the information flows and the resource requirements and translates them into communication and manpower costs through a costing rate multiplication.

This model has been successfully applied to a nationwide U.S. scientific information dissemination system with the following results:

(i) the minimum cost scheme is a regionally organized (= unspecialized) system with centralized acquisition and input processing. Total costs are distributed 24% labor, 39% material, 32% communications and 4% capital equipment costs.

(ii) the maximum cost system is a discipline-oriented system (= very specialized) with total decentralization of functions. Total costs are distributed 57% labor, 26% material, only 10% communications cost and 4% capital equipment cost.

(iii) SCW make the general observation that when the service centres are very specialized, total costs vary little with the degree of decentralization; centralization of both acquisition and input processes or total decentralization makes only small differences in terms of cost.

(iv) On the other hand, in a regionally-organized system (= unspecialized), costs are very sensitive to the level of centralization and centralization seems a requisite.

(v) when the service centres are organized along a project-orientation (= intermediate specialization), the model indicates it is advantageous to decentralize all functions (acquisition, input and service)

(vi) material, communication and capital equipment costs are sensitive to user request volume; labor costs are not. Thus, the servicing activity accounts for a larger percentage of cost than acquisition and input processes. The implication is that people-oriented functions should be decentralized, while document-related functions should be centralized.

Whatever the ambitions of the model, the approach suffers from a number of shortcomings. Methodologically, the model regurgitates what was fed in: in other words, it follows the principle, "garbage in, garbage out." The validity of the model conclusions rests upon the accuracy of the data. This challenge is perhaps better understood when one is aware that more than 47,000 data items are to be fed into the model![1] A second limitation is that it performs a comparison between alternative designs: this discrete approach[2] cannot be subjected to an actual sensitivity

---

[1] This is due to the necessity of filling the coefficient's matrix.

[2] Although one can multiply the examples of design to fit a curve.

analysis of the result to changes in the model coefficients. We here come back to the previous criticism: not only the number of input data is such that a careful direct check is almost impossible, but the nature of the model prevents an indirect check by a study of the impact of changes in the coefficients. The implementation problem is obvious: collecting 47,000 data items is itself as lengthy as to perform a combinatorial analysis of the possible solutions. The problem may be slightly relieved by a reduction in the number of coefficients for smaller user communities such as the financial community, but is still a considerable task.

2.2 Kochen and Deutsch: A Generic Model

A series of papers[1] was published by Kochen and Deutsch, (hereafter K & D) in which they develop mathematical models of decentralization. Their intent is to expose a formal explication of the decentralization concept through an analytical investigation of the parameters of minimum cost configuration. In the operations research terminology, their study focuses on the warehouse allocation problem with the main concern directed towards the optimal number of warehouses.

In order to develop their model, K & D use a certain number of concepts which we expose here:

(i) distance D: it represents the east-west distance of an elongated, one-dimensional region (they assume D to be 3000 miles)

(ii) load L: the load is the volume of requests per month, originating from the strip, and uniformly distributed on the east-west distance. Each mile of the strip thus emits a request.

---

[1] "Toward a rational theory of decentralization: some implications of a mathematical approach," American Political Science Review, 63, (1969) pp. 734-749.

"Decentralization and uneven service loads," Journal of Regional Science, Vol. 10, No. 2 (August 1970) pp. 153-173.

"Decentralization by function and location," Management Science, Vol. 19, No. 8 (April 1973) pp. 841-856.

(iii) average distance: distance from originating request to nearest service station; the n service stations are assumed to be optimally located, i.e. one at each centre of the $\frac{D}{n}$ wide servicing regions. The average distance a request travels is thus $\frac{D}{2n}/2 = \frac{D}{4n}$.

(iv) communication time is the ratio of the average request information volume b in bits to the speed B of the transmission medium in bits per second.

(v) the fixed operating cost (including annualized capital cost) of each service station is C.

The tool of analysis immediately follows; the optimum number of service stations will be reached when the marginal cost of establishing a service station is equal to the marginal saving in communication cost. The total communication cost given a load L is: $c \times \frac{D}{4n} \times \frac{b}{B} \times L$ where c is the unit cost of communication in dollars per seconds per mile for a capacity B. The total cost of operating n service stations is nC. The total system cost is then: $nC + \frac{cDbL}{4nB}$. Differentiating with respect to n and setting the derivative equal to zero yields the optimal n:

$$n = \frac{1}{2}\sqrt{\frac{cDbL}{BC}}$$ [1]

---

[1] This formula is valid when the quantity under the square root is large, in order to reduce the error made in neglecting the integer value of n. The true value is

$$n = \frac{1}{2}\left(\sqrt{1 + \frac{cDbL}{BC}} - 1\right)$$

K & D can already make some conclusions: it is the relative strengths of the parameters, c, b, L, B and C, that determine the optimal configuration. There will be a higher degree of centralization when the cost of communication c decreases or when the technology increases B, the channel speed. There will be more decentralization when the average request information volume b increases, when the fixed cost of a facility decreases, or when the load L increases, as exemplified in the table below:

	small load, small communication unit cost		high load, high communication unit cost	
	low fixed cost, small channel capacity	high fixed cost, medium channel capacity	high fixed cost, large channel capacity	very high fixed cost, very large channel capacity
L	3×10^3	3×10^3	3×10^5	3×10^5
c	10^{-2}	10^{-2}	1	1
C	10^3	10^4	10^4	10^5
B	3.6×10^5 (telex)	3.6×10^6 (telephone)	3.6×10^7 (digital)	3.6×10^8 (digital)
cDLb[1]	36×10^9	36×10^9	36×10^{13}	36×10^{13}
BC	3.6×10^8	3.6×10^{10}	3.6×10^{11}	3.6×10^{13}
order of n*	10	1	30	3

[1] D is taken as being 3000 miles and b, the average request volume, is 40,000 bits (∿10,000 characters)

Further refinements are brought about in the analysis, notably concerning:

(i) the response time, which has a negative utility for the user. More facilities will be installed, to reduce the degree of utilization.

(ii) the reliability of the system: more dependable service will result from more facilities.

(iii) the number of feedback cycles between the service centre and the request location aggravates the average distance, and thus entails more facilities.

When the assumptions of uniform distribution of requests in space and time are revised in the second paper, the conclusions of the first model are qualified. The observation is made that the more uneven the spatial distribution of requests is, the relatively less dispersed the system should be. Two relationships of interest are derived: if n_0 is the optimal number of service centres in the uniform distribution case, n, the optimal number in the uneven distribution case, is related to n_0 by the following equations:

$n \cong n_0$ (1-1/8 V), where V is an index of deviation from the uniform distribution over D.

$n = n_0 \sqrt{U}$ [1], where U is the percentage of the entire region from which requests originate.

---

[1]This formula applies given restrictive assumptions on the form of the distribution (spikes of some height and width)

In contrast, fluctuations over time favor decentralization in proportion to the square root of the ratio of peak load L to average load L_0:

$$n = n_0\sqrt{\frac{L}{L_0}}$$

The relatively simple model of K & D has the merit of providing rich insights into the parameters of dispersion. As the series of papers shows, it easily accommodates more and more complex situations in a fascinating progression. The domain of applicability of their model, however, is limited[1] by its generality, and the intention to derive broad rules. Yet, this impressive work seems to have succeeded in exposing the groundrules of decentralization.

Their last paper points to the problem of definition of decentralization; according to K & D, there are four aspects in decentralization: plurality, dispersal, specialization and adaptation.[2]

They elaborate on their model by allowing another dimension than space: viz. function, which involves adjustment in the function space in the same manner as communication is involved by the geographical space.

---

[1]The authors presumably wanted to limit themselves to concepts and to simplified cases which they could get their hands on.

[2]The similarity with Sparks <u>et al</u>.: Structure and Specialization, should be noted.

2.3 Streeter: The Optimal Number of Computer Installations.

Streeter presents a paper[1] which comes very close to the problem of optimal allocation of data-banks. Streeter's model is directed towards determining the optimal degree of dispersion of computer facilities and providing general guidelines for this decision.

Some of the basic concepts developed in this paper have been outlined in the section 3.1.3.4, Part II, namely the distinction between the internal system cost (e.g. computing costs plus computer-to-computer communication cost) and user-to-system cost (mainly communication links). Dispersion of facilities essentially tends to increase internal system cost while it lowers user-to-system cost. The optimum, of course, is to be found through this trade-off. The main thrust of Streeter's analysis is to propose particular forms of cost functions for both the internal system and the user-to-system costs.

The interesting feature of his cost functions lies in the use Streeter makes of the concept of economies of scale.[2] In Chapter 3, section 1.1, the economies of scale due to indivisibilities of capacity were examined. What is alluded to by Streeter in his

---

[1] IBM Systems Journal, Vol. 12., No. 3, 1973.

[2] Which Kochen and Deutsch extensively discuss in their third paper, and which is in most cases integrated in the mathematical programming formulation of the warehouse allocation problem.

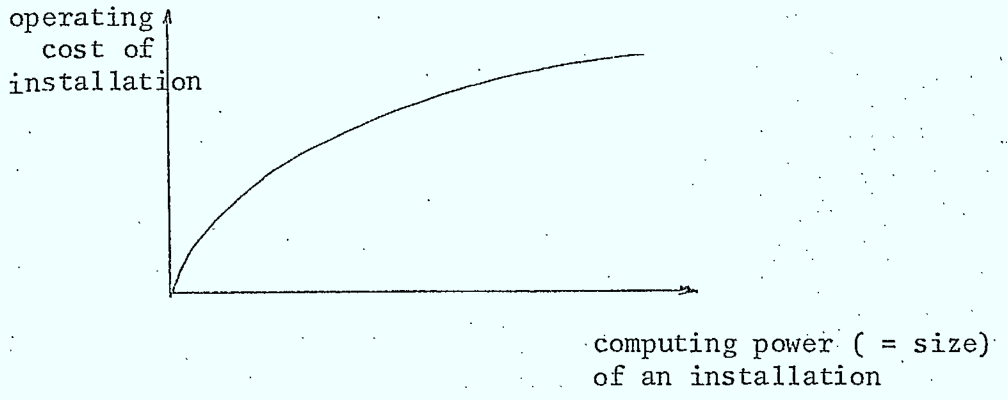
paper is a slightly different concept: his economies of scale arise because costs do not increase by as much as the scale of operation does:

(i) in the hardware cost, the well-known Grosch's law applies, which says that computing power increases as the square of the computer cost. Larger systems thus result in reduced cost per computation.

(ii) other related economies of scale appear through the supervisory software which is more effective in larger machines; because of reductions in storage duplication (file consolidation); and through better utilization of the system over time and over jobs.

(iii) in personnel costs, either systems or operating personnel, which make up an increasing proportion of the total system cost, greater efficiency is achieved by concentration of skilled manpower (synergistic effect) and centralization of program preparation.

Taking all these determinants, Streeter adopts a quadratic expression for economies of scale:



The second important contribution is the consideration of the impact of dispersion on service quality:[1] service interruption and turnaround time. In particular, Streeter uses the queuing theoretic result that a service station of capacity S is more effective (i.e. the turnaround is less) than S stations of capacity one unit, to show that, all other things being equal, turnaround time reduction calls for centralization.[2]

The model gives the total system cost as a function of the number of computer installations, and the optimum is found by setting the derivative equal to zero. Given current relative costs of manpower, computers and communication, Streeter finds that no more than three installations should be set up for a region of one thousand miles radius. Furthermore, high inter-installation costs make a unique computer preferable, while high values of user-to-system communication costs and/or high values of service interruption costs tend to favor dispersion of facilities.

It is interesting to note that the author implicitly allows for two different rates for communications: those applying to

---

[1]Similar to Kochen and Deutsch's response time and reliability.

[2]Note Kochen and Deutsch's opposite conclusion. The discrepancy arises from Streeter's assumption of smaller satellite computer, while K & D consider the service-station capacity as being unrelated to their number.

computer-to-computer links and those for user-to-computer, which may be warranted in certain types of applications, such as low volume of queries and high volume of update.

Like Kochen and Deutsch's work, Streeter's is most useful for design and planning. Unlike K & D, however, Streeter comes closer to the real problem of data-bank dispersion. Although the assumption of economies of scale may be subject to discussion, no one will deny the appropriateness of at least a rule of thumb of this order. Yet the whole analytical approach is still not quite accurate when dealing with plant location, which requires a discrete combinatorial framework (e.g. to take into account local constraints as well as overall optimum).

2.4 Casey: The Optimal Allocation of Files in a Network

An exceedingly interesting paper is presented by R.G. Casey,[1] in which the general operations research model of warehouse allocation is applied to the problem of locating copies of a file in an information network. Casey also demonstrates some mathematical properties of the optimal assignment and derives a solution procedure for his model.

One essential point in this paper is the distinction made between the update activity and the query activity,[2] which was alluded to by Streeter in his model (user-to-system vs. computer-to-computer communications). Both activities require communications, but they tend to have opposite effects on the optimal file assignment: much query activity favors the closeness of the file to the user, and thus dispersion, whereas the update activity favors the closeness of the files to each other, and, at the limit, complete centralization.

The standard expression of the objective cost function in the warehouse allocation problem is:

$$\min \sum_{j=1}^{n} \sum_{k=1}^{m} C_{jk}(\lambda_{jk}) + \sum_{k=1}^{m} \delta_k \sigma_k \qquad \text{(a)}$$

---

[1] Spring Joint Computer Conference, 1972: "Allocation of Copies of a File in an Information Network."

[2] Note the parallelism of this distinction with that applying to the file organization problem. It was remarked earlier (Part II, section 3.1.1.1.1) that the class of problem was similar.

where: $k = 1, \ldots m$ is the index for a service centre

$j = 1, \ldots n$ is the index for a region j.

λ_{jk} is the volume of requests originating from region j and addressed to the service centre k.

$C_{jk}(\lambda_{jk})$ is the communication-transmission cost function between region j and service centre k.

σ_k is the fixed operating cost of maintaining a service centre in location k ($\delta_k = 0$ or 1).

If, as a useful approximation, the cost function $C_{jk}(\lambda_{jk})$ is linear, e.g. $C_{jk}(\lambda_{jk}) = d_{jk}\lambda_{jk}$, the minimand becomes:

$$\sum_{j=1}^{n} \sum_{k=1}^{m} d_{jk}\lambda_{jk} + \sum_{k=1}^{m} \delta_k \sigma_k \qquad \text{(b)}$$

Although this is taken into account by the mathematical expression itself,[1] the minimand can be reduced to:

$$\sum_{j=1}^{n} \lambda_j \min_k \, (d_{jk}) + \sum_{k=1}^{m} \delta_k \sigma_k \qquad \text{(c)}$$

When one allows for updates to all files k from user node j, the following is obtained:[2]

---

[1] By penalizing the communication from one region j to a remote service centre.

[2] Casey assumes throughout that the update communication cost rate is the same as the query communication cost rate, e.g. d_{jk}.

$$\min_j [\Sigma\lambda_j \min_k (d_{jk}) + \Sigma_k \delta_k(\sigma_k + \sum_{j=1}^{n} \psi_j d_{jk})] \qquad (d)$$

which means that the updates originating from j must be forwarded to all files $\delta_k = 1$, $k = 1, . . . m$. This procedure is equivalent to a decentralization of acquisition and input processes.[1] If the data acquisition and input processing is centralized, then only one of the user nodes updates the files.

The bulk of Casey's paper is in fact devoted to the investigation of the mathematical properties of the model (d), and his results are presented in the form of corollaries:

(i) Corollary 1: if the update/query traffic ratio P for each region satisfied $P > \frac{1}{r-1}$ (r integer), then the optimal allocation consists of no more than r service centres (or files)

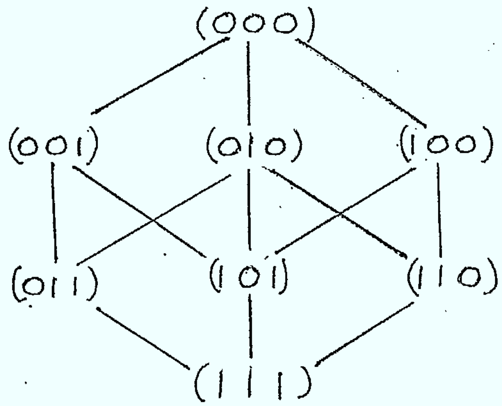
(ii) Corollary 2: if each region generates at least 50% of its traffic in updates, then no more than one service centre is warranted in the network.[2] This is a direct consequence of the preceding corollary.

---

[1] cf. Sparks et al. in section 2.1 of this Part.

[2] see footnote referring to d_{jk}.

Besides the formalization of the problem and these general properties, an important contribution of Casey's paper is the solution algorithm. Casey uses a network graph where each vertex represents a possible file assignment described by the binary vector $(\delta_1, \delta_2, \delta_k, \ldots \delta_m)$. The elements of this vector are 0 or 1, depending on whether a file is placed at location k; this vector is the solution vector of the program (d) above:



Moreover, a cost is associated with each vertex. A purely combinatorial analysis would require examining the cost associated with each vertex (e.g. 2^n) and finding the minimum cost configuration. However, a property of this particular graph is demonstrated: the sequence of costs along the path leading to the minimum cost vertex is monotonically decreasing.[1] That is, as soon as an increase in costs is encountered, the remaining portion of the path can be abandoned.

---

[1] In order to be consistent this procedure must assign an infinite cost to the null vertex (0,0,0).

Accordingly, many fewer vertices are to be examined. In layman's terms, what one investigates is the effect of adding new service centres; and of course as soon as the cost begins to increase, the optimum has been passed.

The ARPA network is then submitted to this analysis as an example, and Casey finds that, with the ratio of update to query activity as a parameter, the optimal number of files varies between one and three.

This paper has the merit of introducing a critical variable into the decentralization study: the update/query traffic ratio, which is an essential input in data-bank design. However, the analysis can be kept much simpler when the acquisition and input processes are centralized: then only one region sends updates to the satellite files and the objective function (d) collapses into:

$$\min\left[\sum_j \lambda_j \min_k (d_{jk}) + \sum_k \delta_k (\sigma_k + \psi_k)\right]$$

since in $\sum_{j=1}^{n} \psi_j d_{jk}$, all ψ_j except one are zero.

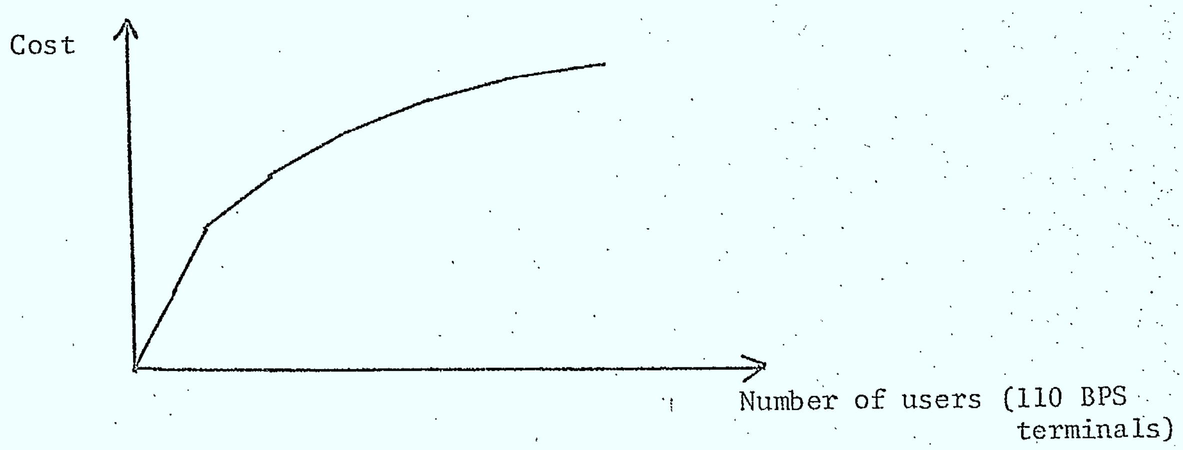
3. A Data-Bank Assignment Problem and Solution

3.1 Assumptions

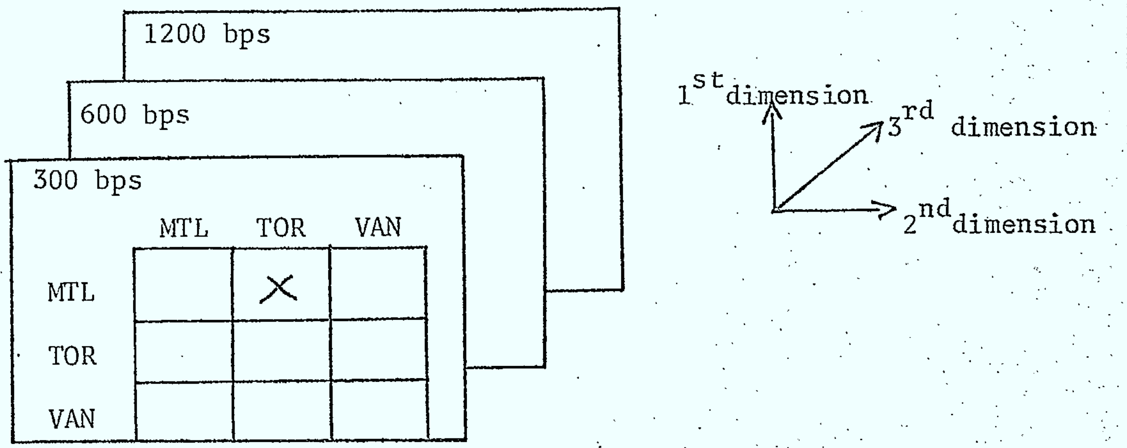
Through the first part of section 2, some material relevant to the data-bank dispersion problem was covered. However, both Kochen and Deutsch's and Casey's papers do not apply to the case where costs are not related to volume. In other words, in a situation where one deals with a dial-up, pay-as-you-use communication network, the variable of interest, λ_{jk}, is the traffic between region j and service centre k; this traffic can either be identified with the total volume of requests in bits, or with the number of requests, to which the cost is either non-linearly or proportionately related.

In the case of private lines, cost is not related to volume but to the number of lines installed between region j and service centre k. The data bank must be then viewed as a line seller, and the λ_{jk} must be identified with the number of lines (which is directly or indirectly, through queuing-theoretic considerations, related to the number of users located in the region).

Recalling the pricing structure of the Dataroute offering, in which the cost of a connection link between a region (DSA) and the computer is a piecewise linear function of the number of users:



and the way DSA-to-DSA rates are presented:



there are a number of alternative ways of expressing the objective cost function: $\min \sum_{j} \sum_{k} C_{jk} (\lambda_{jk}) + \sum_{k} \delta_k \sigma_k$ [1]

(i) the $C_{jk}(\lambda_{jk})$, the cost function of the number of end users can be fitted to the actual cost curve (which is piecewise linear) to yield an analytically convenient non-linear function.[2]

---

[1] Model (a) in section 2.4, but here the λ_{jk} are the number of terminals (or users) in region j, or equivalently, the number of lines to be installed.

[2] Probably quadratic, i.e. of the form $C_{jk}(\lambda_{jk}) = a \sqrt{\lambda_{jk}}$, "a" constant.

(ii) another way of approximating the actual cost curve is by linear fit. We then have a linear mixed-integer program, similar to Casey's model:[1] $C_{jk}(\lambda_{jk}) = d_{jk}\lambda_{jk}$. The d_{jk} would be estimated from the total cost curve or by following the third dimension in the matrix of graph 2 above.

(iii) an elaboration on this would be to take advantage of the piece-wise linear nature of the cost curve, and add constraints to make the program a piece wise linear programming problem.

Algorithms for solving each of the alternative programs outlined have been devised:[2] however, most of the time, they are very expensive to run,[3] even in the near-optimal heuristic procedures.

---

[1] Model (b) in section 2.4.

[2] W.J. Baumol and P. Wolfe, "A warehouse location problem," Operations Research, March-April 1958.

M.L. Balinski, "Integer programming: methods, uses, computation," Management Science, November 1965.

Y.J. Chuang & W.G. Smith, "A dynamic programming model for combined production, distribution and storage," Journal of Ind. Engin., Jan. 1966.

A.S. Manne, "Plant location under economies of scale," Management Science, November 1964.

M.A. Efzoymson & T.L. Roy, "A branch-bound algorithm for plant location," Operations Research, May-June, 1966.

A.A. Kuehn & M.J. Hamburger, "A heuristic program for locating warehouses," Management Science, July 1963.

Feldman, Roy and Lehrer, "Warehouse location under economies of scale," Management Science, May 1966.

[3] See, however, the paper by Khumawala and Kelly, reference [18] at the end of Chapter 1 of Part III.

This leads us to cut short the "number of users" variable and to provide the program directly with the costs. That is, the unknown now becomes the existence or the absence of a link between region j and service centre k, a binary variable x_{jk} which can only take values 1 or 0. If it is 1, either there is a communication link between j and k, and the communication cost a_{jk} of the installed line is incurred, or a data-bank is installed in region j (in this case a_{jj} is the fixed operating cost of the satellite data-bank in j). $x_{jk} = 0$ means there is no communication link between j and k, and $x_{jj} = 0$ implies that no satellite is located in j.

The assumptions leading to this model are, first, that satellite data-banks can be located in any region, with the same usage cost in all locations, and with the same operating costs regardless of the distance between the acquisition centre and the location. The rationale for these assumptions is clear: we do not want differentials in usage cost (stemming from different computer systems, loads and charging algorithms) to interfere with the optimal location decision, and we do not want the operating cost in one location (notably the update transmission cost) to be dependent on the optimal design.[1] It is felt that the error thus made is minimal due to the small proportion of update transmission cost to the total operating cost.

---

[1]Otherwise, the update transmission cost would vary according to the number and location of the data-banks, which are only determined <u>after</u> the optimization procedure has been carried out.

A second critical assumption is that the user population is given every region and all must be serviced.

It must be remembered that the model presented hereafter is a simplification of the general warehouse location model, allowed by the all-or-none kind of decision implied in the particular structure of communication tariffs. Should Dataroute become a switched, pay-as-you-use offering, it would be necessary to come back to the general model developed and exposed in section 1.5.3 of Part III.

3.2 The Model

The mathematical program is:

$$\text{minimize} \quad \sum_{j=1}^{m} \sum_{k=1}^{m} a_{jk}x_{jk}$$

subject to: (i) $\sum_j x_{jk} = 1$ for each $k = 1, \ldots m$.

(ii) $x_{jk} \leq x_{jj}$ for each $k \neq j$ and each j.

where: a_{jk} are the elements of a cost matrix (see later description)

x_{jk}, integers constrained to take the values 0 or 1.

The first type of constraint expresses the requirement that each region k must be serviced by one (and only one) service centre j. The second type of constraint is logical: a communication link from service centre j to region k must necessarily imply a data-bank in location j.

The cost matrix (a_{jk}) gives the communication costs for servicing the region k from service centre j. The diagonal elements a_{jj} give the fixed costs of operating a satellite data-bank in location j.

This cost matrix is computed by a computer program for which the data inputs are:

(i) the fixed costs[1] of operating a service centre in location j. These costs include:

- the storage cost of a duplicate copy of the entire file in the host computer.
- the computer cost for updating the copy.

---

[1]estimates of these costs for an operation similar to that of FRI are shown in Appendix 5.

- the communication cost of the update data, from the acquisition centre to the host computer. and the connection of terminals to the host computer.
- the annualized set-up cost of the duplication in the host computer storage.

(ii) the user population in every region

(iii) the Dataroute DSA-to-DSA rate matrix[1] (a three-dimensional matrix)

The (a_{jk}) cost matrix is then manually or automatically computed from the tariff structure for Dataroute.[2] Since the analysis is only concerned with the system costs which will be modified by the network design (incremental costs), the user local lines and terminal equipment cost is not included. More precisely, the communications costs only include the line cost and the Lower Speed Deriving Service cost.

The elements of the (a_{jk}) matrix constitute the coefficients of the objective function $\sum_j \sum_k a_{jk} x_{jk}$ and the solution of the linear integer problem will give values to the x_{jk}, which are to be interpreted according to the discussion of the preceding section. The computer code used for the solution of the program is the IBM package MPSX.

---

[1] Shown in Appendix 3.

[2] Exposed in Chapter 3, Section 1.2.2 of part II.

The work flow can be summarized as follows:

Input data
- Data bank operating costs
- user population
- Dataroute cost matrix

↓

Computation of the cost matrix (a_{jk})

↓

Preparation of the data deck for the MPSX code

↓

Solution to the assignment problem: MPAX output

3.3 Results and Limitations

For illustrative purposes, a sample output, together with its explanation, is shown in Appendix 6 for a reduced problem where five possible locations are considered: Montreal, Toronto, Winnipeg, Calgary, Vancouver. The method and the numeric assumptions leading to the cost estimates appear in Appendix 5.

We shall only present the general observations. The merit[1] of this method (and model) lies not so much in the particular solution it gives (as it were, the solution is a mere response to the data input) as in the possibility of a thorough sensitivity analysis: the variations in the optimal solution due to variations in the parameters, total number and distribution of data-bank users, communications costs, storage volume, storage costs, update volume.

In particular, through the parametric programming option of MPSX, it is easy to evaluate the sensitivity of the optimal solution to changes in the coefficients of the objective function, coefficients which are either the communications costs or the satellite operating costs. The basis for the changes in the coefficient values rests on the ratio communications costs/storage costs. As shown in Appendix 4, Table 4, the communications costs (expressed in ¢/bps. mile) range from .02 to .2¢/bps. mile with Dataroute. Storage costs currently

---

[1] Besides its simplicity and low cost (this last advantage is discussed in the next section).

are about .002¢/bit. Their current ratio thus varies between 10 and 100. Note that the FRI has much lower storage costs, due to a special arrangement with the McGill Computing Centre. Our numeric estimates do not reflect this arrangement.

The first solution shown in the Appendix is based on this initial ratio range: X22 takes the value 1, therefore a unique databank is desirable, and it is located in Toronto (due to the lower communication costs). The sensitivity analysis which follows this first solution is carried out for a ratio communication cost/storage cost varying between 100 and 1000[1] (a lower ratio would only confirm the optimal solution of one unique location). The table below shows the results:

Ratio[2] Communication Cost/ Storage Cost	Optimal number of facilities
10 - 100	1
200	1
300	1
400	1
1000	2

---

[1]This extreme value corresponds to a decrease in storage cost by a factor of 10. See Lynn Hopewell in "Trends in Communications," Datamation (August 1973) for the cost of mass storage (.002¢/bit in 1973, .001¢/bit in 1975, .0001¢/bit in 1978.

[2]These ratios can be interpreted as follows: ratio of 200 = decrease in storage cost by 50%; ratios of 300, 400, 1000 = decreases by 66%, 75%, 90% respectively.

Some general remarks can be made from these results. Consider for instance the output page for a communication cost/storage cost ratio of 200. The optimal solution is still one unique data-bank located in Toronto. Let us investigate the conditions for setting up a satellite in Winnipeg (X33) for example: the additional operating cost ($6008) would be offset by a decrease in communication costs due to the elimination of the link X23 from Toronto to Winnipeg ($3857) and savings on the servicing of Calgary and Vancouver (X24 is replaced by X34 and X25 by X35; savings are: -2051 + 1773 - 3426 + 3269 = -435 The balance is a net cost increase of:

$$6008 - (2937 + 435) = 2636$$

in favour of centralization. Note that the substitution of a line (X35) from Winnipeg to Vancouver to the line from Toronto to Vancouver (X25) results in savings of $157 only. These savings are small due to the fact that the end-of-line equipment cost (Dataroute Access Arrangement plus Lower Speed Deriving Service) dominate the overall communication cost figure with the consequence of a relative insensitivity to distance (see the graphs in Appendix 3). All other things being equal and notably the operating cost of a satellite, as long as the savings on distance reduction will be minimal compared to the incremental cost of a satellite, there will be a strong advantage in favour of centralization.

The relative insensitivity to distance suggests that the sensitivity to volume (e.g. the demand distribution) is a greater determinant of the structure of the optimal network. The effect of demand distribution was studied in another run of the model, not shown in the Appendix. Twelve locations (corresponding to the twelve largest DSA's of the Dataroute network) were considered. The analysis of the sensitivity to the satellite operating costs is shown on page 336. As can be noticed, only after the operating costs have dropped under $7,000 is dispersion desirable. In fact, this range of operating costs would be very close to the incremental costs the FRI would incur in case of duplication of its data-base. Note the similarity of these results with Kochen and Deutsch's theoretical study.[1] Kochen and Deutsch observe that the optimal number of satellites varies inversely with the square root of the operating cost of a satellite[2] C:

$$n \propto \frac{1}{2}\sqrt{1+\frac{1}{C}}$$

Reverting to the question of the impact of demand distribution, the more unevenly distributed the demand (all other things being equal and notably the total number of users), the more "sticky" at some values of operating costs the optimal number of satellites will be: the most populated areas will be serviced by their own facilities

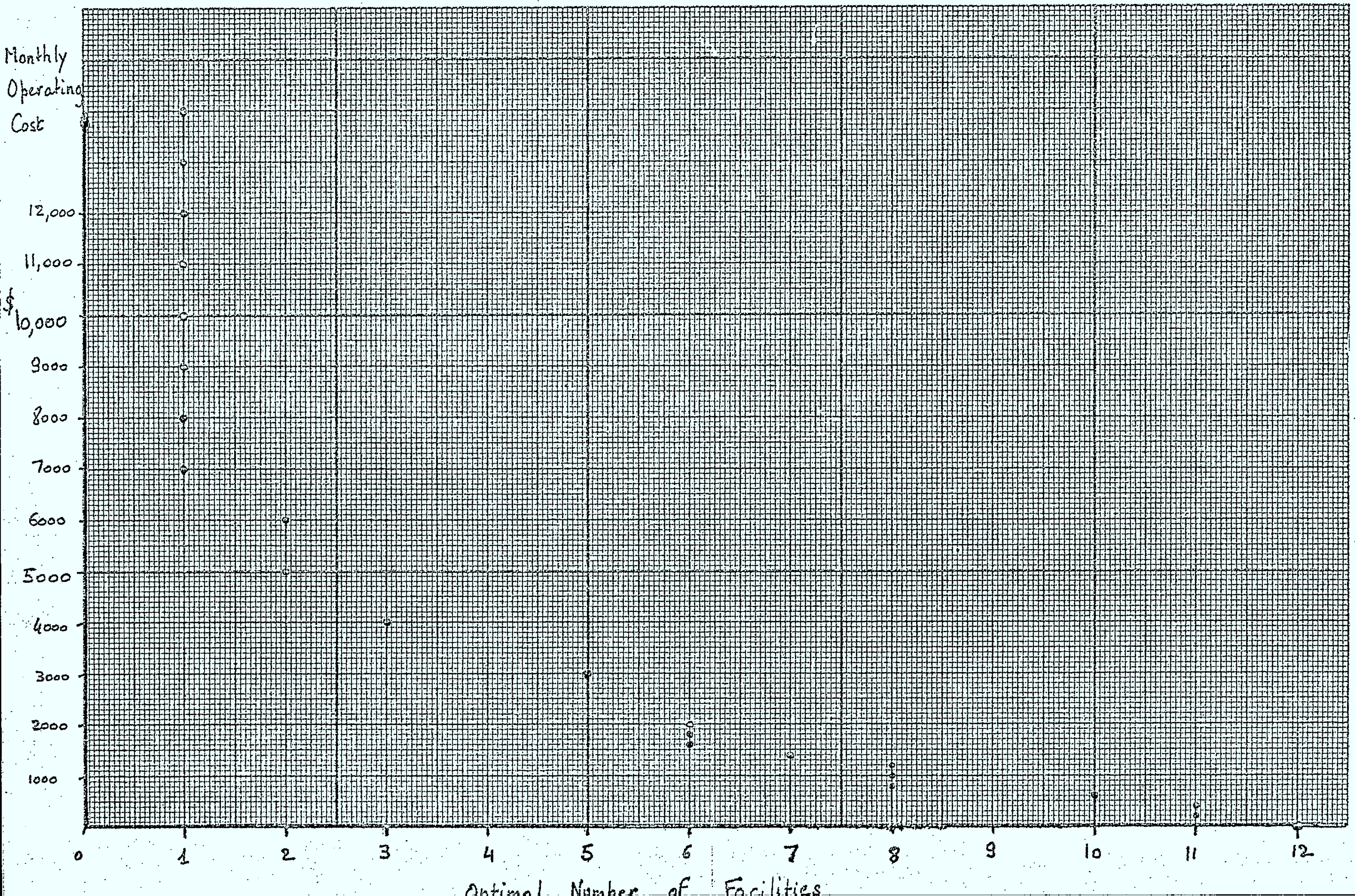
---

[1]Under the restrictive assumption of uniformly distributed demand. Minor differences with the curve of page 185 arise because of this assumption made in Kochen and Deutsch's model.

[2]See section 2.2 above.

DEGREE OF DISPERSION AND OPERATING COST

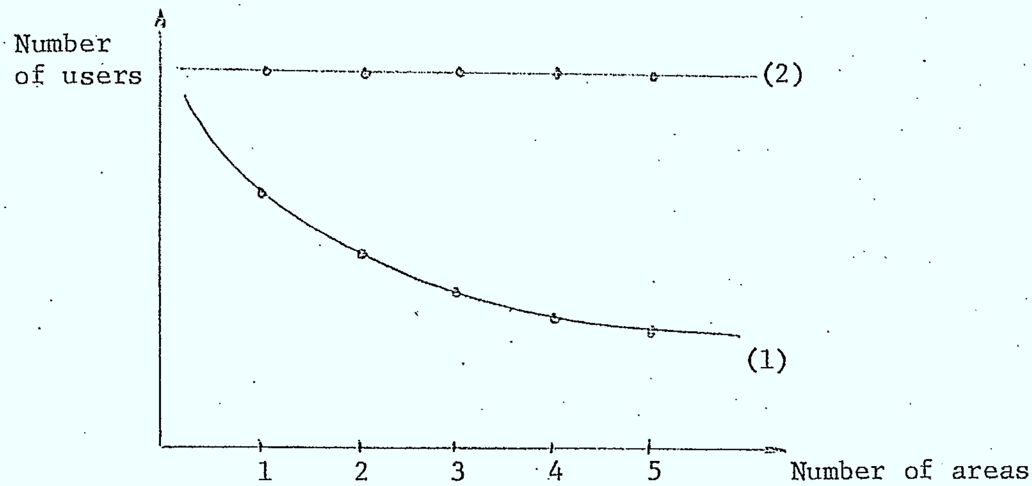
UNIVERSITÉ DE MONTRÉAL



- 336 -

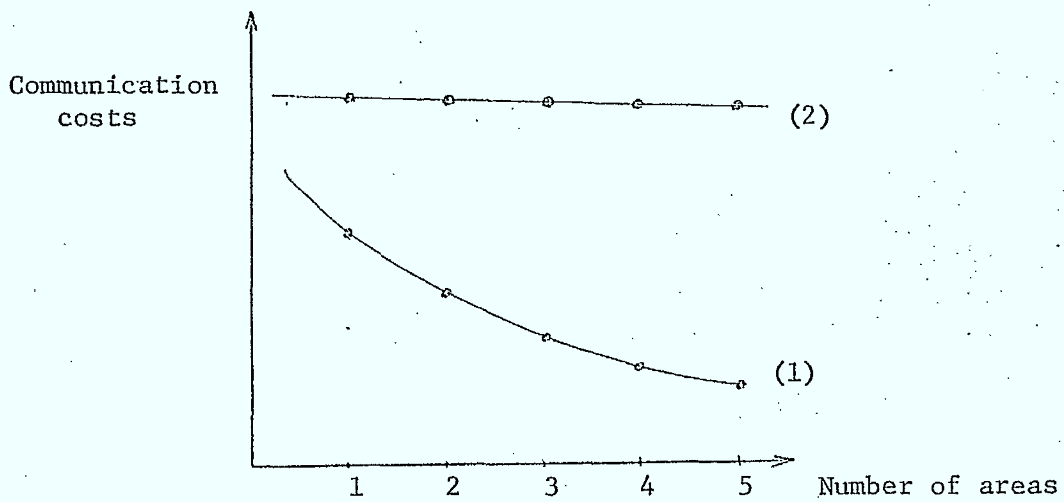
very soon as the operating cost of such a facility decreases. After skimming the most advantageous areas for installation, the operating cost will have to decrease by much for an additional installation to be desirable in a less populated area.

This can be visualized as follows: remembering that a satellite installation will be desirable whenever its operating cost is less than the savings in communication cost. Since the communication cost of servicing an area essentially depends on the number of users, a rough approximation is to make it proportional to the number of users. In the case of an uneven geographical distribution, the curve (1) obtains; in the case of an even distribution, the curve (2) obtains:[1]

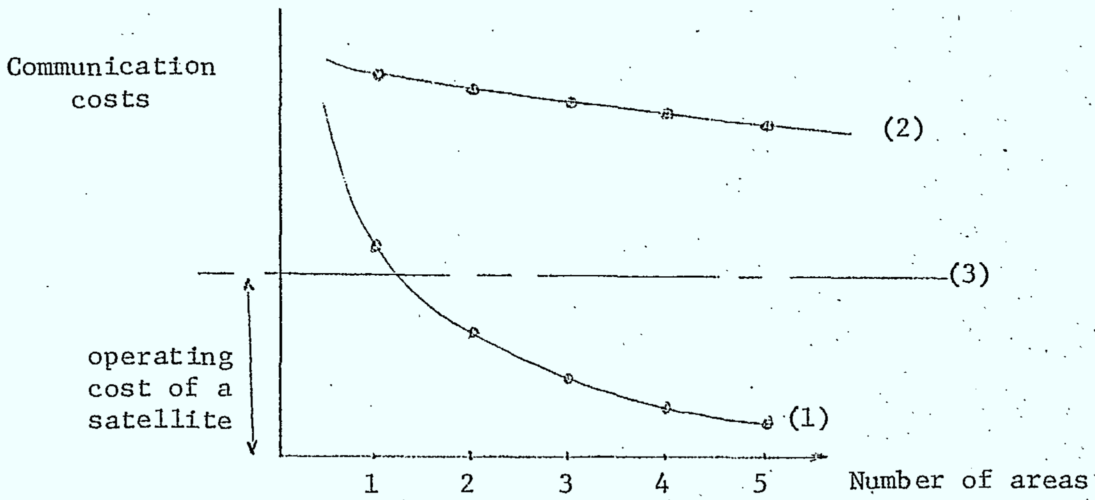


[1]With ordinates labelled in "cumulative number of users," we would get the familiar contribution-by-usage curve, see page 90.

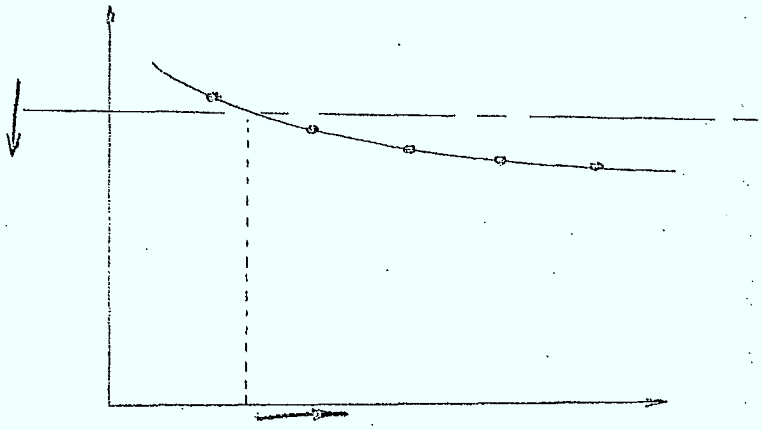
Whatever the curve shape, changing the ordinate axis of the graph from the number of users to the communication cost would operate on homothetic shift of the curve:



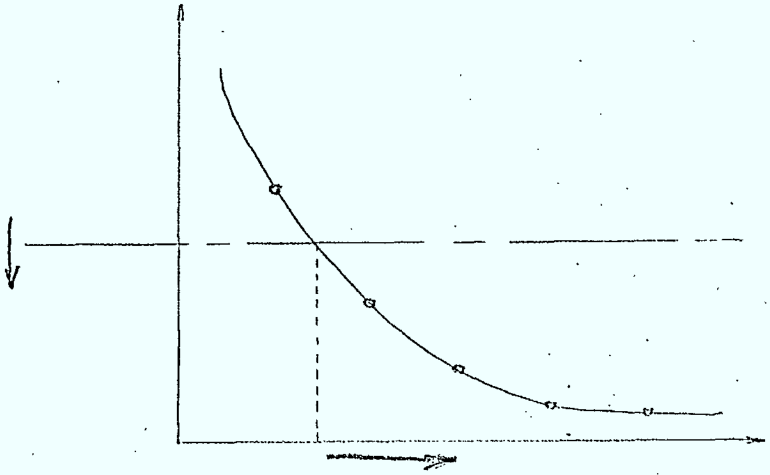
Varying the operating cost of a satellite is equivalent to having a parametric line (3) intersecting curve (1) or curve (2):



As can be seen easily, in the case of an even or near-even distribution, the optimal number of satellites is very sensitive to changes in operating costs in the relevant range, a small change being able to result in a high degree of decentralization:



On the contrary, an uneven distribution is not likely to favour decentralization until the operating cost of a satellite becomes sufficiently low to make duplication worthwhile:



This last curve is typical of the user distribution we assumed in the latter program run (see the graph of page 336).

So far, we outlined the merits of this approach, without mentioning its shortcomings and limitations:

(i) The whole formulation is based upon the current pricing structure of the two communication carriers in Canada, and particularly on the concept of private line. With the development of digital data communications, it is likely that the pricing scheme will turn to a switched pay-as-you-use network.

(ii) The model restricts itself to the consideration of one acquisition centre updating all the other satellites, which is more typical of a process of "dispersion" of one data-bank; this is in contrast with the possibility that each possible location can also update the other satellites.

(iii) The model does not allow for the servicing of one area from more than one satellite (this would destroy the all-or-none type of decision assumed in the model).

(iv) The model does not consider the queuing problem which arises when the notion of end user is more closely looked at. For our purpose, the user distribution is specified in terms of number of required channels, and not in terms of number of terminals, or even users. However, since there is a known relationship[1] between these three numbers, we feel that the loss of generality is minimal. To be more accurate in the formulation would only be a matter of inserting an additional step between the user distribution measured in number of users and the distribution measured in number of channels.

(v) The model assumes there is no midway between the complete duplication of the data-base on a local computer and the direct terminal access to the host computer. In other words, either the terminal is connected to a local computer, or it is connected to a remote computer. In some cases, notably with the advent of computer networks, this may not remain an accurate representation of reality, since computer-to-computer communications will make feasible the data transfer with various degrees of preprocessing: from a complete data-base transfer to a data-base subset transfer or a print-out file transfer. We believe the economics of computer networks have yet to be investigated in further research; however, the concepts and the method underlying the model are still applicable in the new context of computer-to-computer communications, since they revolve around the very common storage-communication tradeoff.

(vi) Administration and control problems arising in a decentralized environment are being neglected and may well overcome any other economic consideration.

---

[1]for a certain grade of service

3.4 The Computer Running Time

A noticeable feature of the program is its speed of convergence. The program running time is in fact much smaller than could be expected for an integer program with so many variables. More precisely, it can be noticed that the algorithm for solving the integer problem from the optimal continuous problem is not necessary: this means that the optimal solution of the continuous linear program is also solution of the integer linear program.

The integer program is:

$$\text{(I)} \quad \begin{cases} \min \sum\limits_j \sum\limits_k a_{jk} x_{jk} \\ \text{s.t. } x_{jk} \leq x_{jj} \text{ for all } j \text{ and all } k \neq j \\ \text{and } \sum\limits_j x_{jk} = 1 \text{ for all } k. \\ \qquad x_{jk} = 0 \text{ or } 1 \end{cases}$$

whereas the continuous program is:

$$\text{(II)} \quad \begin{cases} \min \sum\limits_j \sum\limits_k a_{jk} x_{jk} \\ \text{s.t. } x_{jk} \leq x_{jj} \text{ all } j, \text{ all } k \neq j \\ \sum\limits_j x_{jk} = 1 \text{ all } k \\ x_{jk} \geq 0 \\ x_{jk} \leq 1 \end{cases}$$

If the optimal solution of (II) is also the solution of (I), then (I) is called unimodular. Although this property has not been established yet[1], it would have an important practical implication in that the running of the program would be very cheap.

The property of unimodularity arises from the form of the constraint matrix[2]. Let the general linear programming problem be:

$$\min \quad Ax$$

$$\text{s.t.} \quad Bx = C \qquad \dim (B) = mxn$$

A basic solution is obtained by solving the set of linear equations:

$$B_B^i \, x_B = C^i$$

where B_B^i is a square submatrix of B of dimension m (the m basic variables), e.g. by inverting the matrix B_B^i to get:

$$x_B = (B_B^i)^{-1} C^i$$

In order for the elements of x_B to be integers, the elements of (B_B^i) and C^i being themselves integers (the coefficients in the constraint matrix and in the right hand side vector are -1, 0 or 1),

---

[1]Since writing this section the existence of this property has been proved by Professor R.J. Loulou of the Faculty of Management.

[2]We are grateful to Professor Loulou for pointing to the fact that the expression of the logical constraints :

$x_{jk} \leq x_{jj}$ (all j, all $k \neq i$), could have more parsimoniously been collapsed into $\sum_{k \neq j} x_{jk} \leq Nx_{jj}$ (all i, N large) but at the expense of the loss of unimodularity.

a necessary condition is that the determinants of all the matrices B_B^i be -1, 0 or +1 (recall the computation of an inverse matrix). If this is true, then the problem is unimodular, and standard linear programming codes can economically handle our assignment model.

3.5 Conclusion

In this study, a number of issues have been left aside, in particular the storage problem, which we just mentioned in passing; the possibility of computer networks, which, in my opinion, would not significantly alter the logic of the storage-computation - communication tradeoff; the queuing problem which was not dealt with at all. . . These are suggested as potential areas for further research.

With the above exceptions, the objectives of this paper were more completeness and analysis than synthesis. As a result, much effort has been devoted to the clarification of some issues which arose in the course of the analysis (notably the computer services pricing and the communication costs) at the expense of brevity and perhaps strict relevance.

As for the three main contributions of this paper, the cost of a series or a file, the activity analysis cost model, and the parametric cost model for the centralization-dispersion study, time was not available for more thorough analyses. No test was carried out on the first two models and they still remain theoretical. General statements could, however, be made on the basis of the results of the theoretical model, in particular for the desirability of the dispersion of data-banks given certain cost structures. The impact of the user distribution could not be fully investigated in this paper; yet the tool is ready, and the aim of building a parametric cost model for use in the centralization-decentralization decision has been achieved.

APPENDIX 3

<u>THE DATAROUTE LINE COST MATRIX</u>

----- 300 B.P.S. -----

- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	101									
CALGA	221	199								
CLARK	100	7	199							
EDMON	228	206	33	206						
HALIF	100	144	235	144	242					
HAMIL	123	11	199	4	226	146				
KITCH	105	15	200	14	207	149	18			
LONDO	111	31	222	23	239	153	27	10		
MONCT	83	130	230	129	237	22	132	135	140	
MONTR	49	62	210	61	217	104	65	68	75	87
OTTAW	66	45	207	44	214	117	49	52	60	102
QUEBE	22	83	216	83	223	83	86	89	96	65
REGIN	204	174	75	174	161	218	175	177	181	214
ST JO	70	120	227	119	234	39	122	124	130	16
STJOS	162	190	257	190	264	96	191	192	194	111
TORON	98	4	198	3	205	142	7	11	20	128
VANCO	238	216	75	215	102	252	216	217	219	247
WINNI	191	146	120	145	141	205	148	150	154	200

	MONTR	OTTAW	QUEBE	REGIN	ST. JO	STJOS	TORON	VANCO
OTTAW	20							
QUEBE	27	47						
REGIN	194	190	199					
ST JO	74	90	51	210				
STJOS	164	172	153	241	121			
TORON	59	41	81	172	117	189		
VANCO	227	223	233	131	244	274	215	
WINNI	171	163	182	62	197	227	143	164

- NIGHT CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	60									
CALGA	132	119								
CLARK	60	4	119							
EDMON	137	123	20	123						
HALIF	60	86	141	86	145					
HAMIL	62	6	120	3	124	88				
KITCH	63	9	120	8	124	89	11			
LONDO	67	19	121	14	125	92	16	6		
MONCT	50	78	138	78	142	13	79	81	84	
MONTR	29	37	126	37	130	62	39	41	45	52
OTTAW	40	27	124	27	128	70	29	31	36	61
QUEBE	13	50	130	50	134	50	52	54	58	39
REGIN	122	104	45	124	60	131	105	106	103	128
ST JO	42	72	136	72	140	23	73	75	78	12
STJOS	97	114	154	114	159	58	115	115	116	67
TORON	59	2	119	2	123	85	4	7	12	77
VANCO	143	129	45	129	61	151	130	130	131	148
WINNI	114	87	72	87	85	123	89	90	92	120

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	12							
QUEBE	16	28						
REGIN	116	114	120					
ST JO	45	54	31	126				
STJOS	98	103	92	144	73			
TORON	35	25	48	103	70	114		
VANCO	136	134	140	78	146	165	129	
WINNI	102	93	109	37	118	136	36	98

- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	131									
CALGA	287	258								
CLARK	130	9	258							
EDMON	296	267	43	267						
HALIF	130	187	306	187	315					
HAMIL	134	14	259	5	263	190				
KITCH	137	19	260	18	269	193	24			
LONDO	144	40	263	30	272	199	35	12		
MONCT	107	169	300	168	309	29	172	175	182	
MONTR	64	80	274	80	283	135	84	88	97	113
OTTAW	86	59	269	58	278	152	63	68	79	133
QUEBE	23	109	281	103	290	103	112	116	125	85
REGIN	265	226	97	226	131	234	228	230	235	278
ST JO	91	156	295	155	324	50	158	162	169	21
STJOS	210	247	335	247	344	125	248	249	252	145
TORON	127	5	257	4	265	184	9	14	26	166
VANCO	309	280	93	230	132	325	231	282	285	321
WINNI	245	189	156	159	194	267	192	195	200	260

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	26							
QUEBE	36	61						
REGIN	252	247	259					
ST JO	97	117	67	273				
STJOS	213	224	196	313	158			
TORON	77	54	105	224	153	246		
VANCO	295	291	303	172	317	357	279	
WINNI	222	212	237	81	256	296	136	213

----- 60M. B.P.S. -----

\- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LORDO	MONCT
BRAMP	201									
CALGA	418	391								
CLARK	200	13	391							
EDMON	427	400	67	399						
HALIF	200	288	436	287	445					
HAMIL	206	22	392	8	401	293				
KITCH	211	29	393	28	402	297	36			
LONDO	222	62	395	46	404	306	54	19		
MONCT	165	260	431	259	439	45	264	269	280	
MONTR	98	124	406	123	414	208	129	135	149	174
OTTAW	133	90	401	89	410	233	97	105	121	205
QUEBE	43	167	413	166	421	166	173	179	193	131
REGIN	398	348	149	347	202	416	351	354	362	410
ST JO	140	239	426	239	435	77	244	249	258	32
STJOS	324	380	464	380	473	192	381	382	385	222
TORON	195	7	390	6	399	283	14	22	40	255
VANCO	440	412	151	412	203	458	413	414	416	452
WINNI	381	291	240	291	282	399	296	300	308	393

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	40							
QUEBE	55	94						
REGIN	385	380	392					
ST JO	149	180	103	405				
STJOS	328	345	305	444	243			
TORON	118	83	161	345	235	378		
VANCO	427	422	434	261	447	435	411	
WINNI	341	326	364	125	389	427	287	328

\- NIGHT CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	121									
CALGA	251	234								
CLARK	122	8	234							
EDMON	256	242	40	240						
HALIF	120	173	262	172	267					
HAMIL	124	13	235	5	240	176				
KITCH	127	18	236	17	241	178	22			
LONDO	133	37	237	28	242	183	33	11		
MONCT	99	156	258	155	264	27	159	161	168	
MONTR	59	74	243	74	249	125	78	81	90	104
OTTAW	80	54	240	53	246	140	58	63	73	123
QUEBE	26	100	248	102	253	102	104	107	116	73
REGIN	239	279	89	279	121	249	211	212	217	246
ST JO	34	144	256	143	261	46	146	149	156	19
STJOS	194	228	279	228	284	115	229	229	231	133
TORON	117	4	234	4	239	172	9	13	24	153
VANCO	264	247	92	247	122	275	248	248	252	271
WINNI	228	175	144	174	169	239	178	180	185	236

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	24							
QUEBE	33	57						
REGIN	231	228	235					
ST JO	89	108	62	243				
STJOS	197	207	183	266	146			
TORON	71	50	97	207	141	227		
VANCO	256	253	260	157	268	291	247	
WINNI	225	196	219	75	233	256	172	197

\- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	261									
CALGA	544	528								
CLARK	260	17	508							
EDMON	555	519	86	519						
HALIF	261	374	567	373	579					
HAMIL	268	29	509	11	521	380				
KITCH	274	39	511	37	522	387	47			
LONDO	289	81	514	60	525	397	71	25		
MONCT	214	338	560	337	571	58	344	350	364	
MONTR	127	161	527	160	539	271	168	176	194	226
OTTAW	172	117	521	116	532	323	126	136	157	266
QUEBE	56	217	537	216	548	216	224	232	250	170
REGIN	517	452	194	452	262	540	456	460	470	533
ST JO	181	311	554	310	565	100	317	323	338	42
STJOS	421	495	604	494	615	249	496	497	500	239
TORON	254	9	507	8	518	368	19	29	52	332
VANCO	571	535	196	535	264	595	537	538	541	587
WINNI	495	379	313	378	367	518	385	391	400	511

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	51							
QUEBE	71	123						
REGIN	500	494	509					
ST JO	193	234	134	527				
STJOS	427	448	397	577	316			
TORON	153	105	209	443	345	492		
VANCO	555	543	564	342	582	531	534	
WINNI	444	424	473	162	[illegible]	555	373	426

----- 1200 B.P.S. -----

- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	432									
CALGA	701	668								
CLARK	403	27	668							
EDMON	712	679	133	678						
HALIF	401	541	723	540	733					
HAMIL	411	43	669	17	680	548				
KITCH	419	59	670	56	681	556	73			
LONDO	437	124	673	92	684	567	129	38		
MONCT	330	496	716	495	726	90	503	511	529	
MONTR	196	247	696	245	696	415	259	271	299	348
OTTAW	265	180	680	178	690	454	195	210	242	410
QUEBE	87	334	694	332	725	332	345	357	365	262
REGIN	676	617	298	616	443	698	620	624	633	691
ST JO	279	464	711	463	721	154	471	479	497	65
STJOS	588	656	756	655	767	383	657	658	661	437
TORON	391	14	667	12	677	534	29	44	30	489
VANCO	726	693	301	693	426	748	694	696	698	741
WINNI	656	546	466	545	532	678	554	561	570	670

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	79							
QUEBE	109	188						
REGIN	661	655	669					
ST JO	297	359	206	635				
STJOS	594	613	566	731	470			
TORON	236	166	322	613	457	653		
VANCO	711	705	720	499	736	762	692	
WINNI	609	591	636	249	665	711	539	593

- NIGHT CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	241									
CALGA	421	401								
CLARK	240	16	401							
EDMON	427	407	80	407						
HALIF	240	325	434	324	440					
HAMIL	247	26	401	10	403	329				
KITCH	251	35	402	34	403	333	44			
LONDO	262	74	404	55	410	340	65	23		
MONCT	198	298	429	297	436	54	302	307	317	
MONTR	118	148	411	147	418	249	155	162	179	209
OTTAW	159	128	403	107	414	273	117	126	145	246
QUEBE	52	200	417	199	423	199	207	214	231	157
REGIN	405	370	179	372	242	419	372	375	380	414
ST JO	167	273	426	278	433	93	283	287	298	39
STJOS	353	393	454	393	463	230	394	395	396	262
TORON	234	9	400	7	406	320	17	26	48	293
VANCO	436	416	181	416	244	449	417	417	419	445
WINNI	394	328	279	327	319	407	332	336	342	402

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	47							
QUEBE	69	113						
REGIN	396	393	402					
ST JO	178	216	124	411				
STJOS	356	363	340	439	282			
TORON	141	99	193	363	274	392		
VANCO	427	423	432	299	441	469	415	
WINNI	365	355	382	149	399	427	323	356

- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	523									
CALGA	911	868								
CLARK	520	35	868							
EDMON	925	892	173	892						
HALIF	521	703	939	702	953					
HAMIL	535	56	870	22	883	713				
KITCH	545	76	871	73	885	723	95			
LONDO	568	161	875	120	889	737	141	49		
MONCT	429	645	930	644	944	117	654	664	687	
MONTR	255	321	891	319	905	539	336	352	388	452
OTTAW	345	234	884	231	898	591	253	273	314	532
QUEBE	113	434	903	431	916	432	443	464	501	340
REGIN	879	802	388	801	524	907	807	811	823	898
ST JO	363	603	924	602	937	201	612	622	646	84
STJOS	764	852	983	852	957	498	854	855	859	568
TORON	505	19	867	16	881	694	38	57	104	636
VANCO	944	901	392	901	528	972	903	904	908	963
WINNI	853	710	605	709	692	881	720	729	740	872

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	103							
QUEBE	142	245						
REGIN	859	851	870					
ST JO	386	467	268	891				
STJOS	772	797	736	951	610			
TORON	307	215	419	797	594	849		
VANCO	924	917	935	563	957	1016	900	
WINNI	792	765	527	324	865	924	741	771

----- 2400 S.P.C. -----

- BUSINESS DAY CHARGES -

BRAMP	777									
CALGA	1201	1162								
CLARK	774	51	1162							
EDMON	1213	1174	255	1174						
HALIF	775	1006	1226	1024	1233					
HAMIL	795	83	1163	32	1176	1018				
KITCH	807	112	1165	109	1177	1030	140			
LONDO	836	238	1168	177	1180	1045	289	73		
MONCT	637	933	1218	931	1230	172	945	957	986	
MONTR	377	475	1183	472	1195	300	498	521	576	672
OTTAW	511	346	1176	342	1185	365	374	403	465	791
QUEBE	166	644	1193	640	1205	642	666	690	745	504
REGIN	1172	1103	575	1122	780	1197	1107	1111	1122	1139
ST JO	537	880	1212	878	1224	295	392	904	933	124
STJOS	1069	1143	1265	1147	1277	741	1149	1150	1154	337
TORON	755	28	1161	23	1173	994	55	65	153	921
VANCO	1230	1192	581	1191	765	1256	1195	1194	1198	1247
WINNI	1148	1015	983	1013	992	1175	1026	1038	1043	1165
	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT

OTTAW	152							
QUEBE	210	362						
REGIN	1154	1147	1164					
ST JO	573	694	396	1183				
STJOS	1076	1099	1044	1236	389			
TORON	453	318	622	1099	863	1145		
VANCO	1212	1206	1222	937	1241	1295	1190	
WINNI	1094	1073	1125	479	1159	1212	1003	1075
	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO

- NIGHT CHARGES -

BRAMP	466									
CALGA	720	697								
CLARK	464	31	697							
EDMON	720	705	153	704						
HALIF	465	604	736	602	743					
HAMIL	477	50	693	19	705	611				
KITCH	484	67	699	65	706	619	84			
LONDO	522	143	721	106	733	627	125	44		
MONCT	382	560	731	559	738	123	567	574	592	
MONTR	226	285	710	283	717	432	299	313	346	403
OTTAW	306	208	706	205	713	519	224	242	279	475
QUEBE	100	386	716	334	723	385	400	414	447	302
REGIN	703	662	345	661	468	713	664	667	673	713
ST JO	322	528	727	527	734	178	535	543	560	74
STJOS	642	689	759	658	766	445	689	690	692	502
TORON	453	17	696	14	724	597	33	51	92	553
VANCO	738	715	348	715	471	753	716	717	719	743
WINNI	689	629	530	603	595	704	616	623	629	699
	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT

OTTAW	91							
QUEBE	126	217						
REGIN	692	683	693					
ST JO	344	417	237	710				
STJOS	646	659	627	742	534			
TORON	272	191	373	659	521	687		
VANCO	727	723	733	552	745	777	714	
WINNI	656	644	675	237	695	727	602	645
	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO

- 24 HOUR CHARGES -

BRAMP	1010									
CALGA	1561	1511								
CLARK	1006	66	1510							
EDMON	1577	1527	332	1526						
HALIF	1007	1308	1594	1305	1610					
HAMIL	1033	103	1512	42	1528	1323				
KITCH	1049	145	1514	140	1530	1339	182			
LONDO	1087	309	1519	232	1535	1358	271	95		
MONCT	828	1213	1583	1210	1599	224	1228	1244	1282	
MONTR	490	618	1538	614	1553	1040	647	677	749	874
OTTAW	664	450	1529	444	1545	1124	486	524	604	1029
QUEBE	216	837	1551	833	1567	834	866	896	968	655
REGIN	1523	1434	747	1433	1014	1556	1439	1445	1458	1545
ST JO	699	1144	1575	1142	1591	385	1159	1176	1213	161
STJOS	1390	1492	1645	1492	1651	963	1494	1495	1508	1088
TORON	982	36	1509	32	1525	1292	72	110	199	1197
VANCO	1599	1549	755	1549	1021	1632	1551	1553	1557	1621
WINNI	1493	1319	1148	1317	1259	1525	1334	1349	1362	1515
	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT

OTTAW	197							
QUEBE	273	471						
REGIN	1500	1491	1513					
ST JO	744	[illegible]	514	1537				
STJOS	1399	1423	1359	1607	1156			
TORON	589	414	808	1428	1129	1488		
VANCO	1576	1567	1549	1218	1614	1683	1547	
WINNI	1422	1395	1463	623	1527	1575	1344	1398
	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO

----- 4800 B.P.S. -----

- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	1246									
CALGA	1839	1789								
CLARK	1241	39	1789							
EDMON	1855	1805	446	1805						
HALIF	1243	1571	1872	1568	1887					
HAMIL	1271	145	1791	56	1807	1588				
KITCH	1289	196	1793	139	1800	1605	245			
LONDO	1330	415	1797	309	1313	1626	365	123		
MONCT	1047	1467	1361	1465	1877	301	1484	1501	1543	
MONTR	656	813	1316	813	1832	1279	850	883	961	1097
OTTAW	868	604	1807	597	1323	1370	653	704	803	1267
QUEBE	291	1257	1829	1352	1845	1054	1389	1122	1200	858
REGIN	1802	1786	959	1787	1250	1834	1714	1720	1734	1523
ST JO	906	1392	1853	1390	1869	518	1409	1427	1468	217
STJOS	1661	1771	1922	1771	1938	1195	1773	1774	1779	1331
TORON	1215	48	1783	41	1803	1554	97	148	268	1450
VANCO	1877	1827	958	1327	1258	1910	1829	1331	1835	1899
WINNI	1771	1583	1397	1581	1551	1804	1600	1616	1631	1793

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	265							
QUEBE	367	632						
REGIN	1778	1769	1791					
ST JO	956	1129	691	1815				
STJOS	1670	1702	1625	1834	1406			
TORON	787	556	1026	1702	1376	1767		
VANCO	1854	1845	1867	1473	1891	1960	1826	
WINNI	1695	1666	1740	823	1786	1854	1567	1669

- NIGHT CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	748									
CALGA	1103	1074								
CLARK	745	54	1074							
EDMON	1113	1083	268	1083						
HALIF	746	943	1123	941	1132					
HAMIL	763	87	1075	34	1084	953				
KITCH	773	118	1076	113	1085	963	147			
LONDO	798	249	1078	185	1238	976	219	77		
MONCT	628	880	1117	879	1126	181	892	901	926	
MONTR	395	491	1090	488	1299	767	510	530	577	658
OTTAW	521	363	1084	358	1094	822	392	422	482	760
QUEBE	174	634	1097	631	1107	632	653	673	720	515
REGIN	1081	1025	575	1024	750	1100	1028	1032	1041	1094
ST JO	544	835	1112	834	1121	311	845	856	881	130
STJOS	996	1062	1153	1052	1163	717	1064	1065	1067	798
TORON	729	29	1073	24	1032	933	56	89	161	870
VANCO	1126	1096	531	1095	755	1146	1097	1099	1101	1139
WINNI	1063	950	838	948	930	1082	960	970	978	1076

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	159							
QUEBE	220	379						
REGIN	1067	1062	1075					
ST JO	574	677	415	1039				
STJOS	1002	1021	975	1131	843			
TORON	472	334	615	1021	825	1060		
VANCO	1112	1127	1120	884	1135	1176	1095	
WINNI	1017	999	1044	494	1071	1112	940	1001

- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	1620									
CALGA	2391	2326								
CLARK	1614	116	2326							
EDMON	2411	2347	580	2346						
HALIF	1616	2042	2433	2039	2453					
HAMIL	1652	189	2328	73	2349	2054				
KITCH	1675	255	2331	245	2351	2087	318			
LONDO	1729	540	2336	401	2357	2114	474	156		
MONCT	1361	1907	2419	1904	2440	391	1929	1952	2006	
MONTR	855	1264	2361	1257	2381	1662	1105	1148	1249	1426
OTTAW	1128	786	2359	776	2370	1781	849	915	1044	1646
QUEBE	378	1374	2378	1368	2398	1370	1415	1458	1560	1115
REGIN	2342	2220	1247	2219	1625	2384	2228	2236	2255	2370
ST JO	1178	1810	2409	1807	2430	673	1832	1855	1908	282
STJOS	2159	2302	2498	2302	2519	1553	2304	2307	2312	1730
TORON	1579	63	2324	53	2345	2021	126	192	343	1886
VANCO	2440	2376	1258	2375	1635	2482	2378	2380	2386	2469
WINNI	2303	2058	1816	2055	2016	2345	2080	2101	2120	2331

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	345							
QUEBE	477	822						
REGIN	2312	2302	2329					
ST JO	1243	1467	808	2360				
STJOS	2171	2213	2113	2450	1827			
TORON	1023	723	1333	2212	1768	2297		
VANCO	2410	2399	2427	1915	2459	2543	2373	
WINNI	2273	2165	2262	1770	2321	2410	2037	2170

----- 9600 B.P.S. -----

- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	1596									
CALGA	2352	2286								
CLARK	1589	114	2236							
EDMON	2373	2307	572	2307						
HALIF	1592	2009	2395	2026	2416					
HAMIL	1627	186	2289	72	2309	2030				
KITCH	1649	252	2291	242	2312	2053	314			
LONDO	1702	533	2296	396	2317	2079	468	164		
MONCT	1341	1377	2351	1873	2402	386	1898	1921	1973	
MONTR	844	1049	2321	1042	2342	1637	1089	1132	1231	1405
OTTAW	1112	775	2310	765	2331	1754	837	903	1030	1621
QUEBE	373	1354	2333	1346	2359	1350	1334	1437	1536	1100
REGIN	2302	2132	1229	2182	1600	2345	2189	2197	2215	2331
ST JO	1161	1782	2371	1778	2392	664	1823	1825	1878	278
STJOS	2122	2261	2462	2261	2433	1530	2263	2266	2271	1703
TORON	1556	62	2284	52	2305	1988	124	190	343	1856
VANCO	2403	2337	1240	2336	1611	2446	2339	2341	2347	2432
WINNI	2262	2025	1787	2021	1983	2305	2046	2066	2085	2291

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	340							
QUEBE	471	811						
REGIN	2271	2259	2258					
ST JO	1225	1445	836	2321				
STJOS	2134	2175	2078	2412	1798			
TORON	1009	713	1314	2174	1760	2256		
VANCO	2372	2360	2389	1885	2422	2513	2334	
WINNI	2166	2129	2222	1355	2281	2372	2083	2133

- NIGHT CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	957									
CALGA	1411	1372								
CLARK	954	69	1371							
EDMON	1424	1384	343	1384						
HALIF	955	1205	1437	1203	1450					
HAMIL	976	112	1373	43	1336	1218				
KITCH	990	151	1374	145	1337	1232	188			
LONDO	1021	320	1373	237	1390	1247	281	98		
MONCT	805	1126	1429	1124	1441	232	1139	1152	1184	
MONTR	536	629	1393	625	1405	982	653	679	739	843
OTTAW	667	465	1386	459	1398	1052	582	542	613	973
QUEBE	224	812	1403	829	1416	310	837	862	922	660
REGIN	1331	1329	737	1323	960	1407	1313	1318	1329	1399
ST JO	697	1059	1423	1057	1435	398	1032	1095	1127	167
STJOS	1273	1357	1477	1356	1490	918	1358	1359	1363	1022
TORON	933	37	1370	31	1383	1193	75	114	206	1113
VANCO	1442	1402	744	1402	966	1468	1403	1405	1408	1459
WINNI	1357	1215	1072	1213	1190	1383	1223	1240	1251	1375

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	204							
QUEBE	283	487						
REGIN	1363	1356	1373					
ST JO	735	867	532	1393				
STJOS	1280	1305	1247	1447	1079			
TORON	605	428	788	1305	1056	1354		
VANCO	1423	1416	1433	1131	1453	1508	1401	
WINNI	1299	1277	1333	633	1363	1423	1202	1282

- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	2075									
CALGA	3058	2972								
CLARK	2066	149	2971							
EDMON	3085	2999	744	2999						
HALIF	2069	2612	3114	2607	3141					
HAMIL	2115	242	2975	94	3002	2639				
KITCH	2144	327	2978	315	3005	2668	408			
LONDO	2213	693	2985	514	3012	2703	608	213		
MONCT	1744	2440	3096	2436	3123	502	2468	2497	2565	
MONTR	1097	1363	3018	1355	3045	2128	1416	1471	1631	1826
OTTAW	1446	1027	3003	995	3030	2280	1038	1173	1339	2108
QUEBE	485	1762	3240	1752	3267	1755	1813	1868	1997	1430
REGIN	2993	2336	1598	2835	2080	3049	2346	2856	2030	3031
ST JO	1510	2316	3292	2312	3110	863	2344	2373	2441	361
STJOS	2759	2939	3201	2939	3228	1989	2942	2945	2953	2214
TORON	2022	81	2969	66	2996	2584	162	247	446	2412
VANCO	3124	3036	1612	3037	2094	3180	3040	3044	3051	3161
WINNI	2940	2632	2323	2629	2579	2097	2660	2696	2710	2978

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	442							
QUEBE	612	1054						
REGIN	2953	2937	2975					
ST JO	1592	1879	1152	3217				
STJOS	2774	2027	2701	3136	2338			
TORON	1311	927	1703	2026	2288	2933		
VANCO	3033	3069	3176	2457	3148	3257	3035	
WINNI	2315	2767	2938	1372	2965	3034	2624	2773

----- 19.2K B.P.S. -----

\- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	3192									
CALGA	4704	4572								
CLARK	3179	229	4571							
EDMON	4746	4614	1145	4613						
HALIF	3133	4018	4791	4011	4833					
HAMIL	3154	373	4576	144	4618	4061				
KITCH	3299	504	4581	484	4623	4105	628			
LONDO	3404	1066	4592	791	4634	4158	935	327		
MONCT	2633	3754	4762	3747	4804	772	3796	3841	3946	
MONTR	1687	2008	4642	2035	4654	3274	2178	2263	2462	2810
OTTAW	2225	1554	4620	1530	4652	3507	1674	1805	2060	3243
QUEBE	746	2703	4677	2696	4719	2720	2789	2874	3073	2199
REGIN	4604	4363	2456	4361	3200	4691	4378	4394	4430	4562
ST JO.	2322	3563	4742	3557	4784	1323	3606	3651	3756	556
STJOS	4245	4522	4925	4521	4967	3060	4527	4532	4543	3406
TORON	3111	124	4567	135	4609	3975	249	379	687	3711
VANCO	4836	4673	2479	4672	3221	4892	4678	4682	4694	4864
WINNI	4524	4049	3574	4043	3966	4610	4092	4132	4169	4582

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	680							
QUEBE	942	1622						
REGIN	4542	4516	4577					
ST JO	2450	2891	1772	4642				
STJOS	4268	4349	4156	4825	3597			
TORON	2017	1426	2628	4348	3521	4512		
VANCO	4744	4721	4778	3769	4843	5026	4669	
WINNI	4331	4257	4444	2110	4562	4744	4007	4266

\- NIGHT CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	1915									
CALGA	2823	2743								
CLARK	1907	137	2743							
EDMON	2848	2768	687	2768						
HALIF	1910	2411	2874	2407	2902					
HAMIL	1952	224	2746	86	2771	2436				
KITCH	1979	302	2749	293	2774	2463	377			
LONDO	2043	640	2755	475	2781	2495	561	196		
MONCT	1610	2252	2857	2243	2883	463	2278	2305	2368	
MONTR	1012	1259	2785	1251	2311	1965	1307	1358	1477	1686
OTTAW	1335	932	2772	918	2797	2104	1005	1083	1236	1946
QUEBE	447	1625	2806	1617	2831	1620	1673	1724	1844	1320
REGIN	2763	2618	1475	2617	1920	2814	2627	2636	2553	2797
ST JO	1393	2138	2845	2134	2870	797	2164	2190	2254	334
STJOS	2547	2713	2955	2713	2930	1336	2715	2719	2726	2044
TORON	1867	75	2740	63	2766	2385	149	223	412	2227
VANCO	2883	2804	1483	2803	1933	2935	2807	2809	2516	2918
WINNI	2714	2430	2145	2426	2330	2766	2455	2479	2501	2749

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	408							
QUEBE	565	973						
REGIN	2725	2711	2746					
ST JO	1470	1734	1263	2785				
STJOS	2561	2610	2493	2895	2158			
TORON	1210	855	1577	2609	2112	2707		
VANCO	2846	2833	2367	2262	2926	3016	2801	
WINNI	2599	2554	2656	1266	2737	2847	2404	2559

\- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	4149									
CALGA	6116	5943								
CLARK	4133	298	5942							
EDMON	6170	5998	1488	5997						
HALIF	4138	5223	6228	5215	6283					
HAMIL	4230	485	5949	137	6004	5279				
KITCH	4289	655	5955	629	6010	5337	816			
LONDO	4426	1386	5970	1029	6025	5406	1216	425		
MONCT	3488	4950	6191	4871	6245	1003	4935	4993	5130	
MONTR	2194	2727	6035	2710	6090	4257	2832	2942	3201	3653
OTTAW	2892	2015	6005	1989	6060	4559	2177	2347	2677	4216
QUEBE	969	3521	6080	3504	6135	3510	3625	3736	3995	2359
REGIN	5986	5672	3196	5669	4160	6098	5691	5712	5759	6061
ST. JO	3019	4632	6165	4624	6219	1726	4638	4746	4883	723
STJOS	5518	5879	6402	5878	6457	3978	5885	5891	5906	4428
TORON	4044	162	5937	136	5992	5168	323	493	893	4824
VANCO	6247	6075	3223	6074	4128	6360	6291	6087	6122	6323
WINNI	5881	5264	4647	5255	5156	5993	5319	5372	5420	5956

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	884							
QUEBE	1224	2108						
REGIN	5905	5874	5957					
ST JO	3194	3758	2334	6034				
STJOS	5548	5654	5472	6272	4676			
TORON	2622	1853	3416	5654	4577	5866		
VANCO	6167	6137	6212	4900	6296	6534	6259	
WINNI	5631	5534	5777	2744	5530	5168	5209	5545

----- 50.0K B.P.S. -----

- BUSINESS DAY CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	8789									
CALGA	13327	12930								
CLARK	8751	595	12927							
EDMON	13453	13056	2975	13053						
HALIF	8763	11269	13586	11243	13712					
HAMIL	8976	969	12943	374	13069	11396				
KITCH	9111	1309	12958	1258	13084	11530	1632			
LONDO	9427	2771	12991	2057	13117	11638	2431	850		
MONCT	7262	10475	13501	10455	13627	2206	10503	10737	11053	
MONTR	4386	5507	13141	5469	13257	9837	5749	6023	6581	7544
OTTAW	5589	4029	13073	3978	13199	9736	4352	4692	5393	8943
QUEBE	1938	7339	13245	7301	13371	7313	7580	7835	8433	5312
REGIN	13027	12304	6586	12297	8814	13286	12346	12395	12505	13201
ST JO	6181	9944	13440	9384	13566	3451	12032	10166	10482	1445
STJOS	11948	12750	13959	12773	14115	8394	12794	12509	12342	9433
TORON	8547	323	12916	272	13042	11140	646	955	1735	10347
VANCO	13531	13233	6652	13231	8875	13390	13247	13251	13295	13805
WINNI	12786	11352	9937	11342	11113	13045	11490	11611	11721	12960

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	1768							
QUEBE	2448	4216						
REGIN	12841	12769	12945					
ST JO	6563	7386	4607	13140				
STJOS	12318	12262	11681	13638	10205			
TORON	5265	3706	7097	12259	9776	12751		
VANCO	13445	13377	13549	12522	13744	14292	13220	
WINNI	12208	11985	12545	5545	12899	13447	11234	12011

- NIGHT CHARGES -

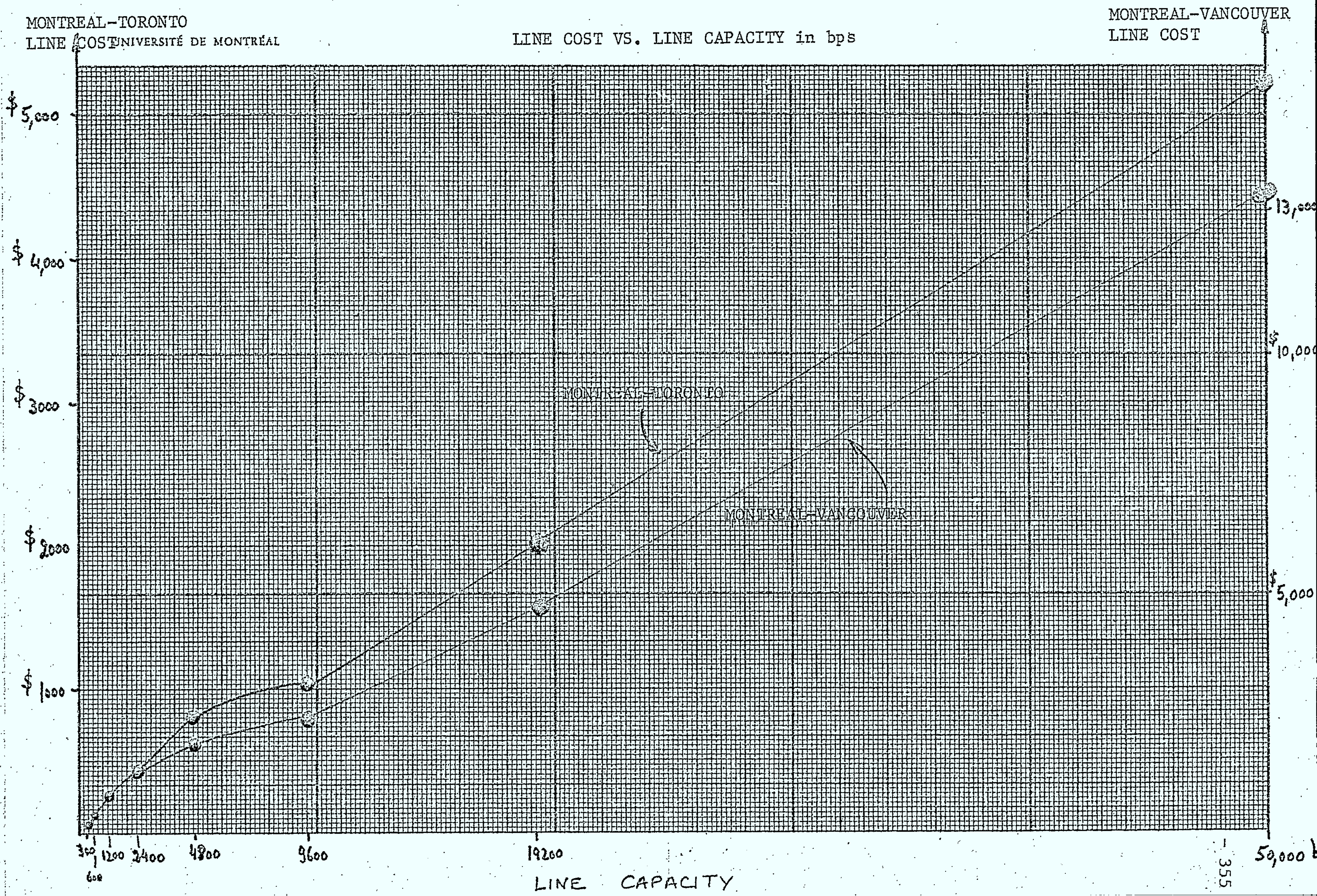
	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	5273									
CALGA	7996	7758								
CLARK	5250	357	7756							
EDMON	8072	7833	1785	7832						
HALIF	5258	6761	8152	6749	8227					
HAMIL	5386	581	7765	224	7342	6837				
KITCH	5467	785	7775	755	7850	6918	979			
LONDO	5656	1653	7795	1234	7870	7013	1459	510		
MONCT	4357	6285	8121	6273	8176	1204	6362	6442	6632	
MONTR	2632	3324	7855	3291	7960	5422	3449	3602	3961	4556
OTTAW	3533	2417	7844	2387	7919	5842	2611	2815	3256	5365
QUEBE	1163	4473	7947	4320	8023	4338	4548	4701	5250	3437
REGIN	7815	7332	3953	7378	5239	7972	7409	7437	7503	7921
ST JO	3709	5942	8364	5930	8140	2071	6019	6100	6239	867
STJOS	7169	7668	8393	7567	8469	5037	7676	7685	7705	5652
TORON	5128	194	7750	163	7825	6684	338	592	1271	5203
VANCO	8178	7940	3991	7939	5327	8334	7943	7957	7977	8283
WINNI	7671	6817	5962	6825	6668	7827	6894	6957	7433	7776

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	1051							
QUEBE	1469	2530						
REGIN	7705	7662	7767					
ST JO	3938	4731	2764	7834				
STJOS	7211	7357	7029	8213	6803			
TORON	3159	2224	4253	7356	5565	7650		
VANCO	8067	8025	8129	6313	8246	8575	7932	
WINNI	7325	7191	7527	3327	7739	8063	6741	7207

- 24 HOUR CHARGES -

	ARVID	BRAMP	CALGA	CLARK	EDMON	HALIF	HAMIL	KITCH	LONDO	MONCT
BRAMP	11425									
CALGA	17325	16808								
CLARK	11376	773	16806							
EDMON	17489	16972	3867	16969						
HALIF	11392	14648	17662	14622	17826					
HAMIL	11669	1267	16826	485	16990	14814				
KITCH	11844	1702	16845	1635	17009	14989	2122			
LONDO	12255	3602	16889	2574	17053	15195	3160	1105		
MONCT	9441	13618	17552	13591	17715	2608	13784	13958	14369	
MONTR	5702	7159	17054	7110	17247	11748	7473	7324	8581	9937
OTTAW	7655	5238	16995	5171	17153	12657	5658	6108	7010	11626
QUEBE	2519	9540	17218	9491	17332	9527	9855	12185	10962	7556
REGIN	16935	15995	8565	15936	11459	17272	16053	16113	16256	17161
ST JO	9036	12875	17472	12349	17636	4486	13741	13216	13626	1573
STJOS	15532	16615	18185	16612	18349	12913	16632	16651	16695	12263
TORON	11111	420	16791	354	16954	14462	840	1282	2320	13452
VANCO	17722	17203	8647	17201	11541	18057	17221	17240	17284	17947
WINNI	16621	14771	12919	14744	14447	15955	14937	15094	15237	16545

	MONTR	OTTAW	QUEBE	REGIN	ST JO	STJOS	TORON	VANCO
OTTAW	2299							
QUEBE	3182	5491						
REGIN	16693	16620	16828					
ST JO	8532	17251	5989	17262				
STJOS	15624	15942	15156	17705	13236			
TORON	6845	4518	9226	16937	12709	16576		
VANCO	17479	17592	17613	13679	17267	13584	17136	
WINNI	15277	15521	16309	7279	16765	17481	14625	15615



MONTREAL-TORONTO
LINE COST
UNIVERSITÉ DE MONTRÉAL
LINE COST VS. LINE CAPACITY in bps
MONTREAL-VANCOUVER
LINE COST
$ 5,000
$ 4,000
$ 3000
$ 2000
$ 1000
13,000
$ 10,000
$ 5,000
MONTREAL-TORONTO
MONTREAL-VANCOUVER
300
600
1200
2400
4800
9600
19200
50,000 b
LINE CAPACITY

APPENDIX 4

THE DATAROUTE NETWORK

- Cost per Channel

- An Aggregate Measure of Communication Cost

TABLE 1: COST PER CHANNEL

110 bps lines

MONTREAL-TORONTO POINT-TO-POINT

Number of Channels	DAA Cost	ACCESS Type	ACCESS Cost	LINE Rate	LINE Cost	Total Cost	Cost per Channel
1	2x40	DAA	-	300	59	139	
10	2x125	LSSD	640	2400	453	1343	134
20	2x125	LSSD	920	2400	453	1623	81
30	2x225	LSSD	1150	4800	787	2387	79
40	2x225	LSSD	1350	4800	787	2587	65
50	2x225	LSSD	1550	4800	787	2787	56
60	2x412	LSSD	1750	9600	1009	3583	59
70	2x412	LSSD	1950	9600	1009	3783	54
80	2x412	LSSD	2150	9600	1009	3983	50
90	2x412	LSSD	2350	9600	1009	4183	46
100	2x412	LSSD	2550	9600	1009	4383	44
110	2x500	LSSD	2750	19200	2017	5767	52
120	2x500	LSSD	2950	19200	2017	5967	50

TABLE 2: COST PER CHANNEL

300 bps lines

MONTREAL-TORONTO POINT-TO-POINT

Number of Channels	DAA Cost	ACCESS Type	ACCESS Cost	LINE Rate	LINE Cost	Total Cost	Cost per Channel
1	2x40	DAA	–	300	59	139	139
10	2x225	LSSD	640	4800	787	1877	187
20	2x412	LSSD	920	9600	1009	2753	137
30	2x412	LSSD	1150	9600	1009	2983	100
40	2x500	LSSD	1350	19200	2017	4367	109
50	2x500	LSSD	1550	19200	2017	4567	91
60	2x500	LSSD	1750	19200	2017	4767	79

TABLE 3: COST PER CHANNEL

110 bps lines

MONTREAL-VANCOUVER POINT-TO-POINT

Number of Channels	DAA Cost	ACCESS Type	ACCESS Cost	LINE Rate	LINE Cost	Total Cost	Cost Per Channel
1	2x40	DAA	-	300	227	307	307
10	2x125	LSSD	2x640	2400	1212	2002	200
20	2x125	LSSD	2x920	2400	1212	2382	119
30	2x225	LSSD	2x1150	4800	1854	3454	115
40	2x225	LSSD	2x1350	4800	1854	3654	91
50	2x225	LSSD	2x1550	4800	1854	3854	77
60	2x412	LSSD	2x1750	9600	2372	4946	82
70	2x412	LSSD	2x1950	9600	2372	5146	73
80	2x412	LSSD	2x2150	9600	2372	5346	67
90	2x412	LSSD	2x2350	9600	2372	5546	61
100	2x412	LSSD	2x2550	9600	2372	5746	57
110	2x5.00	LSSD	2x2750	19200	4600	8350	76
120	2x500	LSSD	2x2950	19200	4600	8550	71

Average Cost of a Channel

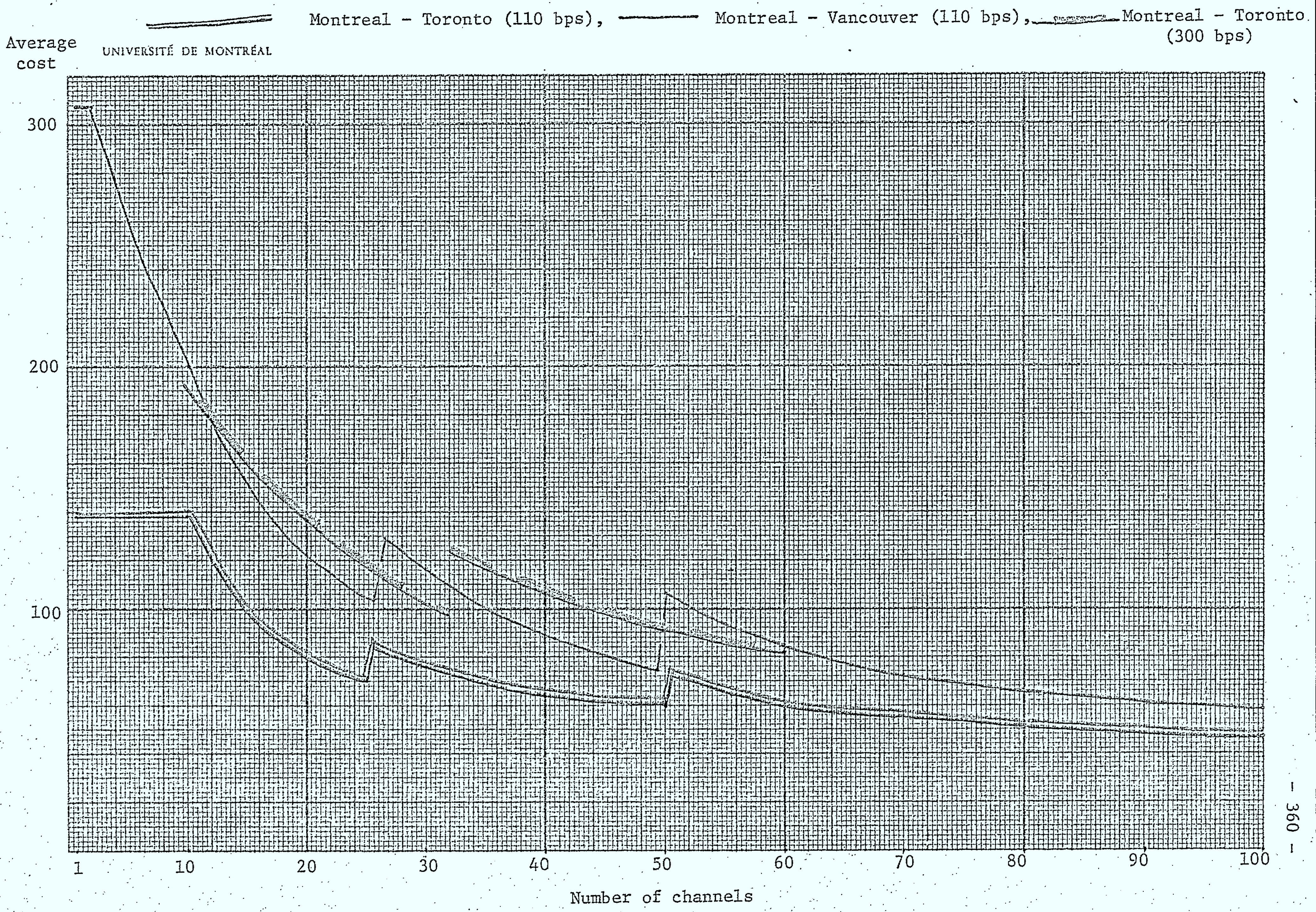


TABLE 4: The Line Cost

A useful measure of communication costs can be expressed in ¢/bps x mile:

MONTREAL TO TORONTO DATAROUTE LINE (313 miles)

	300 bps	2400 bps	4800 bps	9600 bps	19,200 bps	50,000 bps
¢/bps. mile[1]	.057	.055	.047	.030	.030	.030
¢/bps. mile[2]		0.2	0.17	0.13	0.09	

MONTREAL TO VANCOUVER DATAROUTE LINE (2302 miles)

	300 bps	2400 bps	4800 bps	9600 bps	19,200 bps
¢/bps. mile	.026	.017	.013	.0085	.0085
¢/bps. mile		.034	.027	.020	.015

RANGE OF COMMUNICATION COSTS

.023 to .3¢.bps. mile

[1]Exclusing the Lower Speed Deriving Service and the Dataroute Access Arrangement.

[2]Including the LSSD and the DAA costs at both ends (full utilization of packing capacity)

APPENDIX 5

THE ESTIMATES FOR THE $[a_{jk}]$ COST MATRIX

Computation of the Cost Matrix $[a_{jk}]$

The following calculations are based on point-to-point Dataroute lines to which are added the Dataroute Access Arrangement and the Lower Speed Deriving Service at both ends. It is assumed that the channel capacity requirement is 134.5 bps (the speed of an IBM terminal), and that the number of required channels per area is:

Montreal	40
Toronto	60
Winnipeg	20
Calgary	10
Vancouver	30

The high-speed line capacity required, the Dataroute Access Arrangement and the Lower Speed Driving Service costs are:

TABLE 1

(1) Area	(2) Line Speed	(3) DAA	(4) LSDS	(5) Total Equipment Cost 2 x (3) + 2 x (4)
Montreal	9600 bps	$412	$1350	2174
Toronto	9600 bps	412	1750	2574
Winnipeg	4800 bps	225	920	1370
Calgary	2400 bps	125	640	890
Vancouver	4800 bps	225	1150	1600

The line cost is given by the table below (according to the line speed).

TABLE 2

from:	TO:	Montreal	Toronto	Winnipeg	Calgary	Vancouver
	Montreal		1009	1695	1183	1854
	Toronto	1009		1567	1161	1826
from:	Winnipeg	2166	2003		883	1669
	Calgary	2321	2284	1397		968
	Vancouver	2372	2334	1669	581	

Adding the transpose of the vector (5) of Table 1 to each row of Table 2, we obtain the communication cost matrix:

TABLE 3

	Montreal	Toronto	Winnipeg	Calgary	Vancouver
Montreal		3583	3065	2073	3454
Toronto	3183		2937	2051	3426
Winnipeg	4340	4577		1773	3269
Calgary	4495	4858	2767		2568
Vancouver	4546	4904	3039	1471	

As indicated in Part III, Section 3, the diagonal represents the monthly operating cost of a data-bank located in the area. The following are estimates for a data-bank similar in activity and scope to the FRI, and are taken from the data in section 2.4 of Part II.

(i) initial set-up cost: \$10,000; annualized on a ten-year basis: \$150/mo.

(ii) contingencies: \$650/mo.

(iii) storage cost: 11,000 tracks[1] @ 80¢/track/mo.: \$8800

(v) update communication cost: 4800 bps Dataroute line (night):

Montreal to	Toronto	Winnipeg	Calgary	Vancouver
Cost	\$472	\$1017	\$1090	\$1112

Final cost matrix $[a_{jk}]$:

TABLE 4

	Montreal	Toronto	Winnipeg	Calgary	Vancouver
Montreal	11000	3583	3065	2073	3454
Toronto	3183	11472	2937	2051	3426
Winnipeg	4340	4577	12017	1773	3269
Calgary	4495	4858	2767	12090	2568
Vancouver	4546	4904	3039	1471	12112

[1] 10,000 tracks are equivalent to a 100 million byte. The reader will note that the high storage cost rate: \$.80/track/mo. is an unnecessary assumption. From section 2.3.1.1, it appears that this costing rate is much higher than the rental cost of approximately 20¢.track.mo. (including a share of the control unit cost).

It is important to note why it is nto necessary to include items such as terminal, modem and local line costs in the analysis. These are not incremental to the decition of whether or not to locate a satellite in the city under consideration. These costs are unaffected by the decision (terminals, modems and local lines will always be there) and should therefore not enter the equation.[1]

Recalling the list of the elements of a computer-communication system supporting a data-bank (section 3.1.1.1, Part II):

(i) the terminals

(ii) the modems

(iii) the local lines

(iv) the line control unit

(v) the CPU

(vi) the direct access storage devices

(vii) the storage control units.

To this must be added the long-distance communications line and equipment:

(viii) the long-distance transmission equipment.

When the duplication of the data-bank is considered, the categories (i), (ii) and (iii) above are uneffected. The tradeoff bears on categories (vi) and (vii) and (viii), in that dispersal will increase the total storage requirements and decrease the long-distance communication's requirements. What about categories (iv) and (v)?

---

[1]This is more rigorously expressed in section 1.4 of Part III.

Dispersal will decrease the load placed on each computer system. In this sense, the costs associated with categories (iv) and (v) are incremental with respect to the decision. Our assumptions, however, have the effect of eliminating these problems, by supposing that the transfer of a part of the usage load from one computer system to another would not cause any serious concern. This is supported on two grounds: first, the load placed on the computer system by the data-bank users is negligible; second, agreements between the computing centre and the data-bank management would smooth some difficult cost allocation problems: the line control unit cost, for example, would be offset by the benefits of a wider sharing of the computer system cost.

APPENDIX 6

THE COMPUTER OUTPUT

- Structure of the Constraint Matrix
- The Data Deck
- The Optimization Results
- The Sensitivity Analysis

THE STRUCTURE OF THE EXPANDED CONSTRAINT MATRIX

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	RHS
-1	1																								≤0
-1		1																							≤0
-1			1																						≤0
-1				1																					≤0
					1	-1																			≤0
						-1	1																		≤0
						-1		1																	≤0
						-1			1																≤0
										1		-1													≤0
											1	-1													≤0
												-1	1												≤0
												-1		1											≤0
															1			-1							≤0
																1		-1							≤0
																	1	-1							≤0
																		-1	1						≤0
																				1				-1	≤0
																					1			-1	≤0
																						1		-1	≤0
																							1	-1	≤0
1					1					1					1					1					=1
	1					1					1					1					1				=1
		1					1					1					1					1			=1
			1					1					1					1					1		=1
				1					1					1					1					1	=1

```
.MPSX-PTF16.    EXECUTOR.   MPSX   RELEASE 1   MOD LEVEL 4

NAME            INTEX
ROWS
 N   COST
 N   NEW
 L   A1
 L   B1
 L   C1
 L   D1
 L   E1
 L   F1
 L   G1
 L   H1
 L   I1
 L   J1
 L   K1
 L   L1
 L   M1
 L   N1
 L   O1
 L   P1
 L   Q1
 L   R1
 L   S1
 L   T1
 E   A2
 E   B2
 E   C2
 E   D2
 E   E2
COLUMNS
     START      'MARKER'                      'INTORG'
     X11        COST        11000.00000       NEW         -11000.00000
     X11        A1        -     1.00000       B1        -      1.00000
     X11        C1        -     1.00000       D1        -      1.00000
     X11        A2              1.00000
     X12        COST         3583.00000       A1                 1.00000
     X12        B2              1.00000
     X13        COST         3065.00000       B1                 1.00000
     X13        C2              1.00000
     X14        COST         2073.00000       C1                 1.00000
     X14        D2              1.00000
     X15        COST         3454.00000       D1                 1.00000
     X15        E2              1.00000
     X21        COST         3183.00000       E1                 1.00000
     X21        A2              1.00000
     X22        COST        11472.00000       NEW         -11472.00000
     X22        E1        -     1.00000       F1        -      1.00000
     X22        G1        -     1.00000       H1        -      1.00000
     X22        B2              1.00000
     X23        COST         2937.00000       F1                 1.00000
     X23        C2              1.00000
     X24        COST         2051.00000       G1                 1.00000
     X24        D2              1.00000
     X25        COST         3426.00000       H1                 1.00000
```

INPUT DATA DECK (1)

```
.MPSX-PTF16.    EXECUTOR.  MPSX   RELEASE  1   MOD LEVEL  4

     X25        E2                 1.00000
     X31        COST            4340.00000     I1                  1.00000
     X31        A2                 1.00000
     X32        COST            4577.00000     J1                  1.00000
     X32        B2                 1.00000
     X33        COST           12017.00000     NEW            -12017.00000
     X33        I1             -   1.00000     J1             -    1.00000
     X33        K1             -   1.00000     L1             -    1.00000
     X33        C2                 1.00000
     X34        COST            1773.00000     K1                  1.00000
     X34        D2                 1.00000
     X35        COST            3269.00000     L1                  1.00000
     X35        E2                 1.00000
     X41        COST            4495.00000     M1                  1.00000
     X41        A2                 1.00000
     X42        COST            4858.00000     N1                  1.00000
     X42        B2                 1.00000
     X43        COST            2767.00000     O1                  1.00000
     X43        C2                 1.00000
     X44        COST           12090.00000     NEW            -12090.00000
     X44        M1             -   1.00000     N1             -    1.00000
     X44        O1             -   1.00000     P1             -    1.00000
     X44        D2                 1.00000
     X45        COST            2568.00000     P1                  1.00000
     X45        E2                 1.00000
     X51        COST            4546.00000     Q1                  1.00000
     X51        A2                 1.00000
     X52        COST            4904.00000     R1                  1.00000
     X52        B2                 1.00000
     X53        COST            3039.00000     S1                  1.00000
     X53        C2                 1.00000
     X54        COST            1471.00000     T1                  1.00000
     X54        D2                 1.00000
     X55        COST           12112.00000     NEW            -12112.00000
     X55        Q1             -   1.00000     R1             -    1.00000
     X55        S1             -   1.00000     T1             -    1.00000
     X55        E2                 1.00000
     END        'MARKER'                       'INTEND'
RHS
     RH1        A2                 1.00000     B2                  1.00000
     RH1        C2                 1.00000     D2                  1.00000
     RH1        E2                 1.00000
BOUNDS
 UP  INTBND     X11                1.00000
 UP  INTBND     X12                1.00000
 UP  INTBND     X13                1.00000
 UP  INTBND     X14                1.00000
 UP  INTBND     X15                1.00000
 UP  INTBND     X21                1.00000
 UP  INTBND     X22                1.00000
 UP  INTBND     X23                1.00000
 UP  INTBND     X24                1.00000
 UP  INTBND     X25                1.00000
 UP  INTBND     X31                1.00000
```

INPUT DATA DECK (2)

JOJU


```
.MPSX-PTF14.   EXECUTOR.  MPSX  RELEASE 1  MOD LEVEL 4

 UP INTRND     X33          1.00000
 UP INTRND     X34          1.00000
 UP INTRND     X35          1.00000
 UP INTRND     X41          1.00000
 UP INTRND     X42          1.00000
 UP INTRND     X43          1.00000
 UP INTRND     X44          1.00000
 UP INTRND     X45          1.00000
 UP INTRND     X51          1.00000
 UP INTRND     X52          1.00000
 UP INTRND     X53          1.00000
 UP INTRND     X54          1.00000
 UP INTRND     X55          1.00000
ENDATA
```

INPUT DATA DECK (3)

INPUT DATA DECK (4)

```
MPSX-PTF13     CONTROL PROGRAM COMPILER.  MPSX  RELEASE 1  MOD LEVEL 3

        0001                    PROGRAM
        0002                    INITIALZ
        0094                    MOVE(XPBNAME,'SAMPLE')
        0095                    MOVE(XDATA,'INTEX')
        0096                    CONVERT('SUMMARY')
        0097                    BCDOUT
        0098                    SETUP('BOUND','INTBND','MIN')
        0099                    MOVE(XOBJ,'COST')
        0100                    MOVE(XRHS,'RH1')
        0101                    PRIMAL
        0102                    SOLUTION
        0103                    MOVE(XCHROW,'NEW')
        0104                    XPARAM=0.0
        0105                    XPARMAX=0.5
        0106                    XPARDELT=.10
        0107                    PARAOBJ
        0108                    SOLUTION
        0109          *
        0110                    OPTIMIX('COST',0.,0,0,1)
        0206                    EXIT
        0207                    PEND
```

TION 1 - ROWS

MBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
1	COST	BS	23069.00000	23069.00000-	NONE	NONE	1.00000
2	NEW	BS	11472.00000-	11472.00000	NONE	NONE	.
3	A1	UL	.	.	NONE	.	1321.00000
4	B1	BS	.	.	NONE	.	.
5	C1	BS	.	.	NONE	.	.
6	D1	UL	.	.	NONE	.	6390.00000
7	E1	BS	.	.	NONE	.	.
8	F1	UL	.	.	NONE	.	128.00000
9	G1	UL	.	.	NONE	.	22.00000
10	H1	UL	.	.	NONE	.	6418.00000
11	I1	BS	.	.	NONE	.	.
12	J1	UL	.	.	NONE	.	2077.00000
13	K1	UL	.	.	NONE	.	300.00000
14	L1	UL	.	.	NONE	.	6575.00000
15	M1	BS	.	.	NONE	.	.
16	N1	UL	.	.	NONE	.	2443.00000
17	O1	UL	.	.	NONE	.	298.00000
18	P1	UL	.	.	NONE	.	7276.00000
19	Q1	BS	.	.	NONE	.	.
20	R1	BS	.	.	NONE	.	.
21	S1	UL	.	.	NONE	.	1666.00000
22	T1	UL	.	.	NONE	.	602.00000
23	A2	EQ	1.00000	.	1.00000	1.00000	3289.00000-
24	B2	EQ	1.00000	.	1.00000	1.00000	4904.00000-
25	C2	EQ	1.00000	.	1.00000	1.00000	3065.00000-
26	D2	EQ	1.00000	.	1.00000	1.00000	2873.00000-
27	E2	EQ	1.00000	.	1.00000	1.00000	9844.00000-

ORIGINAL PROBLEM COMMUNICATION COST/STORAGE COST

RATIO IS BETWEEN 10 AND 100

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
28	X11	BS	.	11000.00000	.	1.00000	.
29	X12	BS	.	3583.00000	.	1.00000	.
30	X13	BS	.	3065.00000	.	1.00000	.
31	X14	BS	.	2073.00000	.	1.00000	.
32	X15	BS	.	3454.00000	.	1.00000	.
33	X21	UL	1.00000	3183.00000	.	1.00000	106.00000-
34	X22	BS	1.00000	11472.00000	.	1.00000	.
35	X23	BS	1.00000	2937.00000	.	1.00000	.
36	X24	BS	1.00000	2051.00000	.	1.00000	.
37	X25	BS	1.00000	3426.00000	.	1.00000	.
38	X31	LL	.	4340.00000	.	1.00000	1051.00000
39	X32	LL	.	4577.00000	.	1.00000	1750.00000
40	X33	BS	.	12017.00000	.	1.00000	.
41	X34	BS	.	1773.00000	.	1.00000	.
42	X35	BS	.	3269.00000	.	1.00000	.
43	X41	LL	.	4495.00000	.	1.00000	1206.00000
44	X42	LL	.	4858.00000	.	1.00000	2397.00000
45	X43	BS	.	2767.00000	.	1.00000	.
46	X44	BS	.	12090.00000	.	1.00000	.
47	X45	BS	.	2568.00000	.	1.00000	.
48	X51	LL	.	4546.00000	.	1.00000	1257.00000
49	X52	BS	.	4994.00000	.	1.00000	.
50	X53	LL	.	3039.00000	.	1.00000	1640.00000
51	X54	BS	.	1471.00000	.	1.00000	.
52	X55	BS	.	12112.00000	.	1.00000	.

ORIGINAL PROBLEM (END)

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
28	X11	BS	.	5500.00197	.	1.00000	.
29	X12	BS	.	3583.00000	.	1.00000	.
30	X13	BS	.	3065.00000	.	1.00000	.
31	X14	BS	.	2073.00000	.	1.00000	.
32	X15	BS	.	3454.00000	.	1.00000	.
33	X21	UL	1.00000	3183.00000	.	1.00000	341.99992-
34	X22	BS	1.00000	5736.00205	.	1.00000	.
35	X23	BS	1.00000	2937.00000	.	1.00000	.
36	X24	BS	1.00000	2051.00000	.	1.00000	.
37	X25	BS	1.00000	3426.00000	.	1.00000	.
38	X31	LL	.	4340.00000	.	1.00000	815.00008
39	X32	LL	.	4577.00000	.	1.00000	1477.50010
40	X33	BS	.	6008.50215	.	1.00000	.
41	X34	BS	.	1773.00000	.	1.00000	.
42	X35	BS	.	3269.00000	.	1.00000	.
43	X41	LL	.	4495.00000	.	1.00000	970.00008
44	X42	LL	.	4858.00000	.	1.00000	2688.00011
45	X43	BS	.	2767.00000	.	1.00000	.
46	X44	BS	.	6045.00216	.	1.00000	.
47	X45	BS	.	2568.00000	.	1.00000	.
48	X51	LL	.	4546.00000	.	1.00000	1021.00008
49	X52	BS	.	4904.00000	.	1.00000	.
50	X53	LL	.	3039.00000	.	1.00000	1320.00011
51	X54	BS	.	1471.00000	.	1.00000	.
52	X55	BS	.	6056.00217	.	1.00000	.

COMMUNICATION COST/STORAGE COST RATIO IS 300

ECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
28	X11	BS	.	4400.00236	.	1.00000	.
29	X12	BS	.	3583.00000	.	1.00000	.
30	X13	BS	.	3065.00000	.	1.00000	.
31	X14	BS	.	2073.00000	.	1.00000	.
32	X15	BS	.	3454.00000	.	1.00000	.
33	X21	UL	1.00000	3183.00000	.	1.00000	389.19990-
34	X22	BS	1.00000	4588.80246	.	1.00000	.
35	X23	BS	1.00000	2937.00000	.	1.00000	.
36	X24	BS	1.00000	2051.00000	.	1.00000	.
37	X25	BS	1.00000	3426.00000	.	1.00000	.
38	X31	LL	.	4340.00000	.	1.00000	767.80010
39	X32	LL	.	4577.00000	.	1.00000	1423.00012
40	X33	BS	.	4806.80258	.	1.00000	.
41	X34	BS	.	1773.00000	.	1.00000	.
42	X35	BS	.	3269.00000	.	1.00000	.
43	X41	LL	.	4495.00000	.	1.00000	922.80010
44	X42	LL	.	4858.00000	.	1.00000	2026.20013
45	X43	BS	.	2767.00000	.	1.00000	.
46	X44	BS	.	4836.00259	.	1.00000	.
47	X45	BS	.	2568.00000	.	1.00000	.
48	X51	LL	.	4546.00000	.	1.00000	973.80010
49	X52	LL	.	4904.00000	.	1.00000	493.19754
50	X53	LL	.	3039.00000	.	1.00000	762.80260
51	X54	BS	.	1471.00000	.	1.00000	.
52	X55	BS	.	4844.80260	.	1.00000	.

COMMUNICATION COST/STORAGE COST RATIO IS 400

SECTION 1 - ROWS

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
1	COST	BS	31175.00000	31175.00000-	NONE	NONE	1.00000
2	NEW	BS	23584.00000-	23584.00000	NONE	NONE	.70000
3	A1	BS	.	.	NONE	.	.
4	B1	BS	.	.	NONE	.	.
5	C1	BS	.	.	NONE	.	.
6	D1	BS	.	.	NONE	.	.
7	E1	BS	.	.	NONE	.	.
8	F1	BS	.	.	NONE	.	.
9	G1	BS	1.00000-	1.00000	NONE	.	.
10	H1	BS	1.00000-	1.00000	NONE	.	.
11	I1	BS	.	.	NONE	.	.
12	J1	UL	.	.	NONE	.	390.10301
13	K1	UL	.	.	NONE	.	278.00000
14	L1	BS	.	.	NONE	.	.
15	M1	BS	.	.	NONE	.	.
16	N1	UL	.	.	NONE	.	920.39999
17	O1	UL	.	.	NONE	.	170.00000
18	P1	UL	.	.	NONE	.	485.60303
19	Q1	BS	1.00000-	1.00000	NONE	.	.
20	R1	BS	1.00000-	1.00000	NONE	.	.
21	S1	BS	1.00000-	1.00000	NONE	.	.
22	T1	UL	.	.	NONE	.	580.00000
23	A2	EQ	1.00000	.	1.00000	1.00000	3300.00275-
24	B2	EQ	1.00000	.	1.00000	1.00000	3441.60287-
25	C2	EQ	1.00000	.	1.00000	1.00000	2937.00000-
26	D2	EQ	1.00000	.	1.00000	1.00000	2051.00000-
27	E2	EQ	1.00000	.	1.00000	1.00000	3053.60303-

COMMUNICATION COST/STORAGE COST RATIO IS 1000

SECTION 2 - COLUMNS

NUMBER	•COLUMN•	AT	•••ACTIVITY•••	••INPUT COST••	••LOWER LIMIT•	••UPPER LIMIT•	•REDUCED COST•
28	X11	BS	•	3300.00275	•	1.00000	•
29	X12	LL	•	3583.00000	•	1.00000	141.39713
30	X13	LL	•	3065.00000	•	1.00000	128.00000
31	X14	LL	•	2073.00000	•	1.00000	22.00000
32	X15	LL	•	3454.00000	•	1.00000	400.39697
33	X21	UL	1.00000	3183.00000	•	1.00000	117.00275-
34	X22	BS	1.00000	3441.60287	•	1.00000	•
35	X23	BS	1.00000	2937.00000	•	1.00000	•
36	X24	BS	•	2051.00000	•	1.00000	•
37	X25	LL	•	3426.00000	•	1.00000	372.39697
38	X31	LL	•	4340.00000	•	1.00000	1039.99725
39	X32	LL	•	4577.00000	•	1.00000	1525.50014
40	X33	BS	•	3605.10301	•	1.00000	•
41	X34	BS	•	1773.00000	•	1.00000	•
42	X35	LL	•	3269.00000	•	1.00000	215.39697
43	X41	LL	•	4495.00000	•	1.00000	1194.99725
44	X42	LL	•	4858.00000	•	1.00000	2336.79712
45	X43	BS	•	2767.00000	•	1.00000	•
46	X44	BS	•	3627.00303	•	1.00000	•
47	X45	BS	•	2568.00000	•	1.00000	•
48	X51	LL	•	4546.00000	•	1.00000	1245.99725
49	X52	LL	•	4904.00000	•	1.00000	1462.39713
50	X53	LL	•	3039.00000	•	1.00000	102.00000
51	X54	BS	1.00000	1471.00000	•	1.00000	•
52	X55	BS	1.00000	3633.60303	•	1.00000	•

RATIO 1000 (END)

SECTION 1 - ROWS

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
1	COST	BS	48072.00000	48072.00000-	NONE	NONE	1.00000
2	NEW	BS	46601.00000-	46601.00000	NONE	NONE	.80000
3	A1	BS	1.00000-	1.00000	NONE	.	.
4	B1	BS	1.00000-	1.00000	NONE	.	.
5	C1	BS	1.00000-	1.00000	NONE	.	.
6	D1	BS	1.00000-	1.00000	NONE	.	.
7	E1	BS	1.00000-	1.00000	NONE	.	.
8	F1	BS	1.00000-	1.00000	NONE	.	.
9	G1	BS	1.00000-	1.00000	NONE	.	.
10	H1	BS	1.00000-	1.00000	NONE	.	.
11	I1	BS	1.00000-	1.00000	NONE	.	.
12	J1	BS	1.00000-	1.00000	NONE	.	.
13	K1	BS	1.00000-	1.00000	NONE	.	.
14	L1	BS	1.00000-	1.00000	NONE	.	.
15	M1	BS	.	.	NONE	.	.
16	N1	UL	.	.	NONE	.	475.00346
17	O1	UL	.	.	NONE	.	170.00000
18	P1	BS	.	.	NONE	.	.
19	Q1	BS	1.00000-	1.00000	NONE	.	.
20	R1	BS	1.00000-	1.00000	NONE	.	.
21	S1	BS	1.00000-	1.00000	NONE	.	.
22	T1	UL	.	.	NONE	.	302.00000
23	A2	EQ	1.00000	.	1.00000	1.00000	3183.00000-
24	B2	EQ	1.00000	.	1.00000	1.00000	2294.40328-
25	C2	EQ	1.00000	.	1.00000	1.00000	2937.00000-
26	D2	EQ	1.00000	.	1.00000	1.00000	1773.00000-
27	E2	EQ	1.00000	.	1.00000	1.00000	2120.40347-

COMMUNICATION COST/STORAGE COST RATIO IS ∞

(STORAGE COST PROPER IS NIL)

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
28	X11	UL	1.00000	2200.00315	.	1.00000	982.99685-
29	X12	LL	.	3583.00000	.	1.00000	1288.59672
30	X13	LL	.	3065.00000	.	1.00000	128.00000
31	X14	LL	.	2073.00000	.	1.00000	300.00000
32	X15	LL	.	3454.00000	.	1.00000	1333.59653
33	X21	BS	.	3183.00000	.	1.00000	.
34	X22	BS	1.00000	2294.40328	.	1.00000	.
35	X23	BS	.	2937.00000	.	1.00000	.
36	X24	LL	.	2051.00000	.	1.00000	278.00000
37	X25	LL	.	3426.00000	.	1.00000	1305.59653
38	X31	LL	.	4340.00000	.	1.00000	1157.00000
39	X32	LL	.	4577.00000	.	1.00000	2282.59672
40	X33	UL	1.00000	2403.40344	.	1.00000	533.59656-
41	X34	BS	.	1773.00000	.	1.00000	.
42	X35	LL	.	3269.00000	.	1.00000	1148.59653
43	X41	LL	.	4495.00000	.	1.00000	1312.00000
44	X42	LL	.	4858.00000	.	1.00000	3038.60018
45	X43	BS	.	2767.00000	.	1.00000	.
46	X44	BS	.	2418.00346	.	1.00000	.
47	X45	LL	.	2568.00000	.	1.00000	447.59653
48	X51	LL	.	4546.00000	.	1.00000	1363.00000
49	X52	LL	.	4904.00000	.	1.00000	2609.59672
50	X53	LL	.	3039.00000	.	1.00000	162.00000
51	X54	BS	1.00000	1471.00000	.	1.00000	.
52	X55	BS	1.00000	2422.40347	.	1.00000	.

RATIO ∞ (END)

Interpretation of the Computer Output

(i) In Section 1 - Rows - and Section 2 - Columns - the two-character symbol (under the column heading: AT) denotes the status that the constraint (in the row section) or the variable (in the column section) has in the optimal solution:

- BS: in the basis and feasible
- EQ: equality satisfied
- UL: non-basis, constraint binding or variable at upper limit.
- LL: non-basis, constraint binding or variable at lower limit.

(ii) Activity: in the row section, it is the value the objective function COST takes in the solution; in the column section, it is the value the variables x_{jk} take in the solution.

(iii) Slack activity (row section): this is the amount of unexpended potential that is still in the constraint (e.g. the constraint is not binding)

(iv) Input cost (column section): the coefficient of the variable X_{jk} in the objective function, e.g. the cost of using this particular facility as opposed to selecting another one.

(v) Lower limit and upper limit: this is the lowest or highest value that the variable of the constraint and remain feasible.

(vi) Dual activity (row section): it is the rate of increase (resp. decrease) in the value of the objective function per unit increase (resp. decrease) in the constraint limit.

(vii) Reduced cost (column section): it represents the rate of increase (resp. decrease) in the objective function value per unit increase (resp. decrease) in the variable.

The interpretation of these last two items is not a simple matter, because duality interpretation can only be done with continuous variables.[1] Besides the theoretical difficulty, the shodow prices of our "logical" constraints $x_{jk} \leq x_{jj}$ are devoid of any economic interpretation; the same remark applies to the equality constraints

$$\sum_j x_{jk} = 1.$$

[1] A discussion of the significance of the integer programming dual variables can be found in H. Martin Weingartner: "Mathematical Programming and the Analysis of Capital Budgeting Problems," Markham Publishing Co., Chicago 1967. Chapter 5 "Imputation of dual variables," is of particular interest.

PART IV: DATA STORAGE CONSIDERATIONS.

1.1 Introduction

The objective of this chapter is to develop techniques of theoretical analysis of databank costs. Given the details of databank implementation we derive expressions for the cost in computer time and storage space of using and maintaining the databank. Our attention is restricted to computer operations with an emphasis on operations involving peripheral devices. We do not consider costs such as salaries (programming, data preparation), overhead, etc.

This chapter provides a theoretical bridge from the basic costs of peripheral device access and peripheral device storage to the final time-dependent cost of using and maintaining the databank. We shall call these two different levels of costing the user level, or user interface, and the data level, or data interface. Two categories of unknowns separate these two levels: usage-dependent and organization-dependent.

The usage-dependent unknowns can be characterized as rates and distributions. For instance, the costs of acquiring and storing new data depend on the rate of growth of the databank; cost of retrieval depends on the rate at which requests are made. The request rate is particularly difficult to analyze because a "request" can vary from a simple inquiry (e.g. involving the price of a single stock for a single day) to a demand for elaborate combinations of data (e.g. as required by a portfolio management program). Thus requests must be classified in various ways, and the final cost depends on the distributions of requests among the various categories.

The theoretical analysis of this chapter cannot supply empirical information about these rates and distributions. However, it isolates the relevant rates and distributions, to guide investigators analyzing particular databanks, and it includes them in the expressions for databank costs.

Furthermore, the analysis shows how to simplify and interrelate some of the distributions and rates.

The organization-dependent unknowns are much more difficult to characterize. The behaviour of a databank is totally dependent on the organization of the data it contains. For instance, the simplest request may require only one peripheral device access or it may require many, depending on the organization. Classification of the many techniques of databank organization is beyond the scope of this chapter. Instead, we develop and apply our analysis considering a particular databank. This databank, maintained at McGill by the Financial Research Institute (F.R.I.) and recording the daily activity of four North American stock exchanges, is selected for its accessibility and its importance. It is subjected to a high rate of inquiry, has a significant growth rate and is organized accordingly.

The analysis of this chapter thus dwells on a limited number of file structures to the exclusion of other equally important organizations. The structures we consider, however, are widely used, and the analysis presented can be applied without difficulty to some of the other organizations.

1.2 Elements of the Analysis

The analysis to follow identifies a number of general concepts as important elements in assessing databank costs. Some can be precisely defined in terms of mathematical notation; others can be specified only vaguely in general and cannot be made precise until considered in the context of a particular file organization.

1.2.1 Costs

1.2.1.1 User Interface

We break the cost of computer operations and storage into four major subdivisions.

Acquisition cost, C_a. The cost of adding new data to databank.

Storage cost, C_s. The cost of storing the databank.

Retrieval cost, C_r. The cost of requests made of the databank.

Maintenance cost, C_m. The cost of periodic reorganizations of the databank.

Each of these costs is a function of time, and the total cost involves the sum of all four.

1.2.1.2 Data Interface

At the data level, there are two categories of costs:

Operational, C_α: the cost per access to data on peripheral devices.

C_γ: the cost of spooling 1000 cards

C_τ: the cost per second of computation.

Storage, C_σ: the cost per unit time of storing a unit of data.

The operational cost could be developed in terms of time involved - §1.2.5.1.1 shows that the transformation between time and number-of-accesses can be made simply - but the analysis of this chapter will adopt cost per access as fundamental.

The operational cost can be further analyzed:

$$C_\alpha = \mu C_\iota$$

where C_ι is the actual charge for an access or "I/O request" and μ is the effective charge ratio,

$$\mu = \begin{cases} 1 & \text{if the peripheral unit is online} \\ 1 + \frac{C\mu}{\beta C_\iota} & \text{if the peripheral unit must be mounted} \end{cases}$$

This ratio takes into account the mounting charge, C_μ since if there are β blocks per volume, a volume must be mounted after every β I/O requests. The charge C_I will usually vary in an installation depending on time of day, priority, etc.

The storage cost will also vary, depending on whether the file is stored offline, online or with backup. When the file is stored on tape, there may be no charge per unit time, the only charge being the original cost of the tape. This latter possibility has not been included in the analysis: such a case would best be included by estimating C_σ as a rate of amortization.

1.2.2 Growth

The number of blocks, n(t), occupied by the databank is a function of its age, t. (We do not define this quantity directly in terms of rates, but the rate of growth of the databank is the derivative, n'(t), of this function with respect to t.)

A block, or access unit, is the unit of peripheral storage which is read or written in a single access. It must be distinguished from the storage unit, in terms of which storage costs are computed: it is preferable, but not always possible to make these the same size. Usually there are sufficiently many blocks, $n(t)$, that $n(t)$ may be considered a continuous function. The acquisition and storage costs can be related directly to $n(t)$:

$$C_a \propto n(t)$$

for ideal cases, and $$C_s \propto \int_0^t n(t')\, dt'$$

The factors of proportionality depend on the data organization.

For instance, with a linearly growing databank,

$$n(t) = n_o + gt$$

where g is the rate of growth, and, ideally,

$$C_a \propto n_o + gt$$

where the part of the acquisition cost proportional to n_o is due to the initial acquisition of data, and the part proportional to gt is due to the continued acquisition of data at rate g.

Alternatively, with a static databank,

$$n(t) = n_o$$

and

$$C_s \propto n_o t$$

since the data initially acquired must be stored for the lifetime of the databank.

The retrieval and maintenance costs also depend on $n(t)$, but weakly and not nearly as directly.

1.2.3 Requests

We can similarly define a function, $r(t)$, describing the request rate at any time, t. However, $r(t)$ is not as easily characterized in general as $n(t)$ because of the variety of types of requests.

1.2.3.1 User Interface

We classify requests according to their logical structure, as single requests and various types of multiple requests.

Single request. A logically atomic request, usually but not necessarily involving the retrieval of a single block from the databank. Eg. a request for the price of a single stock for a single day.

Simultaneous or batch request. A request consisting of r independent single requests, which may be presented in any order. Eg. a request for prices of several stocks for several days.

A distribution, s_r, is associated with simultaneous requests: s_r is the proportion of all simultaneous requests involving r single requests.

All the above characterizations are necessarily loose: they can only be specified unambiguously for particular databank organizations.

Usage distribution. All requests are associated with an important distribution, u, the usage distribution. This gives the distributuion of requests over the whole databank, and is essential for economic design of a databank. If some parts of the databank are heavily used and other parts lightly used, it may be desirable to organize the different parts in different ways to reduce storage, retrieval and maintenance costs.

The usage distribution is, of course, a function. But we cannot specify what variables it is a function of, or even how many variables, without reference to a particular databank. Usually the variables can be specified at the level of the user interface, in terms of one or more of the keys. As a common example, a file containing chronological information, with time as a key, may have significantly different activities for different times: recent data may be requested frequently and older data less frequently. In this example, the usage distributuion would be a function, u(t), of the age, t, of the data. In other examples, usage might be a function of two or more parameters.

1.2.3.2 Data Interface

At the data level, requests can be analyzed precisely as queries, changes, deletions or additions. We define these operations for the basic access unit, the block.

DEF. A query is the operation of obtaining data in a block and delivering it, possibly after intermediate processing, to the inquirer.

DEF. A change is the operation of altering data in a block.

DEF. A deletion is the operation of obliterating a block of data.

DEF. An addition is the operation of including a new block of data.

These definitions are arranged roughly in the order of the effect of the operations on the organization and maintenance of a file; simpler organization permits the earlier operations, while the latter operations require more elaborate organization and maintenance.

It should be noted that physically the new block of data in an addition may be placed anywhere in the file, including replacement of a previously deleted block. It should also be noted that a deletion is not a special case of a change, even though it may be accomplished by simply changing a flag bit, because of the number of accesses required.

We can describe the above four operations in terms of accesses.

DEF. An access is the operation of locating and copying a block into a buffer area in core, or conversely of locating a suitable space and copying a block from core into secondary storage.

The access is the elementary costing unit for computer operations.

Table 1 shows the minimum number of accesses for each of the four basic operations. In most databanks, these minima cannot be achieved

Operation	Min. Accesses
Query	1
Change	2
Deletion	1
Addition	1

Table 1. Minimum number of accesses for basic requests.

simultaneously for all user-level types of request: the databank organization is a compromise among the request types aiming at minimal overall cost. An important first step in any analysis is to tabulate the actual number of accesses required by each of the four request types and compare with table 1.

Usage distribution.

At the data level, the usage distribution is given by u_i, the probability that the i^{th} block of the databank is accessed by an arbitrary request, $i = 1, \ldots, n$.

1.2.4 Databank Organization

An exhaustive description of databank organization is beyond the scope of this chapter. However, to establish a context, we will define more file organizations than we shall subsequently need for analysis of the FRI databank. In describing the databank which will be the subject of analysis, it is necessary to start with some definitions. Some of the terms will be used in a restricted sense.

DEF. A databank is a collection of files, cross-referenced among themselves and ideally containing no unnecessarily redundant information. The Financial Research Institute Daily Stock Exchange Databank will be referred to as the FRI databank; it contains 19 files.

DEF. A file is a collection of identical records on a secondary storage device. In the case of FRI the device is an IBM 3330 DASD, but since the FRI databank has recently (June '73) been transferred from an IBM 2314 DASD and is still stored in 2314 track images, any discussion in the analysis below which refers to hardware will suppose that the files are stored on a 2314. This definition excludes files with multiple record types.

A file may be organized in different ways, determined by access method. A discussion of access method immediately involves discussing groups of files which are logically related in special ways. The following definition is valuable.

DEF. An n-file is a group of n files which may be logically related. A databank is an n-file. A file is a 1-file.

We can discuss access methods in terms of n-files. We make a basic distinction between the sequential and the relative/direct categories: in the former, individual records have no identifying address; in the latter they do. This distinction applies whether the records are transferred from secondary to primary storage one at a time or whether they are transferred in groups, via a buffer:

records in a relative/direct buffer can be accessed by address (e.g. as array elements); records in a sequential buffer cannot.

<u>Sequential</u>: The records of a file are organized in sequence and a record may be accessed only after all preceding records have been accessed. Sequential access thus involves a 1-file.

<u>Sorted</u>: The records of a file are organized in a sequence which is determined by the <u>key</u> - i.e., the value of some of the data in the records. Sorted sequential access thus involves a 1-file.

<u>Indexed</u>: A sorted file is partitioned into subfiles, each identified by one key value selected from its records (usually the first or the last). Duplicates of these key values are stored in an <u>index</u>, which is either a sequential file or an indexed n-file. Indexed access thus involves an n-file: the simplest instance is a 2-file, and higher levels of index involve n-files with n>2.

<u>Relative/Direct</u>: Each record is stored at a unique position and accessed by the <u>address</u> of this position. In the relative case, the address is measured relative to the beginning of the file; in the direct case, the address is determined by the storage device. Within the general category, there are several methods of obtaining the address of a record.

<u>Sorted</u>: The records are organized in a sequence which is determined by a key. This requires only a 1-file.

<u>Hashed</u>: The addresses are computed from key values by one of many possible <u>hashing functions</u>. Hashing requires a 1-file.

<u>Chain/Multichain</u>: The address of each record is stored as a

field of another record. The records are organized by this chain of pointers into a sequence whose order is determined by a key. Several keys may be handled simultaneously by several chains. This method requires only a 1-file.

List/Multilist: The addresses of all the records are in a list, in an order determined by a key. Several keys may be handled by several lists. The list method requires a 2-file and the multilist method requires an n-file.

Inverted: Each different key value is stored at the beginning of a variable length record, followed by the addresses of all records with that key value. Several keys may be handled simultaneously. Inverted access thus requires a 2-file for a single key and an n-file of $n>2$ for several keys.

Implicit: This category contains the broad range of cases in which an organization or structure is implicit in the data, and the access method uses this structure: versions of the chain, list or inverted approaches may be used.

Note that in the file organizations involving n-files with $n>1$, such as indexed, list or inverted, the term index is often applied to the auxiliary files without discriminating which access method is used. We will follow this practice when it is clear which file organization is being discussed. The FRI files all fall in the relative category: most are accessed by index, using a combination of list and inversion techniques, and one uses chaining.

DEF. A record is a collection of fields. It is important to distinguish between an instance of a record and the form of the record, both often referred to as "record". Many instances of a record combine to make a file; but there is only one form of a record, which defines the positions of the fields in the record. The FRI record forms usually have a single field; in one case there are 10 fields.

DEF. A field is the basic unit of secondary storage. The amount of storage defined for a field depends on the type and range of data to be stored. All FRI fields are 1 word long, with the exception of the stock name field, which is 3 words (12 characters).

1.2.5. Sample Analysis: Sequential File

To exemplify the analytical elements described above, we give a complete cost analysis of a sequential file. We assume that the file

- grows linearly $n(t) = n_0 + gt$
- is sorted in ascending order on the value, k, of a single key
- is stored on an IBM 2314 direct access storage device (DASD)
- has fixed-length blocks

1.2.5.1 Acquisition Cost

Each time new data is added to the file a merge-type update operation is required, combining the old file with the new data to produce a new, extended, file. The new data must first be sorted (Fig. 1)

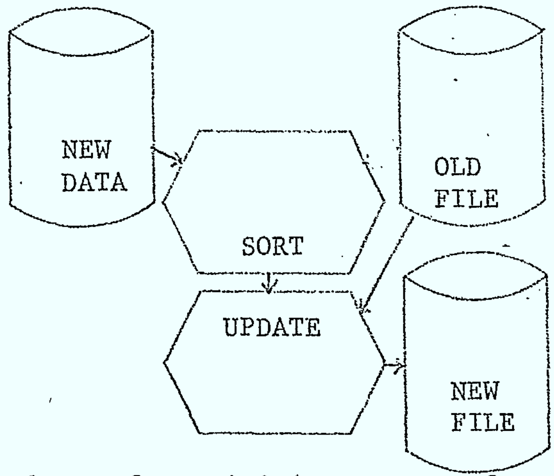


Fig. 1. Flowchart of acquisition process for sequential file.

The original acquisition involves the special case of the old file being empty.

1.2.5.1.1 Analysis of Sorting

Sorting techniques and their theory are major topics in their own right, but the case of external sorting on disk can be made fairly straightforward with a few simplifying assumptions. We follow Knuth (The Art of Computer Programming, §§ 5.4.1, 5.4.6, 5.4.9)

The time required to do a sort-merge on disk can be expressed

$$Nc\omega\tau\ (1 + \log_{\rho} S\ ) \qquad (1)$$

where N is the number of records in the file,

c is the number of characters or bytes per record,

τ is the time required to read a single character,

(On a 2314 $\tau = \dfrac{25\text{ ms./track}}{7294\text{ char/track}} = 3.43\ \mu\text{sec}$),

ω is the "overhead ratio", - the ratio to τ of the effective time to read a character, including arm movement

(on a 2314 with full cylinders and tracks,

$$\omega = \begin{cases} 1 \text{ if the file is on a single cylinder} \\ 1.07 \text{ if the file is on contiguous cylinders} \\ 1.14 \text{ if the file is on noncontiguous cylinders} \\ \quad\quad \text{or if multitasking causes arm contention.} \end{cases}$$

If only a proportion ρ of the available storage is used by the file under consideration ω must be increased by $(1-\rho)/\rho$.),

P is the number of simultaneous merges used, and

S is the number of "initial runs" - the number of subsets of the file that are sorted internally before merging begins.

In expression (1), $Nc\omega\tau$ is the time required to read a single pass of the file and $1 + \log_P S$ is the number of passes: a pass to distribute the S initial runs and $\log_P S$ passes to do the P-way merge. S is determined by the number of records, P', that can fit into core memory (M characters) less three buffers (B characters each):

$$P' = (M - 3B)/c \tag{2}$$

Typically, M = 100k = 102,400 bytes and B $\leq$ 7294 bytes.

Since the use of replacement selection makes it possible for a run of random data 2P records long to be processed in P' records worth of core memory, Knuth gives the number of runs, S:

$$S \approx \left\lceil \frac{N}{2P'} + \frac{7}{6} \right\rceil \tag{3}$$

Once S is determined, the order of merge, P can be found which gives the smallest number of passes, $\log_P S$, subject to P being small enough that P buffers will fit into core memory. The appropriate relationship among P, S and $m \equiv \log_P S$ is

$$P = \left\lceil S^{\frac{1}{m}} \right\rceil \qquad (4)$$

This relationship can be used as follows: given S, find the smallest m for which $P = S^{1/m}$ is not greater than the number of buffers that can fit into core. Thus in Knuth's example B = 5000 and S = 60:

m	1	2	3	4	5
P	60	8	4	3	2

which indicates that an 8-way merge in two passes is required.

An expression for the number of accesses required in a sort can be obtained from (1) by replacing the time $\omega\tau$ by 2/B. Since Nc/B = n, the number of blocks of data in the file, we have the cost for sorting

$$C_s = 2n\,(1 + \log_P S)\,C_\alpha \qquad (5)$$

The factor of two enters because each pass involves simultaneous reading and writing on different disk packs. It must be borne in mind that n is the size of the part of the file being sorted, and not necessarily the size, n(t), of the whole sequential file of §1.2.5

The arguments leading to equations (2), (3) and (4) relate, of course, to an optimum sorting method for a particular file. In analysis of real-life sorting, the parameters P and S must be chosen to correspond to the sort parameters actually used, or else (5) will be a poor approximation to the cost. On the other hand, (5) is an important tool in improving sorting methods to reduce the cost.

1.2.5.1.2 Update

The time and cost for an update operation on a file which finally has N records or n blocks are given respectively by (6) and (7).

$$N\ c\ \omega\ \tau \tag{6}$$

$$2\ n\ C_\alpha \tag{7}$$

To obtain the whole cost of acquisition of a linearly growing file, $n(t) : n_0 + gt$, we must discretize the time

$$T \equiv t/\Delta t$$

$$n_1 \equiv g\Delta t$$

where Δt is the time interval (assumed constant) between successive updates during the growth of the file, T is the age of the file measured with Δt as a unit, and n_1 is the number of blocks of data added to each update. Thus

$$n_T \equiv n \quad + n_1 T = n_o + gt = n(t). \tag{8}$$

The various costs of sorting and updating can now be written as shown in Table 2. In this table the number of passes, $m = \log_P S$, is taken to be the same for the initial update at T = 0 and for subsequent updates.

Time	Sort Cost	Update Cost
0	$2n_o (1 + m) C_\alpha$	$2n \quad C_\alpha$
1	$2n_1 (1 + m) C_\alpha$	$2(n_o + n_1) C_\alpha$
2	"	$2(n_o + 2n_1) C_\alpha$
3	"	$2(n_o + 3n_1) C_\alpha$
...	...	...
T	$2n_1(1 + m) C_\alpha$	$2(n \quad + Tn_1) C_\alpha$
Total	$2n_T(1 + m) C_\alpha$	$(T + 1) (n_o + n_T) C_\alpha$

Table 2. Costs for sorting and updating a sequential file with linear growth.

The totals can be expressed in an instructive way in terms of the continuous function, n(t):

$$\text{Total sort cost} = 2\ (1 + m)\ C_\alpha\ n(t)$$

$$\text{Total update cost} = \frac{2\ C_\alpha}{\Delta t} \int_o^t n(t')\ dt' \qquad (9)$$

which apply to any n(t), as long as updates are performed after equal intervals of time, Δt. We see how much sequential files can diverge from the ideal acquisition cost.

$$C_a \propto n(t).$$

It is quite likely that m or C_α will be different in the initial sort than they are in the subsequent, probably smaller, sorts for updating. Thus it is more general to write (10)

$$\text{Total sort cost} = 2\ (1 + m')\ C'_\alpha\ n_o + 2\ (1 + m)\ C_\alpha\ (n(t) - n_o).$$

1.2.5.2 Storage Cost

To find the storage cost we must relate the blocksize, B, to the unit of storage, Q. Let the number of blocks per unit of storage be the interger b so that

$$bB = \rho Q \qquad (11)$$

where ρ is the proportion of the storage unit actually occupied by data as suggested in §1.2.2 and, ideally, b = 1 so that the blocksize equals the unit of storage as nearly as possible.

The storage cost is

$$C_s = \frac{C_\sigma}{b} \Delta t \ (n_o T + n_1 \sum_{i=1}^{T-1} i)$$

$$= \frac{C_\sigma}{b} \ t \ (n_o \ + n_1 (T-1)) \qquad (12)$$

where C_σ is the cost per unit time of a unit storage space. This can be expressed in terms of n(t):

$$C_s \approx \frac{C_\sigma}{b} \int_o^t n(t') \, dt' \qquad (13)$$

1.2.5.3 Retrieval Cost

In terms of basic request operations, sequential organization is expensive, as a comparison of tables 3 and 1 indicates.

Operation	No. Accesses
Query	1 to n
Change	2 to n+1
Deletion	1 to n
Addition	-

Table 3. Number of access for basic requests: sequential file. n is the number of blocks in the file.

However, this is greatly mitigated if the requests come in batches, since the same figures apply no matter how many requests are made: this consideration shows that sequential files are often more economical than more sophisticated organizations.

The usage distribution may be expressed as a function, $u(k)$, of the single variable, k, the record key:

$$u\ (k)\ \ dk = \text{probability (key requested is in range } (k,\ k + dk)).$$

We relate this to u_i, the probability that the record whose key is requested is on the i^{th} block of the file:

$$u_i = \int_{(i-1)\nu}^{i\nu} u(k(\lambda))\ d\lambda \tag{14}$$

where the key is a function, $k(\lambda)$, of the record location, λ, and there are ν records per block:

$$\nu \equiv \frac{B}{c},\quad n = \frac{N}{\nu} \tag{15}$$

The cost of accessing a single record on block δ is just δC_α, and the average cost of accessing a single record is

$$\sum_{\delta=1}^{n} \delta u_\alpha\ C_\alpha = \sum_{\delta=1}^{n} \delta \int_{(\delta-1)\nu}^{\delta\nu} u(k(\lambda))d\lambda\ C_\alpha \tag{16}$$

For a batch of r requests, the cost of access is again δC_α, but this time δ is the <u>maximum</u> <u>depth</u> of access, i.e. $\delta = \max.\ (i_1,\ldots,i_r)$ where $i_1,\ldots,i_r$ are the blocks required by the r requests. To work out average costs, we need to convert from the distribution $u\ (k)$ of locations requested to the distribution $u_r\ (k)$ of maximum keys requested in a batch of r requests.

1.2.5.3.1 <u>Distribution of Depths</u>

Letting the r requested keys be $k_1,\ldots,k_r$, the distribution we want to find is

$$u_r\ (k)\ dk = \text{probability } (\max\ (k_1,\ldots,k_r) \text{ is in range } (k,k+dk)).$$

We shall define, in the normal way,

$$U_r(k) \equiv \int_k^\infty u(k')\,dk' = \text{probability } (\max(k_1,\ldots,k_r)>k). \qquad (17)$$

Evidently, $u_1(k) = u(k)$ and so

$$U(k) \equiv U_1(k) = \text{probability (single key} > k).$$

We can relate $U_r(k)$ to $U(k)$:

$$U_r(k) = \text{probability } (k_1 > k \ \underline{\text{or}}\ k_2 > k \ \underline{\text{or}} \ldots \underline{\text{or}}\ k_r > k)$$

$$= \text{probability } (\underline{\text{not}}\ (k_1 \leq k \ \underline{\text{and}}\ k_2 \leq k \ \underline{\text{and}} \ldots \underline{\text{and}}\ k_r \leq k))$$

$$= 1 - \prod_{i=1}^{r} \text{probability } (k_i \leq k)$$

$$= 1 - (1 - U(k))^r. \qquad (18)$$

Thus

$$u_r(k) = -\frac{d}{dk} U_r(k) = r\,u(k)\,(1-U(k))^{r-1} \qquad (19)$$

and we can obtain a discrete distribution of depths

$$d_\delta^{(r)} = \int_{(\delta-1)\nu}^{\delta\nu} u_r(k(\lambda))\,d\lambda$$

= probability δ is the maximum depth (in blocks) in a set of r requests. (20)

The average cost of retrieval for r requests is

$$\sum_{\delta=1}^{n} \delta d_\delta\, C_\alpha. \qquad (21)$$

<u>Example</u> When the requests are uniformly distributed over the N record locations, λ,

$$u(k(\lambda)) = \begin{cases} \frac{1}{N} & 0 \leq \lambda \leq N \\ 0 & \lambda > N \end{cases}$$

we have:

$$U(k(\lambda)) = \begin{cases} 1 - \frac{1}{N} & 0 \leq \lambda \leq N \\ 0 & \lambda > N \end{cases}$$

$$u_r(k(\lambda)) = \frac{r\lambda^{r-1}}{N^r}$$

and $d_\delta \simeq \frac{1}{n^r}(\delta^r - (\delta-1)^r)$, using (15)

the average depth is

$$\overline{\delta} = \sum_{\delta=1}^{n} \delta d_\delta \simeq \frac{1}{n^r} \sum_{\delta=1}^{n} \delta^{r+1} - \delta(\delta-1)^r$$

$$= \frac{1}{n^r} \quad n^{r+1} - \sum_{1}^{n-1} \delta^r.$$

Now, it is possible to show that

$$\sum_{1}^{n-1} \delta^r = \frac{n^{r+1}}{r+1} + \text{smaller powers of } n$$

so that when n or r are large

$$\overline{\delta} \simeq \frac{1}{n^r} \quad n^{r+1} - \frac{n^{r+1}}{r+1}$$

$$= \frac{r\,n}{r+1} \qquad (22)$$

Thus the average cost for a batch of r uniformly distributed requests on a sequential file of n blocks is approximately

$$\frac{r\,n}{r+1} \quad C_\alpha$$

1.2.5.3.2 Distribution of Batch Size

The probability, s_r, of a batch containing r requests must be obtained empirically. Once it is known, we can use (19), (20), and (21) to find the average cost over all batch sizes.

For queries or deletions the average cost is

$$C_q \equiv \sum_{r=0}^{\infty} s_r \sum_{\delta=1}^{n} \delta \int_{(\delta-1)\nu}^{\delta\nu} r u(k(\lambda)) \, [1-U(k(\lambda))]^{r-1} d\lambda \; C_\alpha \tag{23}$$

For changes, each of the r requests requires an additional access for the rewrite:

$$C_c \equiv \sum_{r=0}^{\infty} r s_r \sum_{\delta=1}^{n} \delta \int_{(\delta-1)\nu}^{\delta\nu} u(k(\lambda)) \, [1-U(k(\lambda))]^{r-1} d\lambda + 1 \quad C_\alpha \tag{24}$$

Note that the upper limit of the first sum in each equation is not N, since § 1.2.5.3.1 does not include repeated requests. The lower limit can be 0 because of the factor r in the sum.

To find the retrieval cost, C_r, we now need only the request rate as a function of time. We define three functions, which must be obtained empirically:

$$\left.\begin{array}{l} r_q(t) \\ r_c(t) \\ r_d(t) \end{array}\right\} \text{the number of batches, of any size, per unit time, retrieved at time t as} \left\{\begin{array}{l} \text{queries} \\ \text{changes} \\ \text{deletions} \end{array}\right.$$

Using these, we have the retrieval cost

$$C_r = \int_o^t (r_q(t') \; C_q + r_c(t') \; C_c + r_d(t') \; C_q \; dt' \qquad (25)$$

1.2.5.4 Maintenance Cost

A sequential file, updated as described in §1.2.5.1, does not require maintenance because it is always in order.

1.2.5.5 Costs for Sequential File

In summary, the cost for a sequential file is the sum

$$C = C_a + C_s + C_r$$

where the acquisition cost, C_a, is the sum of equations (9) and (10). the storage cost, C_s, is given by equation (13) and the retrieval cost, C_r, is given by equation (25).

The basic unknowns that must be determined empirically are:

$n(t)$, the "growth rate",

$r_q(t)$, $r_c(t)$, $r_d(t)$, the request rates for queries, changes and deletions respectively.

$u(k)$, the usage distribution, as a function of the single sorted keys, k,

$k(\lambda)$, the distribution of keys over the storage locations, λ, and

s_r, the distribution of batch sizes.

Example

$n(t) = n_o + gt$ (linear file growth)

$r_q(t) = r_o + gt$ (linear query rate growth)

$r_c(t) = r_d(t) = 0$ (no changes or deletions)

$u(k(\lambda)) = \frac{1}{N}$ (uniform usage)

$$s_r = \frac{\sigma^r}{r} e^{-\sigma} \quad \text{(Poisson batch size : mean size } \sigma)$$

The above assumptions give : the acquisition cost,

$$C_a = 2(1 + m') n_o C'_\alpha + [2((1+m) g + \frac{n_o}{\Delta t}) t + \frac{g}{\Delta t} t^2] C_\alpha; \quad (26)$$

the storage cost,

$$C_s = \frac{1}{b} (n_o t + \frac{1}{2} gt^2) C_\sigma; \quad (27)$$

and the retrieval cost,

$$C_r = \int_0^t dt' \sum_{r=0}^{\infty} \frac{\sigma^r}{r!} e^{-\sigma} \frac{rn(t')}{r+1} r_q(t') C_\alpha$$

$$= \frac{1}{\sigma} (\sigma - 1 - e^{-\sigma}) (n_o r_o t + \frac{1}{2}(n_o q + r_o g) t^2 + \frac{1}{3} gqt^3) C_\alpha \quad (28)$$

The last result can be shown using

$$\sum_{r=0}^{\infty} \frac{\sigma^r}{r!} \frac{r}{r+1} = \sum_{r=0}^{\infty} \frac{\sigma^r}{r!} (1 - \frac{1}{r+1}) = e^\sigma - \frac{1}{\sigma} (e^\sigma - 1) \quad (29)$$

Note that n in equation (22) is time-dependent, although this time-dependence is suppressed in the discussion of §1.2.5.3.1.

To take the example further, we specify values for m (the number of merge passes in the sort), n_o (the original file size), g (the rate of growth of the file), Δt (the interval between updates), b (the number of blocks per unit storage), σ (the mean batch size of queries), r_o (the original request rate), q (the rate of increase of the request rate) and the fundamental costs C_α and C_σ .

We imagine a file that was created four years ago on an IBM 2314 with n_o = 1000 tracks, and growing at g = 10 tracks a day by means of updates every Δt = 10 days. Analysis following §1.2.5.1.1 indicates that the initial sort could be done either with m = 3 in 100k or with m = 2 in 200k, and that subsequent sorts can be done with m = 2 in 100k. Since we shall consider a C_α that varies with the core memory used, we shall analyse both initial sorts.

We suppose b = 1, ie. each block occupies one 2314 track and that there are 91 records per track (card image file).

If we suppose the request rate is such that 2% of the records in the file are queried daily, in batches of mean size $\sigma = 10$ queries, we have $r_o = 182$ batches of queries per day and q = 1.82 batches of queries/day^2.

Finally, we suppose that the file is stored online with backup, so that there is no mounting charge and $C_\alpha = C_{10} = 0.133$¢ and $C_\sigma = 4$¢/ track/day. (These charges correspond to current McGill charges assuming programs that run in 100k at priority 2 : for the original sort in 200k, $C_\alpha = 0.167$¢ since I/O charges depend on the amount of core being used.)

Acquisition Cost

We first must decide which initial sort to use. From equation (10) the cost of the initial sort is $2(1+m')C'_\alpha n_o$. Comparison of the two alternatives gives

$$\frac{(m = 3,\ 100k)}{(m = 2,\ 200k)} = \frac{4 * 0.133}{3 * 0.167} = \frac{16}{15}$$

so that the 200k sort with m = 2 is 7% cheaper than the other, and we accordingly adopt it.

With these numbers, the total acquisition cost over the age, t, of the file is

$$C_a = \$10 + \$0.2133t + \$0.00133t^2$$

That is, over the four-year, or 1040-day lifetime of the file (5-day weeks), the total cost for acquiring the data has been \$1677.

The major part of this amount, \$1445, comes from the t^2 part of the expression and is due to the repeated passes through the file required by the update process. In this case there have been 104 updates of 100 tracks each, requiring over a million block accesses as the file grew from 1000 to 11400 blocks, so the updates alone account for \$1460 of the cost. Since we are dealing with a sequential file, we should make the time, Δt, between updates as large as is consistent with the needs of the users for up-to-the minute information.

Storage Cost

The total storage cost over the age, t, of the file is

$$C_s = \$40t + \$0.20t^2$$

or, in 1040 days, \$257,920. Of course, we have chosen the most expensive storage, online disk with backup. No backup would reduce the cost (at McGill) by a factor of 2 and offline storage would reduce it by a further factor of 20: costs here are determined by the value we attach to the data and by the amount of query response time we will tolerate. The cheapest way to store the file would be on tape: the present 11400 blocks of the file could be stored on two 2400 tapes recorded at 1600 b.p.i. for a total cost of about \$25 (double this for backup).

Retrieval Cost

The total query cost over the age, t, of the file is

$$C_r = \$218.40t + \$2.184t^2 + \$0.00728t^3$$

In 1040 days this is \$10.78 million, which shows the inadequacy of sequential files for query processing. In the course of four years 1.17 million batches of queries have been run against the file, each

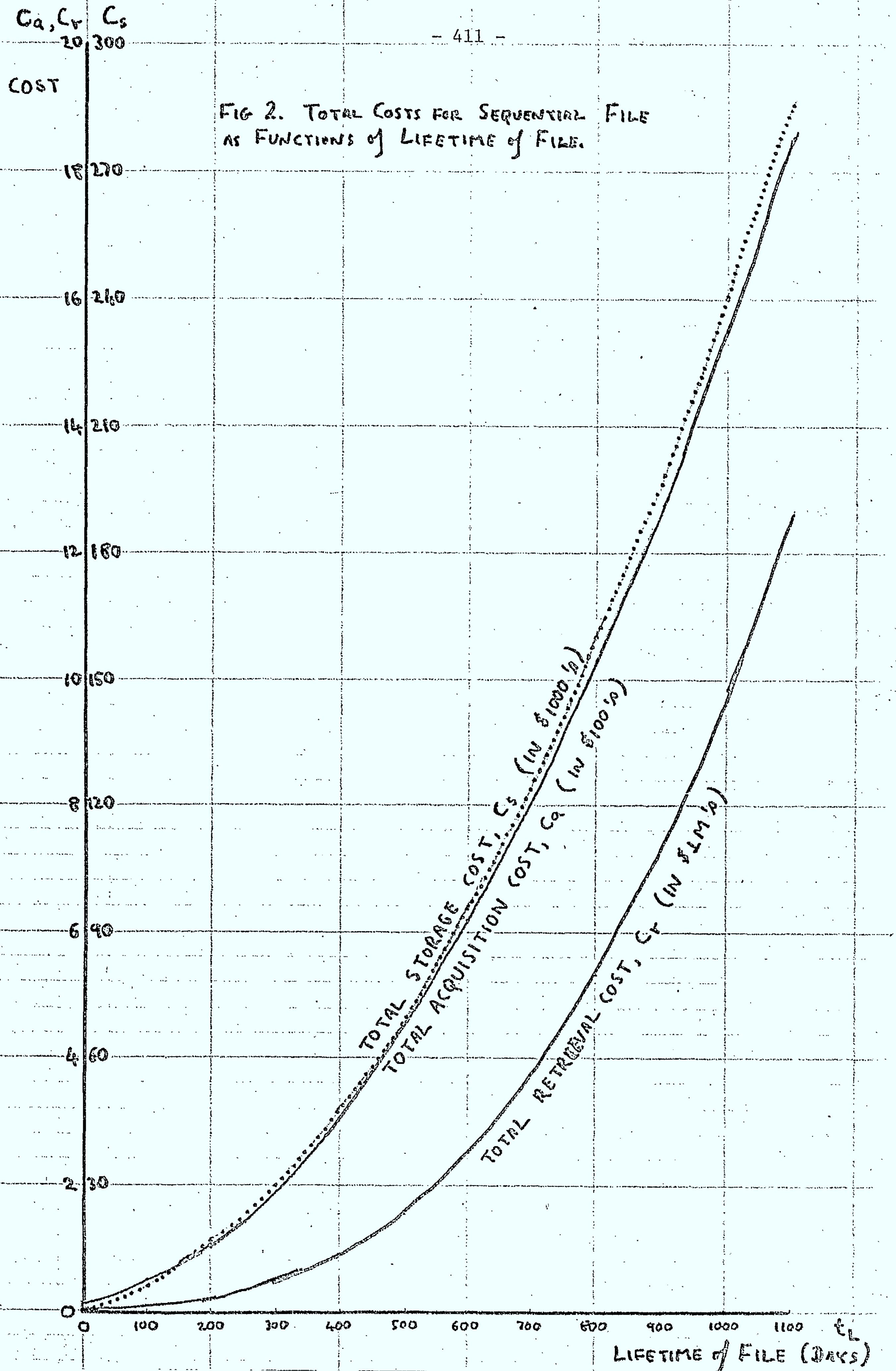


FIG 2. TOTAL COSTS FOR SEQUENTIAL FILE AS FUNCTIONS OF LIFETIME OF FILE.

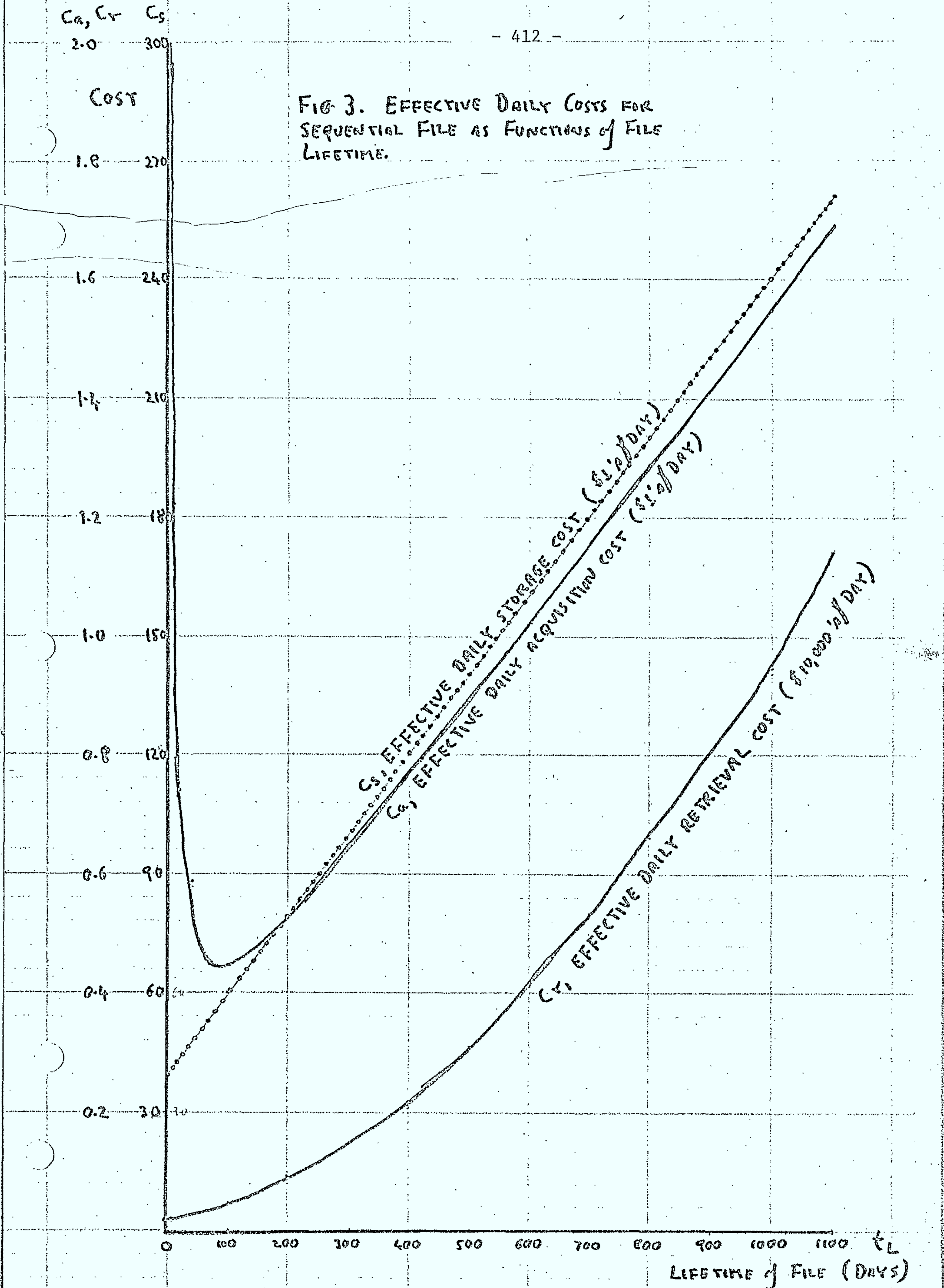


FIG 3. EFFECTIVE DAILY COSTS FOR SEQUENTIAL FILE AS FUNCTIONS of FILE LIFETIME.

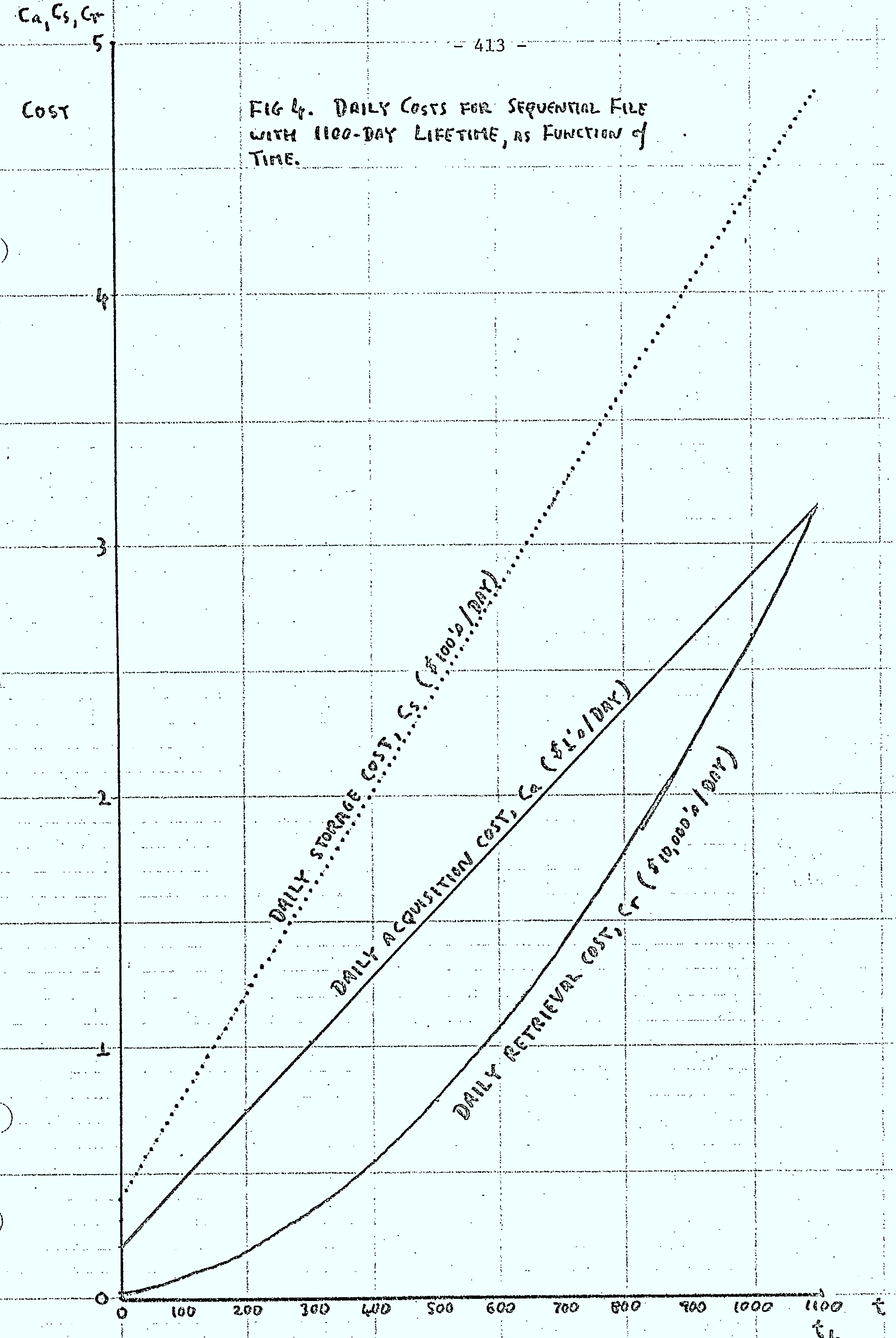


FIG 4. DAILY COSTS FOR SEQUENTIAL FILE WITH 1100-DAY LIFETIME, AS FUNCTION OF TIME.

batch requiring a pass through the file at costs ranging from \$1.33 (1000 tracks) to \$15.20 (11400 tracks) per pass. Most of this figure is due to the fact that the retrieval cost is proportioned to the file size - a unique property of sequential files. Even so, the cost could be reduced by increasing the mean batch size, σ : the retrieval rate and hence the retrieval cost vary in inverse proportion. The more a given number of requests on a sequential file are batched up, the cheaper and faster the access.

Another important factor is the usage distribution. We have assumed an expensive case, a uniform distribution. A distribution which tails off at a certain depth into the file would be much cheaper. On the other hand, a distribution with a mode deep in the file could be worse although the file can be organized to avoid this case.

1.2.5.6 Interpreting the Costs

It is important to bear in mind that the costs we have discussed so far are total costs added up over the whole lifetime of the file. Figure 2 shows a plot of these totals as a function of the lifetime. This way of expressing the costs is useful if a capital amount has been made available for a file or databank: we can determine how long we can afford to keep it.

However, it is often preferable to know the costs as rates per day (or month or year): Figures 3 and 4 show two different versions. In Figure 3, the total cost has simply been amortized over the lifetime of the file. If each cost shown in Figure 2 is written

$$C = C_o + C_1 (t_L)$$

where t_L is the lifetime, then the costs shown in Figure 3 are

$$(C_o + C_1 (t_L))/t_L$$

This is useful if we have a final daily amount to spend on the file or databank. In C_a in Figure 3 we see the effect of the C_o/t_L term when t_L is small: it is not worth the expense of setting up the file if we only keep it a few days.

The cost of Figure 3 does not indicate the actual daily cost during the lifetime of the databank, but only an average of effective daily cost. In Figure 4 we show the actual daily cost, which is

$$C_o/t_L + \frac{d}{dt} C_1(t)$$

for a file with lifetime $t_L = 1100$ days. Now the cost is a function of time, t, during the life of the databank, and of t_L, the lifetime. If we change t_L, only C_a will change: it will shift up and down parallel to itself as t_L decreases or increases respectively. Only when t_L is quite small will there be significant changes.

Figure 4 is useful if the databank is paid for by a group of subscribers or members paying a fixed subscription: it shows how the number, m(t), of members must increase as the databank grows if their subscriptions are not to increase. In the case of a sequential file we see clearly that as the file grows even a growing number of members will not get uniform service over a period of time at unvarying costs.

1.3 The FRI Daily Stock Exchange Data Bank

The Financial Research Institute daily stock exchange databank contains price, volume and related information on some 6,700 equities (December 1973) and 1,700 bonds for a period of time commencing in the middle of 1969 (some 1,100 days as of December 1973). As shown in Figure 1, the number of equities grows essentially linearly with time at a rate of about 510 per year. Evidently, the daily information also grows linearly, so the overall size of the databank is a quadratic function of its age, t: $n(t) = n_o + gt + 1/2g't^2$. (The growth coefficients g and g' are derived in §1.3.1)

The databank is characterized at the user level by two keys, stock and day. At this level, the usage distribution is thus essentially two-dimensional, being a function, $U(s,d)$, of stock, s, and day, d. The key, s, for stock is a four-byte ticker symbol, and the relation, $s(\lambda)$, between stock and location, λ, is a complicated one, depending on the historical order of appearance of the stocks. The key, d, for day is a five-digit number holding the Julian date in the form YYDDD. (Weekends and statutory holidays are excluded.) The relation $d(\lambda)$, between d and location, λ, is straightforward, with data for subsequent days being stored in subsequent positions within each partition of the databank.

The two keys, s and d, can be used to analyze the databank into three different classes of data: data dependent only on s (stock-dependent, data dependent only on d (day-dependent), and data which requires specification of both s and d to determine location (stock-day-dependent). The bulk of the data is stock-day-dependent; the first two classes contain summary information and data used to

FRI: GROWTH of STOCKS DATA BANK

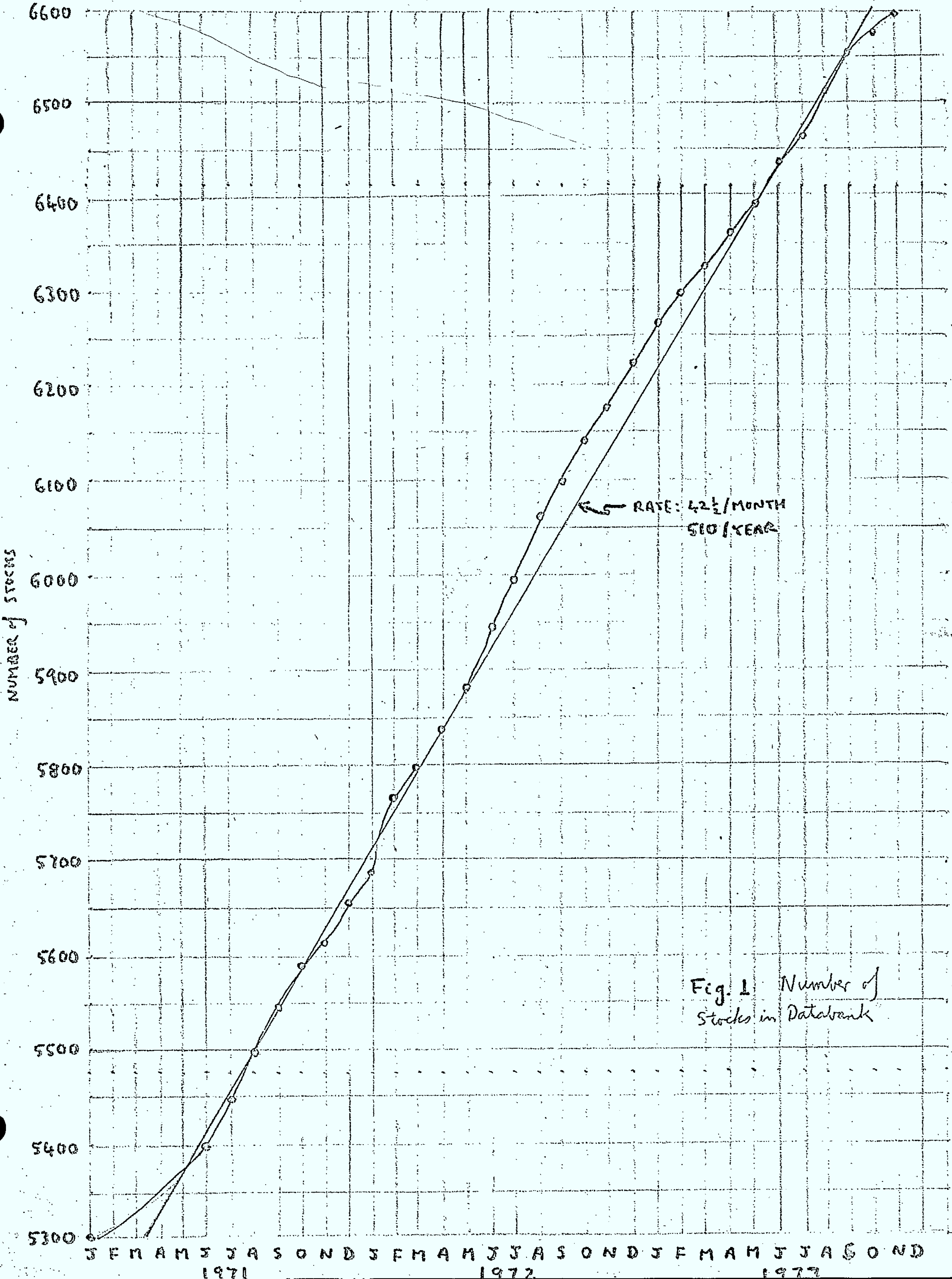


Fig. 1 Number of Stocks in Databank

locate the stock-day data. Table 1 gives details of the data in these three classes.

A fourth data class is an overflow area for historical data on dividends and stock splits. This data is primarily stock-dependent, and the most recent record for each stock is part of the stock-dependent class. This historical information is not sufficiently extensive to be included in the class of stock-day data, and is held in a chained organization in an overflow area. Table 3 indicates the extent of this overflow data.

Table 2 gives the query-change-delete-add analysis of the fields described in Table 1. Note that there are no deletions of data. As well as the number of accesses required for a simple request, Table 2 shows the approximate frequencies of requests, and, in the case of queries, the codes associated with the queries by FRI.

The databank currently (December 1973) resides on two IBM 3330 direct access storage devices (DASD) packs, but it is stored in blocks compatible with the IBM 2314 DASD, for historical reasons.

CLASS	DESCRIPTION	TYPE	FILE			FIELD		
			NO.	DESCRIPTION	SIZE (31.12.73)	NO.	DESCRIPTION	LENGTH
S	Stock Index	2-file	1	Ticker Symbols	5 trk	1	Ticker Symbol	1 wd.
			2	Locations	5 trk	1	Location	1 wd.
	Stock Information	8-file	1	Names	15 trk	1	Name	3 wd.
			2	Exchanges	5 trk	1	Exchange (1,2,3,4)	1 wd.
			3	Types	5 trk	1	Type (1,2,3,99)	1 wd.
			4	Ticker Symbols	5 trk	1	Ticker Symbol	1 wd.
			5	No. Shares	5 trk	1	No. Shares	1 wd.
			6	Volumes	5 trk	1	Volume	1 wd.
			7	$ Volumes	5 trk	1	$ Volume	1 wd.
			8	STOX Pointers	5 trk	1	STOX Pointer	1 wd.
	Dividends	1-file	1	Dividends	15 trk	1	Amount	1 wd.
	Stock Splits	1-file	1	Stock Splits	15 trk	2	Date	1 wd.
	Dividend Rates	1-file	1	Dividend Rates	15 trk	3	Overflow Pointer	1 wd.
D	Day Index	1-file	1	Day Index	6 trk	1	Date	1 wd.
						2	Block Pointer	1 wd.
						3	Track Pointer	1 wd.
						4-10	Stock Index	1 wd.
SD	STOCKS	2-file	1	Prices	5500 trk	1	Price	1 wd.
			2	Volumes	5500 trk	1	Volume	1 wd.
	STOX	2-file	1	Prices	800 trk	1	Price	1 wd.
			2	Volumes	800 trk	1	Volume	1 wd.
O'FLOW	Dividends	1-file	1	Dividends	72 trk	1	Amount	1 wd.
	Stock Splits			Stock Splits		2	Date	1 wd.
	Dividend Rates			Dividend Rates		3	Chain Pointer	1 wd.

Table 1. The Structure of the FRI Stocks Databank

CLASS	FILE/FIELD DESCRIPTION	QUERY CODE[2]	QUERY ACCESSES	QUERY FREQ[1]	CHANGE ACCESSES	CHANGE FREQ	ADDITION ACCESSES	ADDITION FREQ
	Ticker Symbols	–	–	–		0	(merge)	12–30[5]
	Locations	1	2	50		0	1	"
	Names	7	1[2]	40		0	1	"
	Exchanges	5	1[2]	40		0	1	"
	Types	6	1[2]	40		0	1	"
	Ticker Symbols	2	1[2]	40		0	1	"
	No. Shares	21	1[2]	40		0	1	"
	STOX Pointers	–	1[2]	40		0	1	"
	Volumes	22	1[2]	40	2+	10	1	"
	$ Volumes	23	1[2]	40	2+	10	1	"
	Dividends		1[2] [3]		2	0–4[7]	1[3]	"
	Amount	10,12		40				
	Date	11,13		40				
	O'flow Pointer			35				
	Stock Splits		1[2] [3]		2	0–4[7]	1[3]	"
	Factor	14		40				
	Date	15		40				
	O'flow Pointer			35				
	Dividend Rate		1[2] [3]		2	0–4[7]	1[3]	"
	Amount	19		40				
	Date	20		40				
	O'flow Pointer			35				
D	Day Index		1[2] [3]			0	1[3]	30[6]
	Date	–		–				
	Block Pointer	–		–				
	Track Pointer	–		–				
	Stock Indices	18		40				
SD	STOCKS Prices	3	1[2]	40	2[4]	12–30[5]	1[4]	30[6]
	STOCKS Volumes	4	1[2]	40	2[4]	"	1[4]	30[6]
	STOX Prices	3	1[2]	40	2[4]	30[6]	1[4]	12–30[5]
	STOX Volumes	4	1[2]	40	2[4]	30[6]	1[4]	"
O'FLOW	Dividends		(chain)			0	1	0–4[7]
	Amount	10,12		35				
	Date	11,13		35				
	Stock Splits		(chain)			0	1	0–4[7]
	Factor	14		35				
	Date	15		35				
	Dividend Rates		(chain)			0	1	0–4[7]
	Amount	19		35				
	Date	20		35				

TABLE 2. Operations on the FRI Stocks Databank

Notes   Data is never deleted.

0. The code used in subroutines STOCKS and STOX for queries.

1. Frequency code: 0 not applicable, 1 annual, 2 semi-annual, 4 quarterly, 10 montly, 20 weekly, 30 daily, 50 hourly.

2. When access requires a key, it is assumed that the key has already been found.

3. Accesses to several fields in a record can be done simultaneously.

4. Accesses given per track.

5. New stocks are added at a rate of 510 per year at intervals varying from three weeks to one day. In the case of file STOCKS, these additions amount to changes since they simply involve the rewriting of previously created blank records. Periodically, space is depleted and a maintenance operation is required to reorganize the data.

6. Historical data is added daily. In the case of file STOX, these additions are considered changes in the same sense as in Note 5.

7. New data on dividends are added up to four times a year according to Table 3. Each addition involves copying the record from the stock-dependent area to the overflow area and then rewriting it with the data from the addition.

Additions per year	Dividends	Stock Splits	Dividend Rates
0	42%	98%	70%
1	8%	2%	23%
2	2%	-	7%
3	-	-	-
4	48%	-	-

Table 3. Acquisition of Dividend Data: Percentage of all Securities.

1.3.1 Growth of the FRI Databank

The four classes of the data in the databank grow in time according to different rules. The stock- and day-dependent data grow linearly with time wile the stock-day and overflow data grow quadratically.

In analyzing the growth of the data, we shall need to know ν, the number of records per block, for various files. This is given in terms of the number of characters per unit of storage, b:

$$\nu = \left\lfloor \frac{Q}{bc} \right\rfloor$$

The FRI files come in only three different record sizes, 1, 3 and 10 words, where a word is four bytes. Table 4 shows the possible values for ν for the IBM 2314 and 3330 DASDs when b = 1.

Record Size	ν 2314	ν 3330
1 wd.	1823	3257
3 wd.	607	1085
10 wd.	182	325

Table 4. Records per track for FRI file.

1.3.1.1 Stock- and Day-dependent Data (Classes S and D)

In Class S there are nine files with 1-word records, one with 3-word records and three dividend files, with 3-word records, which are treated separately. All but the dividend files contain N_s records, where N_s is the number of stocks at any given time, t. The dividend files contain the following numbers of records, where the coefficients are obtained from Table 3:

FILE	Dividends	Stock Splits	Dividend Rates
NO. RECORDS	$.58N_s$	$.02N_s$	$.30N_s$

N_s varies linearly with time:

$$N_s^{(t)} = N_{s0} + \gamma_s t$$
$$= 4430 + \frac{510}{260} t \tag{1}$$

where the numbers in the second line give the original number of stocks in July 1969 and the daily increase since then. In addition, 1700 bonds were added in June 1972, so that after this date, t_1,

$$N_s(t) = N'_{s0} + \gamma_s (t - t_1) = 7630 + \frac{510}{260} (t - t_1) \tag{2}$$

In Class D there is a single file, with 10-word records, containing N_D records:

$$N_D^{(t)} = N_{D0} + \gamma_D t$$
$$= t \tag{3}$$

Equation (3) states that at $t = 0$ there are no records and that subsequently one record is added per day. The addition of bonds has no effect.

The number of blocks, $n(t)$, allocated for each file at time t is

$$n(t) = \lceil N(t)/\nu \rceil$$
$$= \begin{cases} \lceil n_o + gt \rceil & t < t_1 \\ \lceil n'_o + g(t - t_1) \rceil & t \geq t_1 \end{cases} \tag{4}$$

where

$$n_o = \begin{cases} N_{s0}/\nu & \text{blocks (class S)} \\ 0 & \text{blocks (class D)} \end{cases} \tag{5}$$

$$n_o' = (N_{s0} + \gamma_s t_1 + 1700)/\nu \text{ (class S only)} \tag{6}$$

$$g = \begin{cases} \gamma_s/\nu & \text{(per day (class S)} \\ 1/\nu & \text{per day (class D)} \end{cases} \tag{7}$$

and

$$t_1 = \begin{cases} 720 \text{ days} & \text{(class S)} \\ \infty & \text{(class D)} \end{cases} \tag{8}$$

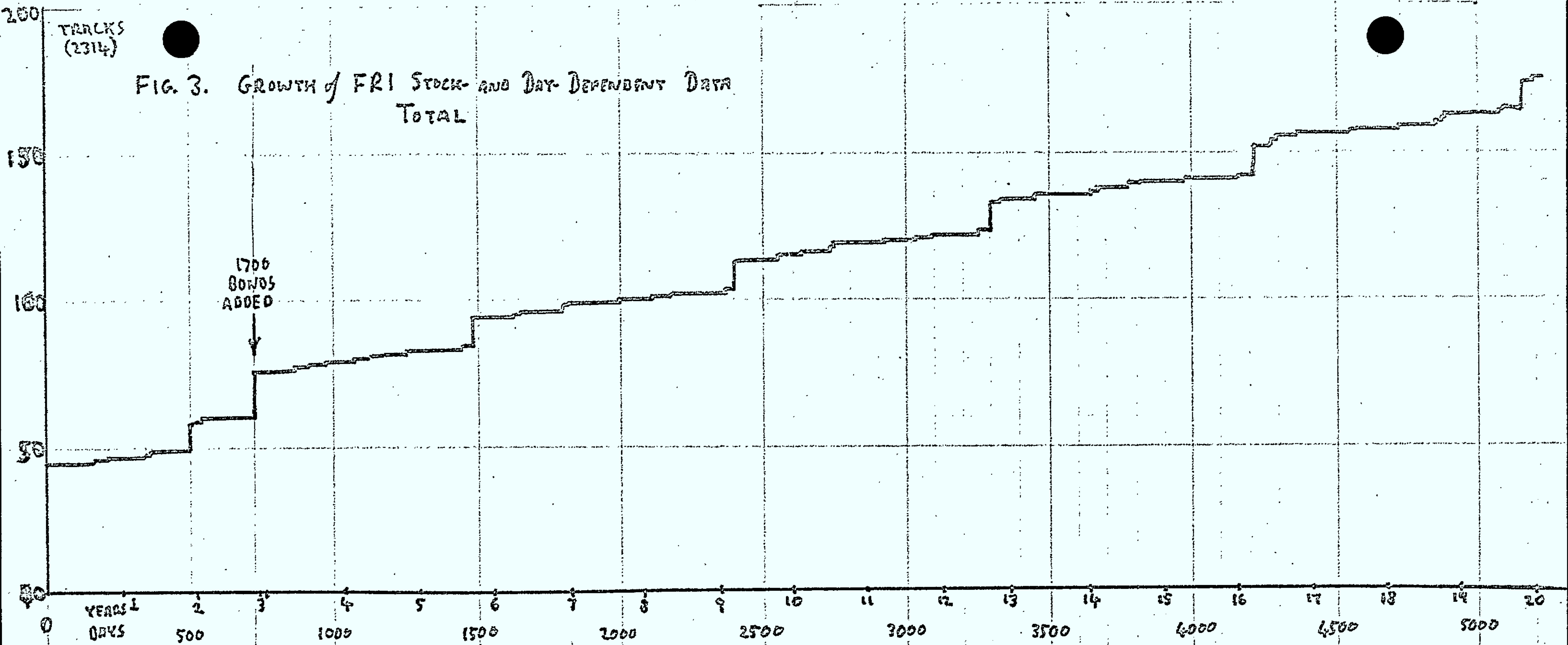
This last relation, for t_1, enables us to treat class S and class D data together.

Figure 2 illustrates equation (4) for all the files of classes S and D, assumed stored on an IBM 2314 with b = 1. The quantities used in the graph, n_o, n_o, Δt, $\Delta_o t$, $\Delta_1 t$, $\Delta_1' t$, are presented in Table 5 for all the type of files, where

$$\begin{aligned} \Delta t &\equiv 1/g \\ \Delta_o t &\equiv (\lceil n_o' \rceil - n)\Delta t \\ \Delta_1 t &\equiv (\lceil n_o' \rceil - n_o')^o \Delta t \\ \Delta_1' t &\equiv (t_1 - \Delta_o t) - \lfloor (t_1 - \Delta_o t)/\Delta t \rfloor \Delta t \end{aligned} \tag{9}$$

The times Δt, $\Delta_o t$, $\Delta_1 t$ and $\Delta_1' t$ are shown in Figure 4:

Δt is the time between allocations of new blocks to the file, $\Delta_o t$ and $\Delta_1 t$ are the times required to bring N_0 and N_0' up to an exact number of blocks and



TRACKS (2314)
FIG. 3. GROWTH of FRI STOCK- AND DAY-DEPENDENT DATA
TOTAL
1700 BONDS ADDED
YEARS
DAYS

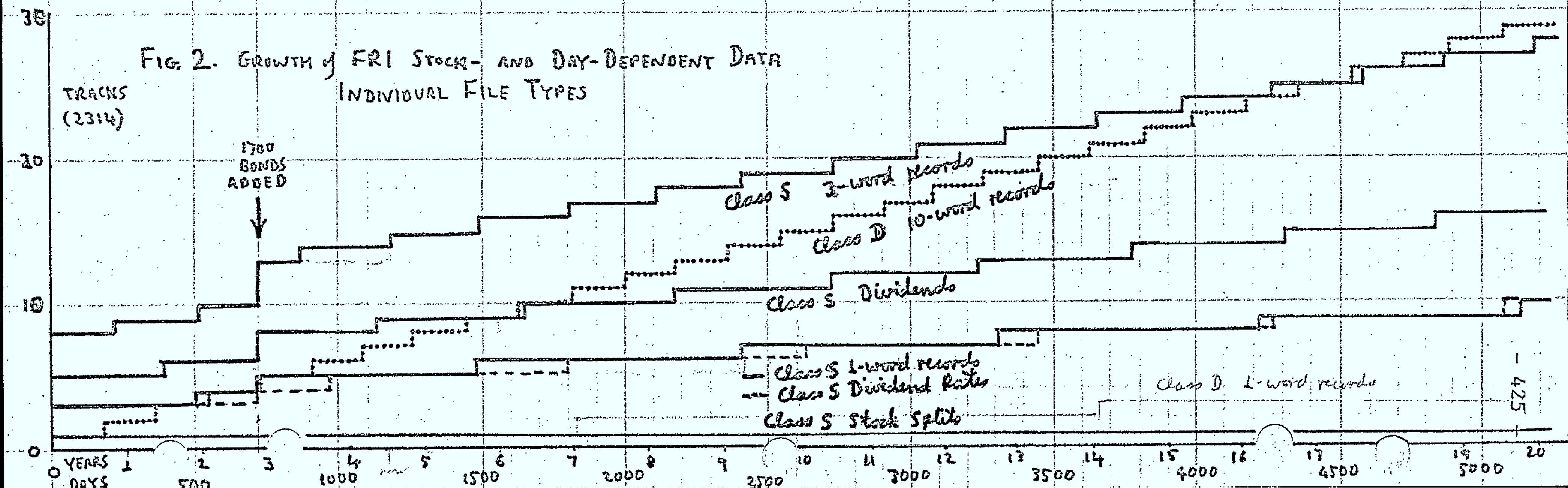


FIG. 2. GROWTH of FRI STOCK- AND DAY-DEPENDENT DATA
INDIVIDUAL FILE TYPES
TRACKS (2314)
1700 BONDS ADDED
Class S 2-word records
Class D 10-word records
Class S Dividends
Class S 1-word records
Class S Dividend Rates
Class D 1-word records
Class S Stock Splits
YEARS
DAYS

Class	File type	No. Files[1]	Device	n_o (tracks)	n_o' (tracks)[2]	Δt(days)	$\Delta_o t$(days)	$\Delta_1 t$(days)[2,3]	$\Delta_1' t$(days)[2,3]
S	1-wd record	9	2314	2.430	4.137	929.4	529.7	801.8	190.3
			3330	1.360	2.316	1660.4	1062.4	-	-
	3-wd record	1	2314	7.298	12.426	309.5	217.2	177.8	193.4
			3330	4.083	6.951	553.1	507.3	26.9	212.7
	Dividends	1	2314	4.233	7.207	533.5	409.3	423.2	310.8
			3330	2.368	4.032	953.7	602.6	923.3	117.4
	Stock Splits	1	2314	0.146	0.249	15472.6	13214.1	-	-
			3330	0.082	0.139	27656.9	25398.4	-	-
	Dividend Rates	1	2314	2.189	3.728	1031.5	836.1	-	-
			3330	1.225	2.085	1843.8	1429.2	-	-
D	1-wd record	0	2314	0	-	1823	0	-	-
			3330	0	-	3257	0	-	-
	10-wd records	1	2314	0	-	182.3	0	-	-
			3330	0	-	325.7	0	-	-

Table 5. Initial allocations and allocation time increments for stock and day data.

Notes

1. The number of files in classes S and D of the type described: data for the 1-word class D file with 0 in this column is used in the section on class SD.
2. For files of class D, columns n_0', $\Delta_1 t$, $\Delta_1' t$ are not used.
3. For files of class S with $\Delta_o t > t_1 = 720$ days, $\Delta_1 t$, $\Delta_1' t$ are not used.

$\Delta_1't$ is the time between t_1 and the time of the preceeding increase in the number of blocks.

Evidently $\Delta t \geq \Delta_o t$, $\Delta_1 t$, $\Delta_1' t$

Finally, Figure 3 shows the total size of the class S and class D components of the FRI databank as a function of time.

Equation (4) and Figures 2 and 4 give the size, $n(t)$, of each file in classes S and D. It will be necessary for later analysis to know the area, $\int_o^t n(t')dt'$. Rather than compute this by the discrete sum suggested by Figure 4, we can use an approximation,

$$\int_o^t n(t')dt' \approx \begin{cases} \int_o^t (n_0 + 1/2 + gt')dt' & t \leq t_1 \\ \int_o^t (n_0' + 1/2 + g(t' - t_1)dt' - h_1 t_1 & t > t_1 \end{cases} \tag{10}$$

where the addition of 1/2 to equation 4 improves the approximation and $h_1 \equiv n_0' - n_0 - gt_1$.

It is possible to rewrite equation (10) exactly:

$$t_n(t')dt^1 = \begin{cases} \lceil n_0 \rceil \; t & 0 \leq t \leq \Delta_o t \\ (n_0 + 1/2)t + gt^2/2 + c_0 + c_t(t,\Delta_0 t) & \Delta_o t < t \leq t_1 \\ (n_0 - \lceil n_0' \rceil + 1/2 + 1/2\; gt_1)t_1 & t_1 < t \leq t_1 + \Delta_1 t \\ \quad + \lceil n_0' \rceil \; t + c_0 + c_t\; (t_1, \Delta_o t) & \\ (n_0 - n_0' + gt_1)t_1 + (n_0' + 1/2 - gt_1)t & t > t_1 + \Delta_1 t \\ \quad + gt^2/2 + c_o + c_1 + c_t(t, t_1 + \Delta_1' t) & \end{cases} \tag{11}$$

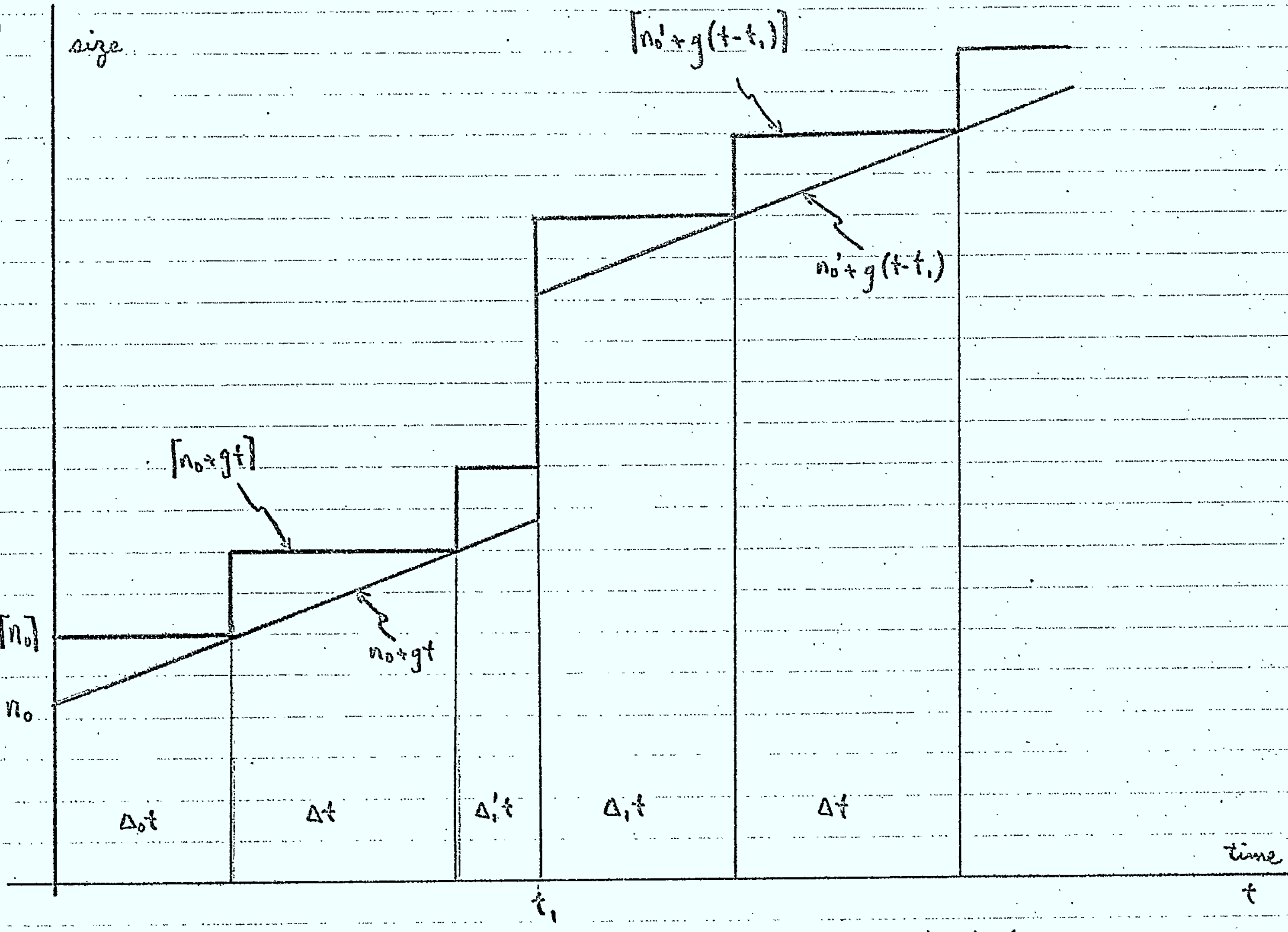


Fig. 4. Characteristics of a linearly-growing file with an increment at time t_1.

where small correction terms, c_0, c_1, c_t have been introduced:

$$
\begin{aligned}
c_0 &= (\Delta_o t/\Delta t - 1)\Delta_o t/2 \\
c_1 &= (\Delta_1' t - \Delta_1 t)(1 - (\Delta_1 t + \Delta_1' t)/\Delta t)\ /2
\end{aligned}
\qquad (12)
$$

and

$$c_t(t,x) = (1 - \tilde{t}(t - x)/\Delta t)(\tilde{t}(t - x))/2$$

where

$$\tilde{t}(y) = y - \lfloor y/\Delta t \rfloor \Delta t$$

Of these correction terms, c_0 is the correction between 0 and $\Delta_o t$ and is in the range $(-\Delta t/8, 0)$; c_1 is the correction at t_1, in range $(-\Delta t/8, \Delta t/8)$; and c_t is the correction at t, in range $(0, \Delta t/8)$.

In Figure 4 we have implicitly assumed that $t_1 > \Delta_o t$. If this is not the case, (see Note 3 of Table 5), one should use:

$$
n(t) = \begin{cases} \lceil n_o \rceil & 0 \le t < t_1 \\ \lceil n_o' + g(t - t_1) \rceil & t \ge t_1 \end{cases}
\qquad (13)
$$

and

$$
\int_o^t n(t')dt' = \begin{cases}
\lceil n_o \rceil\ t & 0 \le t \le t_1 \\
(\lceil n_o \rceil - \lceil n_o' \rceil)t_1 + \lceil n_o' \rceil t & t_1 \le t < t_1 + \Delta_1 t \\
(\lceil n_o \rceil - \lceil n_o' \rceil)t_1 + (t_1 + \Delta_1 t) & t > t_1 + \Delta_1 t \\
\quad \cdot\ (g(t_1 + \Delta_1 t) - 1)/2 & \\
\quad + c_t(t, t_1 + \Delta_1 t) + (n_o' + \frac{1}{2} - g t_1)t + g t^2/2 &
\end{cases}
$$

(14)

Equations (10) and (11) are easily extended to deal with the case of any number of arbitrary additions of bulk data. We give the area for j such additions, assuming that $t_i - t_{i-1} > \Delta t$ for all $i = 1, \ldots, j$, and that $t > t_j + \Delta_j t$:

$$\int_0^t n(t')dt' = a(t, n_0^{(j)}, t_j + \Delta_j t) - \sum_{i=1}^{j} h_i t_i + c_0 + \sum_{i=1}^{j} c_i + c_t(t, t_j + \Delta_j t) \tag{15}$$

with suitable definitions for t_i, $\Delta_i t$, h_i and c_i. Equation (15), and similar equations for special cases, may be used when more than one bulk addition is made, apart from the normal growth of the file. It is, of course, unlikely that more than a limited number of such additions will be made during the lifetime of the file: otherwise the approximations presented here can become almost as cumbersome as a direct summation over the discontinuous n(t).

The penalty for neglecting c_0, c_i, c_t in equation (15) is at worst

$$\pm(j + 1)\Delta t/8$$

Since the total area is roughly $t^2/2\Delta t$ for large t, we see that this is a relative error of

$$\pm(j + 1)(\Delta t/2t)^2$$

The actual error is likely to be much less than this, since individual errors can cancel each other.

The coefficients for equations (11), written as $\int_0^t n(t')dt' = v + wt + zt^2 +$ corrections, are displayed in Tables 6 and 7 for four ranges of t. Note that for the two ranges, $\min_s(\Delta_o t) < t \leq \max_s(\Delta_o t)$

Range of t	Class	File type	No. files	v	w	z	c_o	c_1	c_t [1]
0, $\min_s(\Delta_o t)$	S	1-wd. records	9	–	3	–	–	–	–
(0,217.2)		3-wd. records	1	–	8	–	–	–	–
		Dividends	1	–	5	–	–	–	–
		Stock splits	1	–	1	–	–	–	–
		Div. rates	1	–	3	–	–	–	–
	D	1-wd. records	0	–	0.5	.00027	0	–	11.40 (1±1)
		10-wd. records	1	–	0.5	.00274	0	–	1.14 (1±1)
	Total		14	–	44.50	.00274	0	–	1.14 (1±1)
$\max_s(\Delta_o t)$, t_1	S	1-wd. records	9	–	2.930	.00054	-113.90	–	58.09 (1±1)
(529.7, 720)		3-wd. records	1	–	7.798	.00162	– 32.38	–	19.34 (1±1)
		Dividends	1	–	4.733	.00094	– 47.67	–	33.35 (1±1)
		Stock splits	1	–	1	–	–	–	–
		Div. rates	1	–	3	–	–	–	–
	D	1-wd. records	0	–	0.5	.00027	0	–	11.40 (1±1)
		10-wd. records	1	–	0.5	.00274	0	–	1.14 (1±1)
	Total		14	–	43.40	.01016	-1115.31	–	576.60 (1±1)
$t_1, t_1+\min_s(\Delta_1 t)$	S	1-wd. records	9	-1211.46	5	–	-113.90	–	75.67
(720,897.8)		3-wd. records	1	-2907.69	13	–	– 32.38	–	36.27
		Dividends	1	-1866.46	8	–	– 47.67	–	64.88
		Stock splits	1	–	1	–	–	–	–
		Div. rates	1	– 720	4	–	–	–	–
	D	1-wd. records	0	–	0.5	.00027	0	–	11.40 (1±1)
		10-wd. records	1	–	0.5	.00274	0	–	1.14 (1±1)
	Total		14	-16391.29	71.50	.00274	-1105.15	–	783.33 ± 1.14

Range of t	Class	File type	No. files	v	w	z	c_o	c_1	c_t [1]
$t_1+\max_s(\Delta_1 t),\infty)$ (1521.8, ∞)	S	1-wd. records	9	-671.42	3.863	.00054	-113.90	20.63	58.09 (1±1)
		3-wd. records	1	-2016.47	10.599	.00162	- 32.38	- 1.56	19.34 (1±1)
		Dividends	1	-1169.55	6.357	.00094	- 47.67	21.12	33.35 (1±1)
		Stock splits	1	-	1	-	-	-	-
		Div. rates	1	- 734.9	3.5295	.00048	-	-	64.47 (1±1)
	D	1-wd. records	0	-	0.5	.00027	0	-	11.40 (1±1)
		10-wd. records	1	-	0.5	.00274	0	-	1.14 (1±1)
	Total		14	-9963.70	56.75	.01062	-1105.13	205.21	641.07 (1±1)

Table 6 : Coefficients of $\int_o^t n(t')dt' = v + wt + zt^2 +$ corrections for stock-and day-dependent data, stored on an IBM 2314 DASD

c_t is expressed either as the range $\Delta t/16$ (1±1) or as $c_t(t_1, \Delta_o t)$

Range of t	Class	File type	No. files	v	w	z	c_o	c_1	c_t [1]
$0, \min_s(\Delta_o t)$ (0,507.3)	S	1-wd. records	9	–	2	–	–	–	–
		3-wd. records	1	–	5	–	–	–	–
		Dividends	1	–	3	–	–	–	–
		Stock splits	1	–	1	–	–	–	–
		Div. rates	1	–	2	–	–	–	–
	D	1-wd. records	0	–	0.5	.00015	0	–	20.36 (1±1)
		10-wd. records	1	–	0.5	.00154	0	–	2.03 (1±1)
	Total		14	–	29.5	.00154	0	–	2.03 (1±1)
$\max_s(\Delta_o t)$, t_1 (602.6,720)	S	1-wd. records	9	–	2	–	–	–	–
		3-wd. records	1	–	4.583	.00090	– 21.04	–	34.57 (1±1)
		Dividends	1	–	2.868	.00052	–110.92	–	59.61 (1±1)
		Stock splits	1	–	1	–	–	–	–
		Div. rates	1	–	2	–	–	–	–
	D	1-wd. records	0	–	0.5	.00015	0	–	20.36 (1±1)
		10-wd. records	1	–	0.5	.00154	0	–	2.03 (1±1)
	Total		14	–	28.95	.00297	–131.96	–	96.21 (1±1)
t_1, $t_1 + \min_s(\Delta_1 t)$ (720,746.9)	S	1-wd. records	9	– 720	3	–	–	–	
		3-wd. records	1	–1271.68	7	–	– 21.04	–	65.46
		Dividends	1	–1263.17	5	–	–110.92	–	51.47
		Stock splits	1	–	1	–	–	–	
		Div. rates	1	– 720	3	–	–	–	
	D	1-wd. records	0	–	0.5	.00015	0	–	20.36 (1±1)
		10-wd. records	1	–	0.5	.00154	0	–	2.03 (1±1)
	Total		14	–9734.85	43.50	.00154	–131.96	–	118.96 ±2.03

Range of t	Class	File type	No. files	v	w	z	c_o	c_1	c_t [1]
$t_1 + \max_s(\Delta_1 t)$, ∞	S	1-wd. records	9	$-$ 610.67	2.3822	.00030	–	–	103.78 (1±1)
$(1623 \cdot 3, \infty)$		3-wd. records	1	$-$1128.11	6.150	.00090	$-$ 21.04	52.68	34.57 (1±1)
		Dividends	1	$-$ 654.30	3.777	.00052	$-$110.92	36.77	51.61 (1±1)
		Stock splits	1	–	1	–	–	–	–
		Div. rates	1	$-$ 594.48	2.1948	.00027	–	–	115.24 (1±1)
	D	1-wd. records	0	–	0.5	.00015	0	–	20.36 (1±1)
		10-wd. records	1	–	0.5	.00154	0	–	2.03 (1±1)
	Total		14	$-$7872.92	35.06	.00595	$-$131.96	89.45	1145.44 (1±1)

Table 7 : Coefficients of $\int_o^t n(t')dt' = v + wt + zt^2$ + corrections for stock- and day-dependent data, stored on an IBM 3330.

[1] See note 1 of Table 6.

and $t_1 + \min_s(\Delta_1 t) < t \le t_1 + \max_s(\Delta_1 t)$, the total does not have a simple form: these ranges have been omitted, but can easily be included using interpolation.

Table 8 combines the data of Tables 6 and 7 into expressions for the total area for class S and class D files.

2314		3330	
t	$\int_0^t n(t')dt'$	t	$\int_0^t n(t')dt'$
(0, 217.2)	$1.14 + 44.5t + .00274t^2$ ± 1.14	(0,507.3)	$2.03 + 29.5t + .00154t^2$ ± 2.03
(529,7, 720)	$-538.71 + 43.4t + .01016t^2$ ± 576.60	(602.6, 720)	$-35.75 + 28.95t + .00297t^2$ ± 96.21
(720, 897.8)	$-16713.11 + 71.5t + .00274t^2$ ± 1.14	(720, 746.9)	$-9747.85 + 43.5t + .00154t^2$ ± 2.03
(1521.8, ∞)	$-10222.55 + 56.75t + 01061t^2$ ± 641.07	(1623.3, ∞)	$-6769.99 + 35.06t + .00595t^2$ ± 1145.44

Table 8. Integrated number of blocks in all class S and D files as a function of lifetime, t.

1.3.1.2 Stock-day-dependent and Overflow Data (Classes SD and O)

There are three groups of files in these two classes. STOCKS consists of two files of 1-word records: a track or set of tracks is allocated every day to contain information for that day for the entire range of stocks. STOX also consists of two files of 1-word records: information for 800 stocks is maintained on 800 tracks, with each track containing all the historical data for a given stock. The overflow area is a file of 3-word records: Table 3 tells us that 33% of all stocks add to this file at the rate of one record per year, 9% at two records per year and 48% at four records per year. The growth rate is thus $2.43*N_s(t)$ words per year, where $N_s(t)$, the number of stocks at time t, is specified in (1).

STOCKS

The number of blocks occupied by each file in STOCKS is just the time integral of the size of the corresponding stock-dependent file. Thus, using (11) and (14), we obtain table 9, for the total size of the two STOCKS files.

The integral $\int_0^t n(t')dt'$, can be obtained easily from Table 9.

STOX

The number of blocks occupied by each file on STOX is $800*\lceil N_D(t)/\nu \rceil$ so that

$$n(t) = 1600 \lceil t/\nu \rceil \tag{16}$$

for STOX. The integral

$$\int_0^t n(t')dt' = 1600(t/2 + t^2/2\nu + \nu t/16 \pm \nu t/16) \tag{17}$$

2314		3330	
t	$n(t)$	t	$n(t)$
(0, 529.7)	$6t$	(0, 720)	$4t$
(529.7, 720)	$-111.62 + 5.86t + .00108t^2 \pm 116.18$	(720, 1956.2)	$-1440 + 6t$
(720, 1521.8)	$-2498.38 + 10t$	(1956.2, ∞)	$-1013.78 + 4.764t + .00060t^2 \pm 207.56$
(1521.8, ∞)	$-1413.2 + 7.726t + .00108t^2 \pm 116.18$		

Table 9. Number of blocks in the STOCKS files as a function of lifetime, t.

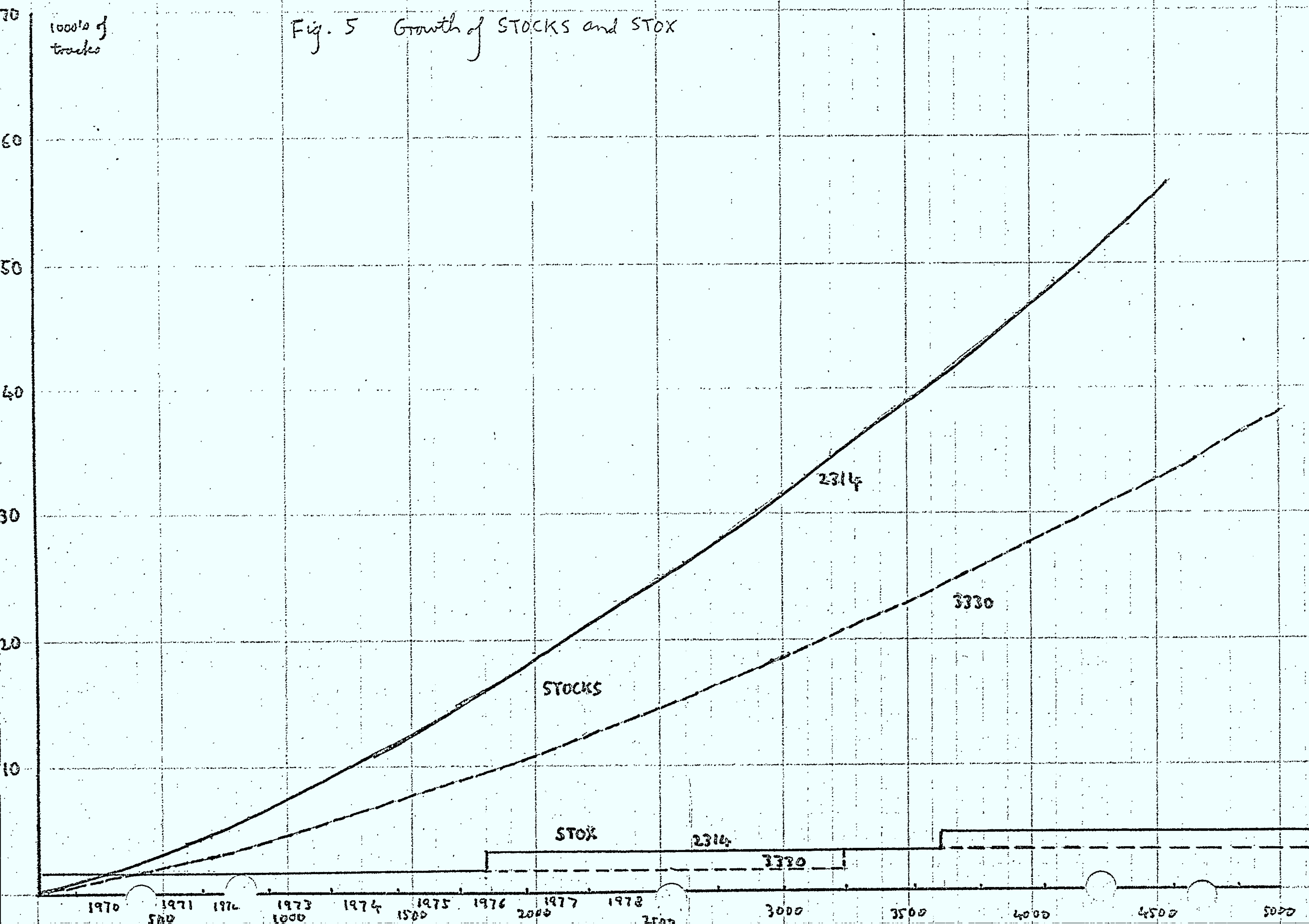


Fig. 5 Growth of STOCKS and STOX

can alternatively be evaluated as a sum, in view of the large size of ν for 1-word records.

The quantities, n(t), are plotted in Figure 5 for STOCKS and STOX. We see that the growth of STOCKS dominates the databank, and that the quadratic terms do not have a very pronounced effect.

Overflow

The overflow data will require

$$n(t) = \lceil 2.43 * N_s(t)/\nu * t/260 \rceil \tag{18}$$

blocks (using Table 3) which gives the expression shown in Table 10, using (1) and (2).

t	2314	3330
(0, 720)	$.0227t + .00000101t^2$	$.0127t + .00000057t^2$
(720, ∞)	$.0242t + .00000101t^2$	$.0135t + .00000057t^2$

Table 10. Number of blocks in Overflow file as a function of lifetime, t.

The integral, $\int_0^t n(t')dt'$, can be obtained easily from table 10.

1.3.2 Acquisition Cost

The acquisition of data for the FRI databank can be considered under three categories: the initial acquisition, the acquisition of new stocks and the daily acquisition of data for all stocks. The first happens only once, the last happens daily and the acquisition of new stocks happens 224 times a year on the average, assuming a constant rate of 510 new stocks per year of 260 days. This number can be arrived at by the following argument.

We assume that new stocks are created at random with a distribution that is uniform over the whole year. Updates to the databank are performed at most daily. The probability that a new stock is created on any given one of the y=260 days of the year is $1/y$. If r_a=510 such stocks are created in the year, it can be shown that the expected number of days on which stocks (see §1.3.4.1) are created is

$$y\left(1 - \left(\frac{y-1}{y}\right)^{r_a}\right) \qquad (19)$$

giving 224.

The following sections elaborate the cost of each of the three categories of data acquisition. This cost is computed in terms of time as well as in terms of number of accesses, since both methods of charging may be applicable. In the following, the algorithms for the data acquisition are conceived in terms of a straightforward implementation which ignores some of the constraints to which the FRI is subjected. In particular, by allowing somewhat more than 100k of core for the programs, we can reduce the number of accesses to a minimum. Thus the results of this analysis are a lower bound to, rather than a true estimate of, the cost.

1.3.2.1 . Initial Acquisition

The description of a single new stock can be contained in the 80 columns of a punched card, eg. as shown below,

Stock name	12 col
Dividend: amount, data	10 col
Stock Split: amount, data	10 col
Dividend rate: amount, data	10 col
Exchange	1 col
Type	1 col
Number of shares	7 col
Volume	5 col
Dollar volume	7 col
Ticker Symbol	3 col
	66 col

We assume, then, that the 4430 original stocks are input to the initial acquisition program of 4430 cards at a spooling charge of C_Y per 1000 cards. The buffers required for this are shown in Fig. 6, together with the buffers required for output of each of the 13 stock-dependent files to be created. Four of these files have three-word records and 4430 records require eight 2314 tracks each. Six have one-word records and require three tracks each. The ticker symbol and location files must be simultaneously sorted on ticker symbol before they can be stored, and space has been allocated to hold all 4430 values of these fields for an in-core sort. The ticker symbol is written twice, accounting for the extra three tracks output. The time of 2.4 seconds is worked out assuming an average seek time of 72.5 ms for each of the 13 files plus a data transfer time of 8*25 ms for the eight-track files and 3*25 ms for the three-track files. This estimate is probably low. The program is used once.

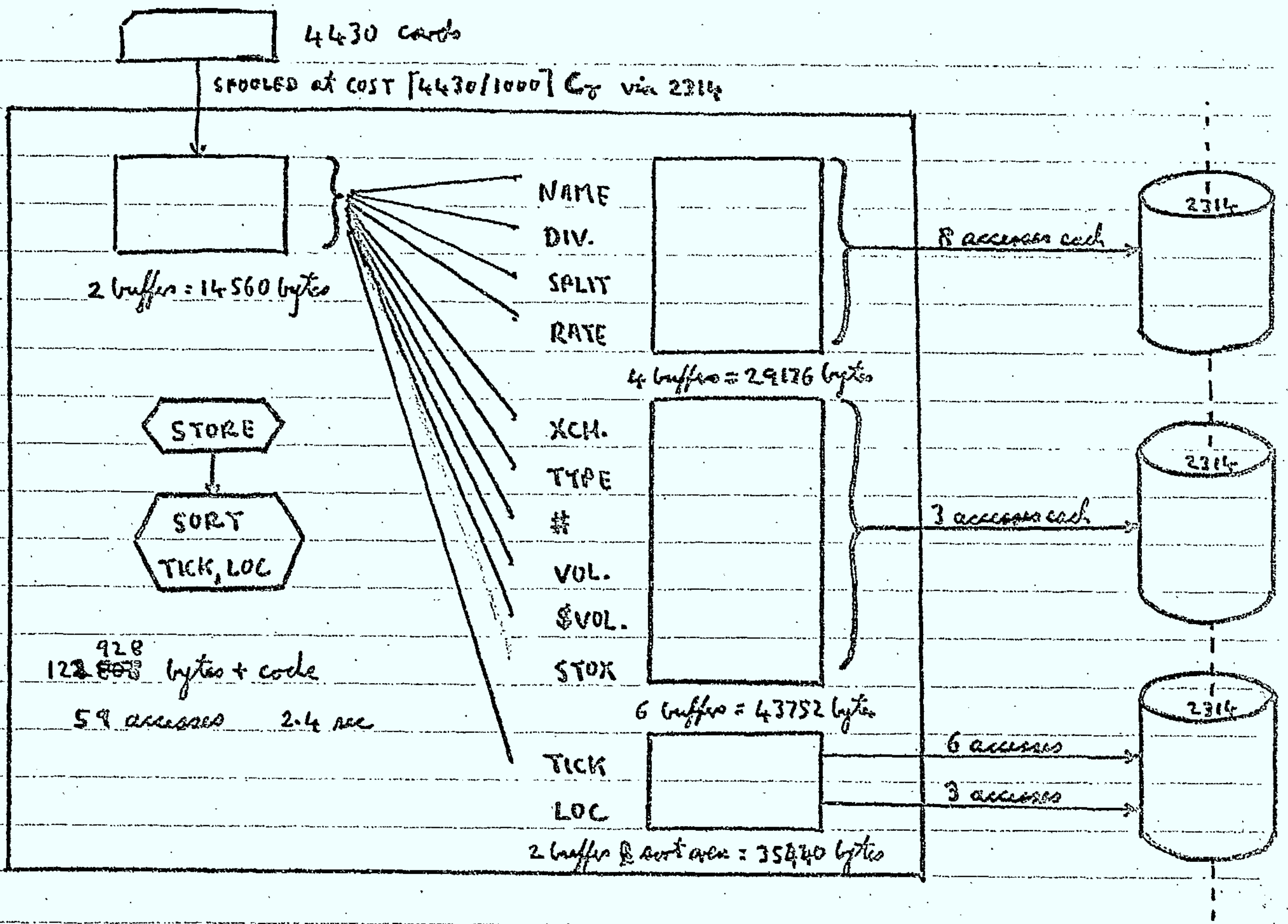


Fig. 6. Initial acquisition: program to store data, allocate STOX location, sort TICK & LOC

1.3.2.2 New Stocks

New stocks are assumed to be punched one per card in the same way as the original stocks. The program of Fig. 7 will handle batches of up to 1823 new stocks, and so can be used both for the bulk acquisition of 1700 bonds and for the regular acquisition of 510/224 = 2.28 (on an average) new stocks on each update. The program is the same as that for initial acquisition, except that the space allocated for sorting TICK and LOC is smaller and that these records must be merged into the existing TICK and LOC fields. The costs in parentheses are for the bulk addition of 1700 bonds. For regular acquisitions, average values are used for number of accesses because there is a probability $2.28/\nu$ that an update will complete a track, where $\nu = 607$ 3-word records on a 2314 track and $\nu = 1823$ 1-word records on a 2314 track. The time of 1.5 + 0.1a seconds (2.4 seconds for the bulk addition) has been worked out under similar assumptions to those for the initial acquisition.

The program is used 224 times a year, or 0.86 times per day. This gives a total cost of

$$.86*22.026t + 4 \sum_{i=1}^{T} \lceil N_s(t_i)/1823 \rceil \quad \text{accesses}$$

or

$$.86*1.5t + 0.1 \sum_{i=1}^{T} \lceil N_s(t_i)/1823 \rceil \quad \text{seconds}$$

over a span of $t = T\Delta t$ days where $\Delta t = 1/0.86$, the average number of days between updates. The summation $\sum_{i=1}^{T} \lceil N_s(t_i)/1823 \rceil \Delta t$ can be replaced by the integral $\int_0^t n(t')dt'$, whose coefficients are given for 1-word stock-dependent files in Table 6. Thus the total cost of adding new stocks to the databank for a period of t days from inception is

$$.86(-2685.68 + 37.478t + .00216t^2) \text{ accesses}$$

or

$$.86(-67.142 + 1.8863t + .000054t^2) \text{ seconds.}$$

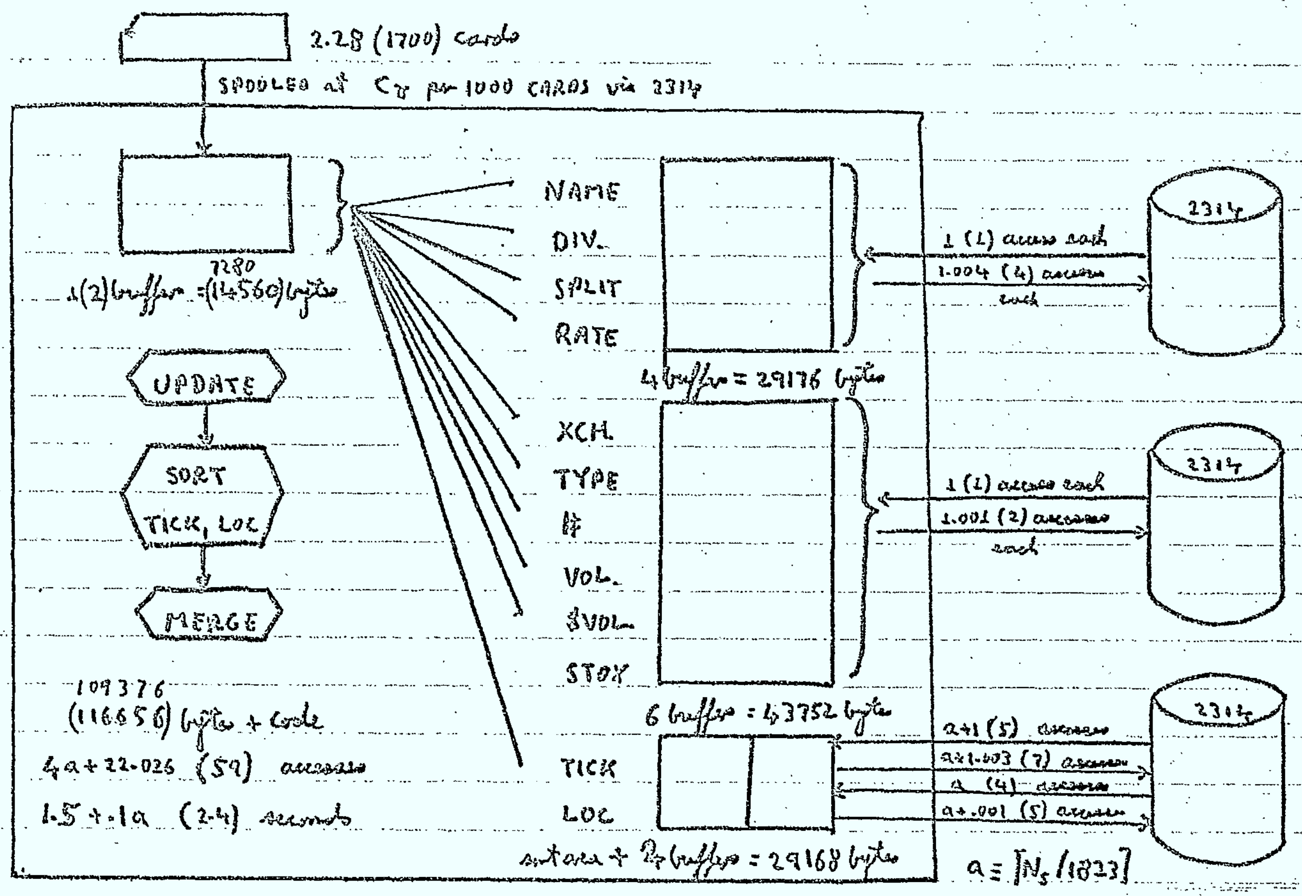


Fig. 7. New stocks: program to update stock-dependent data, sort new TICK & LOC & merge with old.

(These figures are valid for $t \geqslant t_1 + \Delta_1 t$ = 1522 days i.e. from about April 1975.) Using C_α = 0.167¢ per access and C_τ = 15¢ per second, the costs are

$$- \$3.86 + \$0.0536t + \$0.00000309t^2$$

or

$$- \$8.65 + \$0.243t + \$0.00000696t^2$$

respectively.

1.3.2.3 Daily Data

Daily update information for five securities may be punched onto a single 80-column card, e.g. as shown below.

Ticker Symbol	3 col
Price	5 col
Volume	6 col
	14 col

The first seven input records contain data on the seven stock indices maintained by FRI.

The first of the two programs of Fig. 8 sorts the input into order of ticker symbol, uses this sequence to look up the location, then sorts the update data into location order. Using the analysis of §1.2.5.1.1 we conclude that 1-pass sorts are adequate for all stocks in the foreseeable future: this dictates the number of accesses to the work files. In terms of time the cost is low because it is possible to overlap I/O.

The second program uses the price and volume data on the seven stock indices plus information which can be generated internally to add a new record to the Day Index File. The two files STOCKS and STOX are then

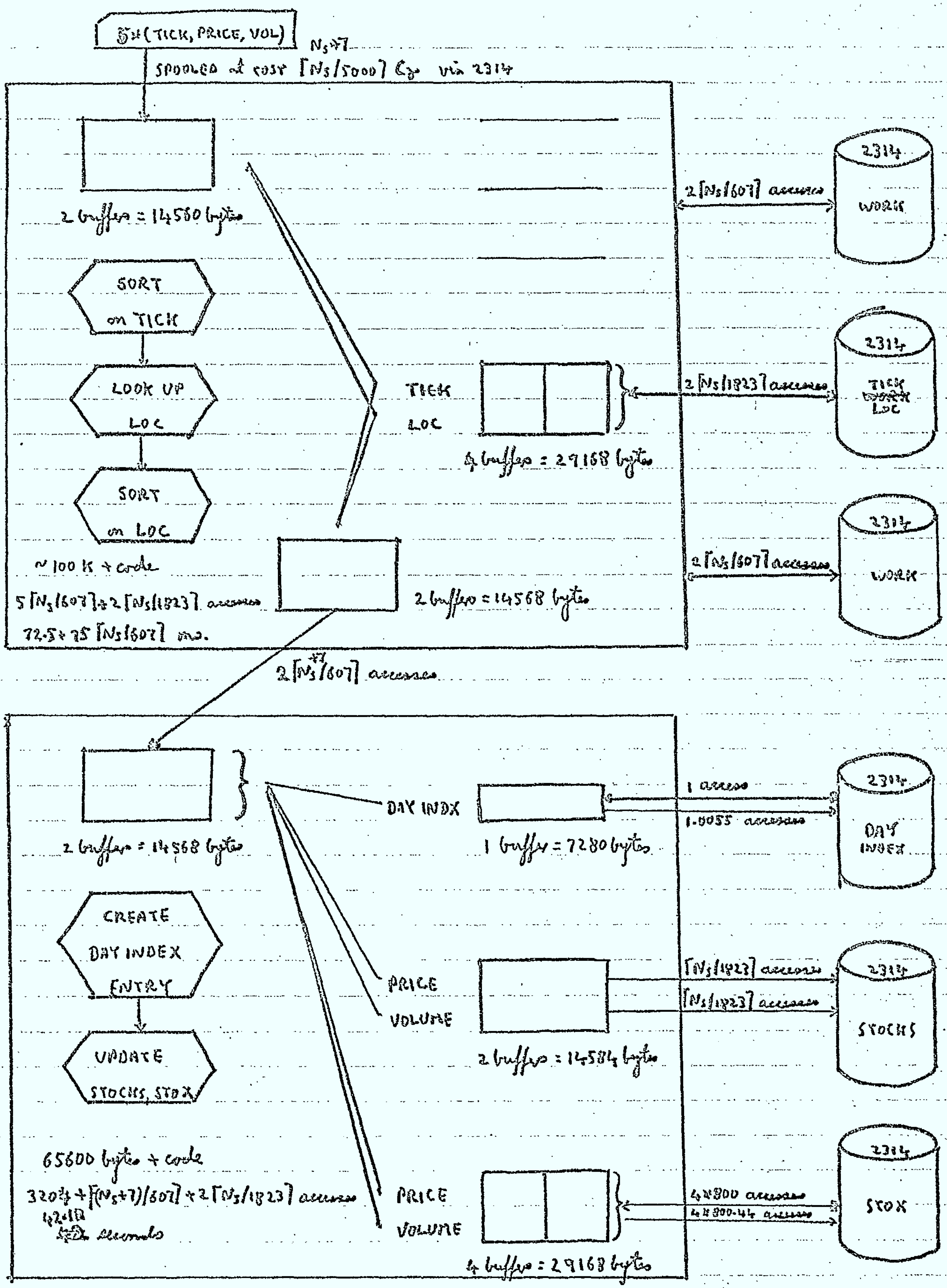


Fig. 8. Daily Data: programs to sort and add daily price and volume information to databank.

updated. Overlapped I/O again simplifies the time calculation, and it is plain that the updates to the 1600 tracks of STOX will dominate.

From Table 6, the coefficients of $\int_0^t n(t')dt'$ may be used to obtain the cost over a period of time, t, from inception of the daily updates. For $\lceil N_s/1823 \rceil$ and $\lceil N_s/607 \rceil$ we use the coefficient for 1-word and 3-word stock-dependent records respectively, and obtain for the two programs in Fig. 8,

$$3204t + 4\int_0^t n_1(t')dt' + 6\int_0^t n_3(t')dt' = -14784.5+3283.046t+0.01188t^2 \text{ accesses}$$

or

$$42.15t + 3/40\int_0^t n_3(t')dt' = -151.236+43.005t+0.000122t^2 \text{ seconds}$$

(Valid from t=1522 days.)

Using C_α = 0.167¢ per access and C_τ = 15¢ per second, the costs are

$$- \$24.60 + \$5.45t + \$0.0000198t^2$$

or

$$- \$22.70 + \$6.45t + \$0.0000183t^2$$

respectively.

1.3.2.4 Total Acquisition Costs

The total of the three contributions is

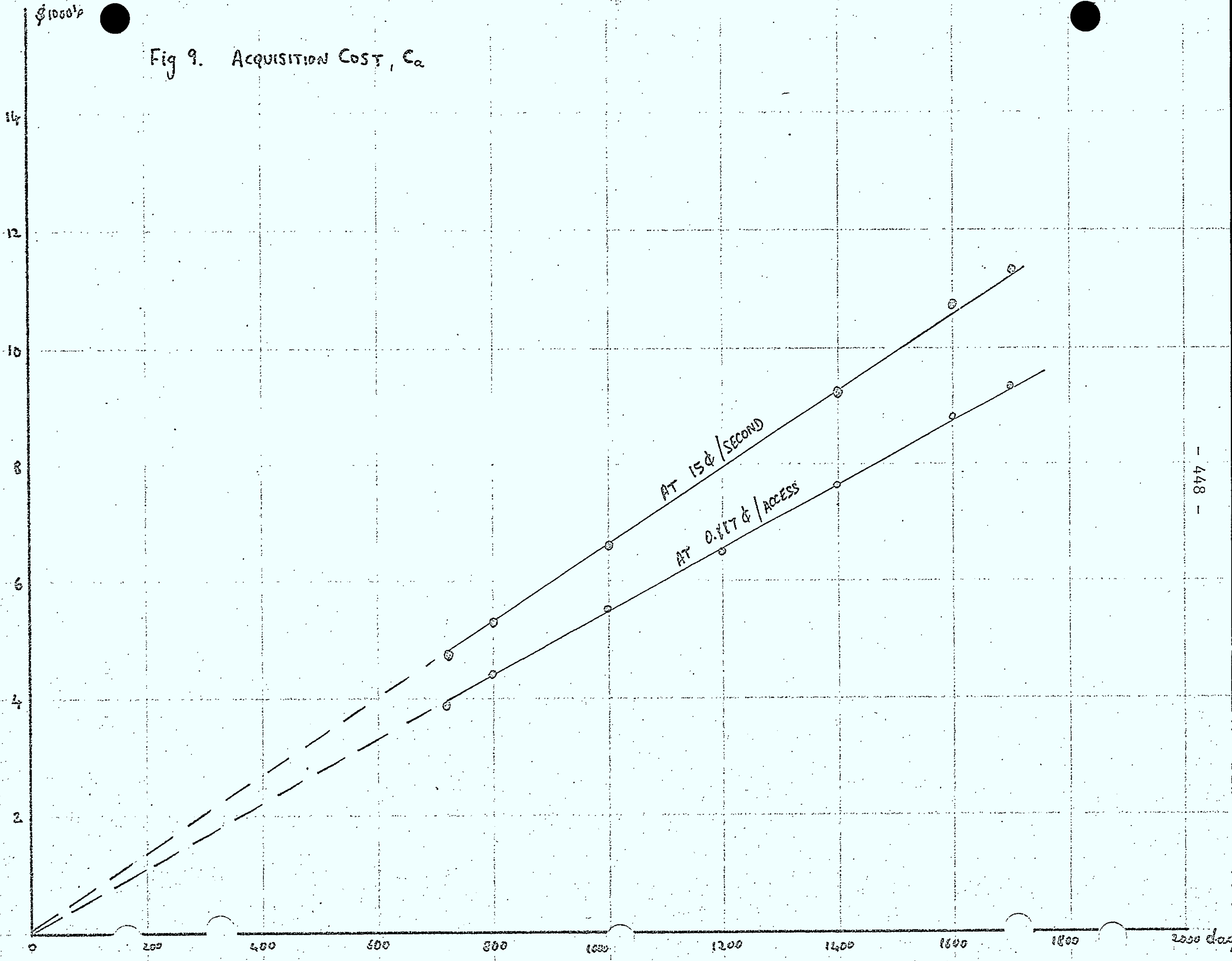
$$C_a = -\$28.36 + \$5.50t + \$0.0000229t^2 \quad (20a)$$

if the charges are made by access at 0.167¢ per access, or

$$C_a = -\$30.99 + \$6.69t + \$0.0000250t^2 \quad (20b)$$

if the charges are made by I/O time requirements at 15¢ per second. These total costs are plotted as a function of the databank age in Fig. 9.

Fig 9. Acquisition Cost, C_a



- 448 -

1.3.3 Storage Cost

The storage cost is given in §1.2.5.2 as

$$C_s = \frac{C\sigma}{b} \int_o^t n(t')dt'$$

where $C_\sigma = 4¢$/track/day, $b = 1$ is the number of blocks per track and $n(t)$ is the total number of blocks in the file as a function of time. The coefficients can be found in or from Tables 8, 9 and 10 and Fig. 5. The total is

$$C_s = -\$408.902 - \$54.262t + \$0.156t^2 + \$0.0000144t^3 + \begin{cases} \$32t & 1522<t<1823 \\ \$64t - \$58336 & 1823<t<3646 \end{cases} \quad (21)$$

on a 2314.

It should be noted that this formula does not describe present FRI expenditure on storage, since FRI pays a fixed weekly amount of \$665 per 3330 disk pack. The cost is thus.

$$C_s = \begin{cases} \$133t & t<1300 \\ \$266t - \$172900 & 1300<t<2300 \end{cases}$$

Both these total costs are plotted as a function of the databank age in Fig. 10.

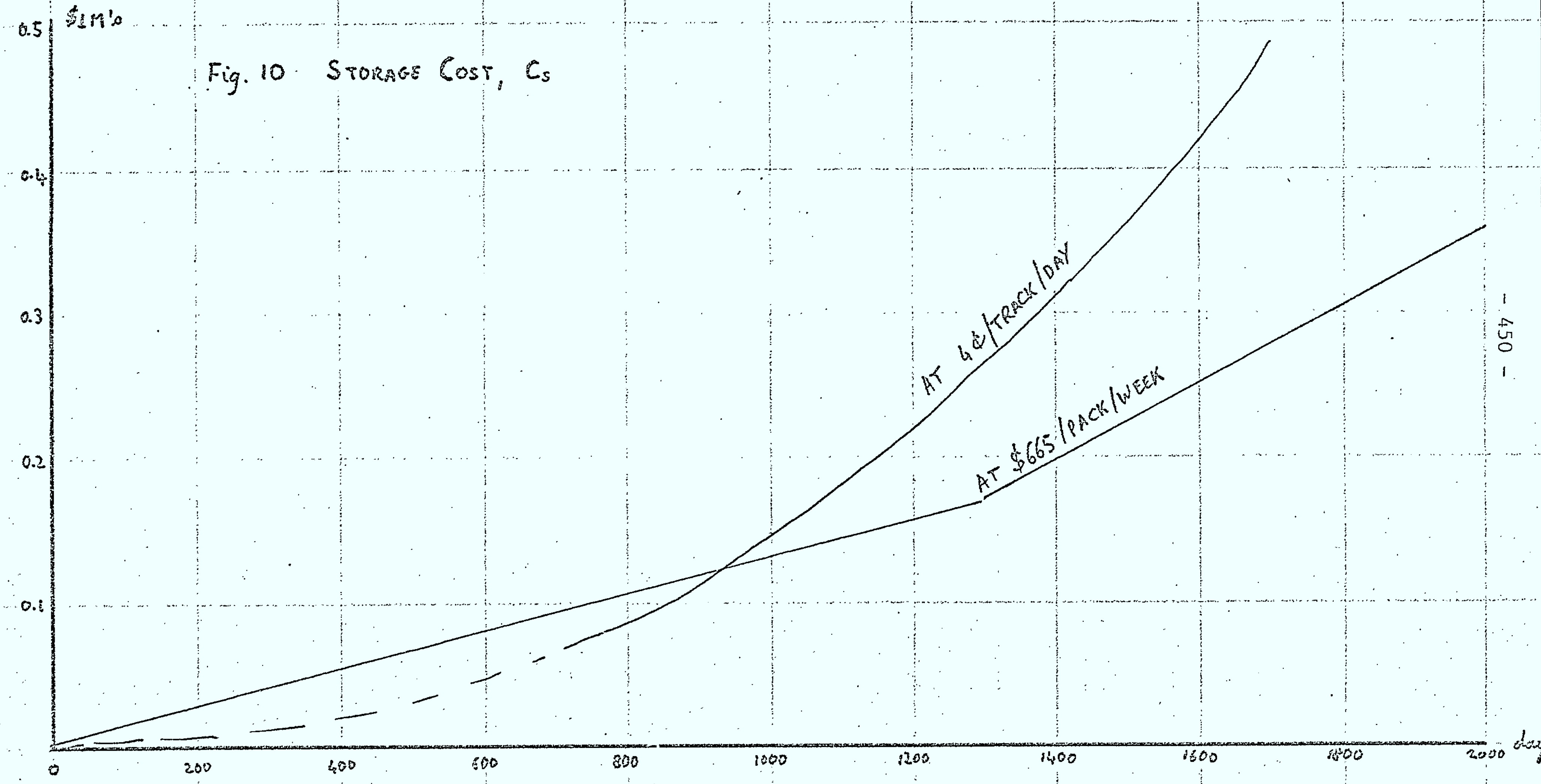


Fig. 10 STORAGE COST, C_s

1.3.4 Retrieval Cost

The cost of retrieving information from the daily stock exchange data bank is borne directly not by FRI but by the individual members who pay for the execution of their retrieval programs. Lack of usage statistics makes detailed analysis impossible, so what follows is an exploration of various assumptions that could be made about the usage distribution.

1.3.4.1. Treatment of Usage Distribution for Direct Organization

We consider a directly organized file occupying n blocks. If we know the probability, u_i, of accessing block i, $i = 1,\ldots,n$, in a request, we can find the expected number, $\overline{X}_r$, of blocks accessed in a batch of r requests. We define

$$X_i = \begin{cases} 0 & \text{if block } i \text{ is not accessed} \\ 1 & \text{if block } i \text{ is accessed.} \end{cases}$$

For r requests $\quad X_i = 0$ with probability $(1-u_i)^r$

$\quad X_i = 1$ with probability $1-(1-u_i)^r$

and thus

$$\overline{X}_r = \sum_{i=1}^{n} 0 * (1-u_i)^r + 1 * (1-(1-u_i)^r)$$

$$= \sum_{i=1}^{n} 1-(1-u_i)^r. \tag{22}$$

For example, if the usage distribution is uniform, $u_i = {}^1\!/n$,

$$\overline{X}_r = n\left(1-\left(\frac{n-1}{n}\right)^r\right). \tag{23}$$

(This result was used in §1.3.2 in a context of days in a year instead of blocks in a file.) It is easy to see that the uniform distribution constitutes the worst case for direct organization: any non-uniform distribution will require fewer accesses on the average.

A family of distributions which will be of use to us is the "abc" family. A special case is the "80-20" distribution in which 20% of the data on the file is used 80% of the time, and within this 20%, 20% of the data on the file is used 80% of the time (i.e. 4% of the data on the file is used 64% of the time) etc. This distribution has the form (KNUTH Sorting & Searching §6.1 p.397)

$$U_i = (i^\Theta-(i-1)^\Theta)/n^\Theta \qquad (5)\ (24)$$

where $\Theta = \ln.8/\ln.2$ for the 80-20 distribution

and $\Theta \varepsilon (0,1]$ for the abc family.

It is easy to see that the uniform distribution is given by the special case of $\Theta= 1$. For non-uniform distribution in the abc family, the expected number of access, (22) does not simplify readily.

1.3.4.2. <u>STOCKS and STOX</u>

We examine first the major groups of files in the FRI databank, STOCKS and STOX. As classified above, both consist of stock-day-dependent files, and so have two - dimensional usage distributions, with probabilities u_i^s and u_j^d. The product of these probabilities, $u_s u_d$, is the probability that track (s,d) is accessed in a single request. The file is partitioned into N_s partitions on the stock access and n_d partitions on the day access, so that there are $n_s\ n_d$ blocks in all. This is illustrated in Fig.11.

The expected number of accesses for a batch of r requests becomes

$$\bar{X}_r = \sum_{i=1}^{n_d} \sum_{j=1}^{n_s} 1-(u_i^d\ u_j^s)^r \qquad (25)$$

where, with a growing file, n_s generally depends on i.

1.3.4.2.1. STOX is the easier to analyse: $n_d = 1$ (for the first 1823 days of the life of the databank) and $n_s = 800$. Fig. 12 shows the effect of setting

$$u_j^s = (j^\Theta - (j-1)^\Theta)/n_s^\Theta \qquad j=1,\ldots,n_s$$

For various values of Θ_s on $\bar{X}r/r$ the average number of accesses per request.

We can see that the cost will be very dependent on the usage distribution, u_i^s, and on the number of requests, r, in a batch. Through the latter dependence, the cost depends strongly on the distribution of batch sizes, s_r - whether all batches have the same number, r, of requests, or whether the number of requests varies significantly from user to user. Thus, to permit further analysis of STOX, two distributions must be known: the stocks usage distribution and the batch size distribution. Then the average number of accesses for all batches in

$$\bar{X} = \sum_{r=1}^{\infty} s_r \bar{X}_r \qquad (26)$$

The result (26) must be multiplied by a suitable factor to include the accesses required to locate the appropriate block of STOX. This is done by a series of indexes, described in Tables 1 and 2. If we assume that only a ticker symbol is provided with each request, two additional accesses are required, one for "location" and one for "STOX pointer." Thus the average number of accesses for prices or for volumes is $3\overline{X}$ and the average number of accesses if prices and volumes are both requested is $4\overline{X}$.

Finally, we must multiply by the request rate, $r(t)$, as defined in § 1.2.5.3.2, the number of batches per unit time retrieved at time t. For FRI, $r(t) = r_q(t)$ since the user is not permitted changes or deletions. This gives the retrieval cost

$$C_r = \int_0^t 3\overline{X}\ C\ \alpha\ r_q(t')\ dt' \qquad (27a)$$

or

$$C_r = \int_0^t 3\overline{X}\ B\ \omega\ \tau\ C_\tau\ r_q\ (t')\ dt' \qquad (27b)$$

if we are costing by the time required, where $B = 7292$ bytes per block, $\tau = 3.43 * 10^{-6}$ seconds per character for the 2314 and the overhead ratio, $\omega = 3.9$, if we include arm movement time.

In order to estimate the approximate magnitude of the retrieval cost for STOX, we carry out the analysis assuming a uniform usage distribution, $u_i^s = 1/n_s$, a Poisson batch size distribution, $s_r = \rho^r\ e^{-\rho} / r!$, where ρ is the mean batch size, and a request rate which is proportional to the amount of data in the databank, i.e. essentially proportional to the size of STOCKS, $r_q(t) = C\ (-1413.2 + 7.726t + 0.00108t^2)$ (for 2314 with, strictly speaking, $t > 1522$ days). Then

$$\overline{X} = \sum_{r=1}^{\infty} s_r \overline{X}_r = n_s\ (1 - e^{-\rho/n_s}).$$

Thus

$$C_r = 3 \begin{Bmatrix} C\alpha \\ C_\tau \beta\omega\tau \end{Bmatrix} \rho \int_0^t r_q(t')dt' \qquad (28)$$

where $C\alpha = 0.133$ ¢ per access and $C_\tau = 15$¢ per second.

The quantity $\rho\ r_q(t)$ is interpreted simply as the average number of requests per day. We have two interesting consequences of the uniform usage distribution and Poisson batch size distribution: the cost is independent of the number of blocks, n_s, (for large n_s) and the cost is independent of the batch size.

The quantities

$$C_r/\rho C = 3\begin{Bmatrix} C\alpha \\ C_\tau \quad \beta\tau \end{Bmatrix} (-1413.2t+3.863t^2+0.00036t^3)$$

are plotted in Fig. 14. The daily average number of requests is assumed proportional to the size of STOCKS, and we can suppose that it is no greater than the number of tracks in STOCKS at any point in time (e.g. 10,000 tracks after 1250 days: 10,000 requests per day is a large number). Thus the constants $\rho C < 1$ (or even $\rho C << 1$), which means that the retrieval cost, C_r, is smaller (possibly by orders of magnitude) than the curves of Fig. 14.

1.3.4.2.2 STOCKS, as Fig. 11 indicates, has a more complicated structure, with n_s depending on i in (25):

$$n_s = \begin{cases} 3 \text{ blocks} & 0 < i \leq 529 \\ 4 \text{ blocks} & 529 < i \leq 720 \\ 5 \text{ blocks} & 720 < i \leq 1521 \\ 6 \text{ blocks} & 1521 < i \leq 2451 \\ \text{etc.} & \end{cases}$$

Consequently, u_j^s depends on i - e.g. for a uniform distribution $u_j^s = 1/n_s\ j = 1,\ldots,n_s$. Fig. 13 shows the effect of setting

$$u_1^d = 1/n_d \qquad i = 1,\ldots,n_d$$

and

$$u_j^s = (j^\theta-(j-1)^\theta)/n_s^{\ \theta} \qquad j = 1,\ldots,n_s$$

for various values of θ on $\overline{X}_r/r$, the average number of accesses per request.

As with STOX, we need to know the usage distributions, u_i^d and u_j^s, the batch size distribution, s_r, and the request rate $r(t) = r_q(t)$ in order to complete the analysis. As with STOX, we have no knowledge of the actual distributions and rates, and so we make a sample analysis using the most expensive (uniform) distribution and a plausible request rate. Thus, with $u_i^d = 1/n_d = 1/t \; \forall_i$, $U_j^s = 1/n_s \quad \forall_j$, $s_r = \rho^r e^{-\rho}/r!$ and $r_q(t) = C\,(-1413.2 + 7.726t + 0.00108\,t^2)$ as before we have

$$\overline{X} = \sum_{r=1}^{\infty} s_r X_r = \sum_{r=1}^{\infty} \rho^r e^{-\rho}/r! \sum_{i=1}^{t} n_s \left(1-\left(\frac{tn_s - 1}{tn_s}\right)^r\right)$$

$$= \sum_{i=1}^{t} n_s (1-e^{-\rho/tn_s})$$

$$\simeq \rho \qquad \text{for large } tn_s.$$

This gives the same result for C_r as (28),

$$C_r = 3\begin{Bmatrix} C\alpha \\ C_\tau B\omega\tau \end{Bmatrix} \rho \int_0^t r_q(t')dt',$$

which is plotted in Fig. 14. The discussion of results for STOX in § 1.3.4.2.1 applies here.

1.3.4.3 Files other than STOCKS and STOX

The retrieval analysis of stock-dependent and day-dependent files is straightforward and gives retrieval costs that are negligible in comparison with C_r for STOCKS and STOX. It is not presented here.

The overflow file has a chained organization and is logically a sequential file. Thus the analysis of §1.2.5.3 can be applied. If the file consists of unblocked records, the application is straightforward, with δ accesses required to retrieve the δth record in the file. If the records are blocked, there are two alternatives: either the records are stored in the order in which they are acquired or else the files are periodically rearranged so that all records pertaining to a given stock and overflow type are stored together in a single block.

If overflow records are stored in blocks in the order in which they are acquired, Table 3 shows that each block will contain the following:

13.6% of the block will contain stocks with 1 addition per year
7.4% of the block will contain stocks with 2 additions per year
79% of the block will contain stocks with 4 additions per year

Since the data consists of 3-word records, we have the following numbers of records of each type per block.

TYPE	IBM 2314	IBM 3330
annual	82	147.5
semi-annual	45	83.5
quarterly	480	867

Since the overflow must accommodate 30% of $3N_s$ files, the average number of records per file of each of these types is obtained by dividing these figures by .9Ns. We can see that at all times there is considerably less than 1 record per block for a file of any type, so that the analysis of blocked overflow records will give the same result as the analysis of unblocked overflow records (except that storage space is better used).

The records can be periodically rearranged so that each file is completely contained within a block: the 607 records per block of the 2314 will hold

$$\left\lfloor \frac{607}{4t/260} \right\rfloor = \left\lfloor \frac{39455}{t} \right\rfloor$$

quarterly files when the databank is t days old. Then only one access is required for an overflow record in any given file.

The detailed retrieval analysis of the overflow file is not presented here because the costs are negligible in comparison with C_r for STOCKS and STOX.

1.3.5 Maintenance Cost

The file organizations are such that in principle no maintenance is needed except for that suggested above for the overflow data. In practice, however, various activities take place that can be classified as maintenance. Errors in the input data can escape editing, and on-line programs are needed to patch the data. Storage space limitations sometimes necessitate temporary storage of data off-line.

If the overflow data is organized according to the second scheme described in §1.3.4.3, it must be copied periodically to incorporate new records into the home block of the file. This costs

$$2 \sum_{i=1}^{T} n\,(i\Delta t) \qquad (29)$$

accesses, where the relations between t, T and Δt are summarized in §1.2.5.1.2 and $n(t)$ is given by (18). In this case, Δt is the interval (perhaps 1 year) between reorganizations of the file. For a reasonably well-behaved function, $N_s(t)$, (29) can be taken as the midpoint rule approximation to the integral

$$2/\Delta t \int_0^t 2.43\ t' N_s(t')\ dt\ /2600$$

and the maintenance cost assessed from this.

1.3.6 Summary of FRI Costs

We can list the four categories of cost in descending order of size: storage cost, retrieval cost, acquisition cost and maintenance cost.

Storage Cost. The storage cost is high because of the requirement that data be instantly accessible to the user. This means storing all data online on a direct access device, and it also means duplicating some data in STOX to provide rapid access to time series. The current allocation of space to STOX adds about 20% to the total cost (at 1000 days); duplicating all data in STOX would almost double the storage requirements.

A remote alternative to duplicating data in STOX would be to develop software and hardware capable of accessing data either by track in the conventional fashion or across tracks by electronic switching of heads. It would be necessary to use, in this application, some fairly sophisticated low-level processing to identify a time series of a particular stock stored in the same location on many tracks. This could not be done efficiently by existing direct search methods on DASD because it would mean reorganizing each record to carry keys and to be stored unblocked: on a 2314 such a reorganization would reduce the number of records per track from 1823 to 46.

Retrieval Cost. We have been unable to specify the retrieval cost with any precision because of unknown usage distributions and request rates. Furthermore, the expense of retrieval is borne directly by the users. However, the analysis we have been able to give indicates that retrieval cost is second in magnitude to the storage cost but greater than acquisition cost. Reduction of retrieval cost is unlikely since the databank is designed to minimize retrieval time and cost. Some suggestions, which depend on a knowledge of usage distributions and request rates, follow.

Increasing the size of STOX, either by duplication or as an alternative to STOCKS, might reduce average retrieval costs for all users if users generally require time series as well as or instead of market cross-sections. However, duplication would cause increases in storage costs, and replacement of STOCKS by STOX would increase the acquisition cost. (The technique suggested above for reducing storage cost would not have these disadvantages).

The storage organization used is that of a transposed file: all fields of a given record are separately stored and individually accessible using a single address for the record. If users are frequently interested in more than one field per record, the more conventional record/field structure would require only one access per record, although fewer records could be stored and accessed on one track.

<u>Acquisition Cost</u>. The databank is organized to make data acquisition very economical. The only inefficient process is adding to STOX, since data input as a market cross-section must be converted to time series.

1.3.7 Criterion for introducing additional file arrangements

This section on the analysis of data bank costs would not be complete without reference to the following question. Since ease of accessing a data bank without a detailed knowledge of programming, delay in obtaining service, and the efficiency of the accessing program in terms of fully exploiting the data store (recombining or restructuring its elements) all depend on file organization, what is the decision rule for deciding when an additional file arrangement should be made available to users?

Setting up a file in a different order involves adding to fixed costs of the data bank in order to reduce variable costs, principally variable costs of users. The fixed costs comprise design of the new file arrangement after consultation with users, writing of the basic accessing program, and provision of program documentation. The reduction in variable costs accrues principally to users, in reduced computing costs and faster service.

Adding a new file arrangement is therefore analogous to investment in fixed assets or a change in plant layout: the development expenditure in setting up the new file arrangement is of the nature of capital expenditure which, once undertaken, becomes a fixed cost (depreciation). In the present case the investment yields up its benefits in the form of lower costs rather than higher receipts, and these benefits go mainly to users of the new service over its life. What policy should an efficient data bank management adopt in respect of introducing new file arrangements for users, and how should it charge for this extra service, the benefits of which accrue mainly to its customers?

The criterion for when it becomes desirable to restructure a file (while maintaining the existing file) is that the difference in costs with and without the additional file arrangement should be negative. That is, the restructured file should be introduced if the variable operating costs when both the original and restructured file are provided, plus the annual equivalent capital cost of developing the restructured file, are less than the variable cost of continuing to operate with the original file arrangement only. By "variable operating cost" is meant all costs relating to the files in question, incurred by the data bank and users, which vary more or less in proportion to level of usage, which is denoted below by the symbol x. If $c_o(x_o)$ and $c_o(x_1)$ denote the variable operating costs relating to the original file arrangement before and after introducing the new file arrangement, respectively, $c_1(x_2)$ the variable operating costs of the new file arrangement, K the capital cost of restructuring the file, whose estimated life is n years, the condition is:

$$\{c_o(x_1) + c_1(x_2) + Ka_{\overline{n|}}^{-1}\} < c_o(x_o) \quad ,$$

where $a_{\overline{n|}}^{-1} = 1/(v + v^2 + \ldots\ldots + v^n)$, $v = 1/(1+i)$, and i is the data bank's time value of money.

It will only pay the data bank to provide the new service if it is able to recoup through higher annual subscriptions from users of the new file arrangement an amount at least equal to the annual equivalent development charge, less any reduction in its own variable operating costs (denoted below by a superscript D) as a result of the new file arrangement. If P is the existing annual subscription charged to users who would avail themselves of the new file arrangement, then

$$\Delta^{+}P \geq Ka_{\overline{n}|}^{-1} - \{c_o^D(x_o) - [c_o^D(x_1) + c_1^D(x_2)]\} .$$

Strictly, the expression inside the braces on the right-hand side should include any increment to revenues (as well as any reduction in variable costs) of the data bank brought about by new subscribers attracted by the augmented service.

1.4 Conclusion

In the preceding sections, we have introduced analytical concepts and approaches and applied them to the fundamental file organizations, sequential and direct, and to an important and relatively complete working databank. The purpose has been to develop, illustrate and prove in practice tools for cost analysis of databanks. We believe we have established a powerful and flexible framework that can be applied to any databank structure, and to many with only small extensions of the results contained in this chapter. We make both pedagogical and practical claims for the analysis.

The methods are of pedagogical value because they permit a thorough overview of file organizations and insight into the various applicabilities of different file structures that otherwise come only after long experience with file and databank systems. Students with an elementary pragmatic introduction to file manipulation on peripheral devices can acquire, through learning the analytical methods of this chapter, a rapid insight into the scopes of various data organizations as well as an analytical approach to the construction of practical systems.

Analysis following the methods of this chapter is a cheap and accurate way to design new file systems, providing a good framework for formulating and assessing design requirements. File growth, usage distributions, request rates, core requirements, etc. can all be taken explicitly into account in the models offered here, and a design can be evaluated in terms of acquisition, storage, retrieval and maintenance costs.

We have not attempted in this chapter to make more than qualitative observations on the results of applying our methods to the two main examples. More precise comparisons and suggestions will be made for the F.R.I. databank as part of on-going work in this project and reported elsewhere. We have been content to show that the elements of the analysis are simple and precise and that they can be applied to practical databanks.

Acknowledgements

This work was made possible by a grant from the Department of Communications. The author is indebted to Professor H.R. Howson for coordinating and encouraging this work and for many helpful discussions. The cooperation of the Financial Research Institute has been indispensible for the latter part of the work, and the author is particularly indebted to Mr. William Calder for his time and effort in supplying the facts and figures concerning operation of the Daily Stock Exchange Databank.

Glossary of Symbols

SYMBOL	MEANING	DEFINED
b	number of blocks per unit of storage	eq. 1.2.5.2.(11)
B	size of block (bytes)	eq. 1.2.5.1.1(2)
β	number of blocks per volume	§ 1.2.1.2
c	size of record (bytes)	eq. 1.2.5.1.1(1)
c_o, c_1, c_t	error terms in $\int n(t')dt'$	eq 1.3.1.1(12)
C	total cumulative cost (user level)	§1.2.5.5
C_a	acquisition cost (user level)	§1.2.1.1
C_α	access cost (data level)	§1.2.1.2
C_γ	cost of spooling 1000 cards (data level)	§1.2.1.2
C_ι	cost per I/O request (data level)	§1.2.1.2
C_m	maintenance cost (user level)	§1.2.1.1
C_μ	mounting charge (data level)	§1.2.1.2
C_r	retrieval cost (user level)	§1.2.1.1
C_s	storage cost (user level)	§1.2.1.1
C_σ	storage cost (data level)	§1.2.1.2
γ_s	rate of growth of stocks in FRI	eq.1.3.1.1(1)
γ_D	rate of growth of days in FRI	eq.1.3.1.1(3)
d	day (FRI)	§1.3
δ	depth of block in sequential file	eq.1.2.5.3(16)
d_δ	probability δ is maximum depth	eq.1.2.5.3(20)
Δt	time increment-various uses	§1.2.5.1.2, eq.1.3.1.1(9)
$\Delta_0 t, \Delta_1 t, \Delta_1' t$	time increments in FRI file growth	eq.1.3.1.1(9)
g	growth rate of linearly growing file	§1.2.5
k	record key	§1.2.5.3
λ	record location	eq.1.2.5.3(14)
m	number of passes required by sort	eq.1.2.5.1.1(4)

M	core memory available (bytes)	eq.1.2.5.1.1(2)
μ	effective charge ratio (data level)	§1.2.1.2
$n, n(t), n_T$	number of blocks in file	§1.2.2
n_o	initial number of blocks in file	§1.2.2
n_1	block increment in linearly growing file	§1.2.5.1.2
N	number of records in file	eq.1.2.5.1.1(1)
$N_s(t)$	number of stocks in FRI files	eq.1.3.1.1(1)
$N_d(t)$	number of days in FRI files	eq.1.3.1.1(3)
ν	number of records per block	§1.3.1
P	P-way merge in sort	eq.1.2.5.1.1(1)
q	growth of query rate (linear) (batches/day^2)	§1.2.5.5
Q	unit of storage (bytes) (usually 1 track)	eq.1.2.5.2(11)
r	number of single requests per batch	§1.2.3.1
r_a	addition rate (batches/day)	§1.2.5.3.2
r_c	change rate "	"
r_d	deletion rate "	"
r_q	query rate "	"
ρ	proportion of available storage used	eq.1.2.5.2(11)
s	stock (FRI)	§1.3
s_r	probability batch size is r	§1.2.5.3.2
S	number of initial runs in sort	eq.1.2.5.1.1(1)
σ	mean size of batches	§1.2.5.5
t	age of file (days)	§1.2.2
t_L	lifetime of databank (where different from t)	§1.2.5.6
T	age of file, discretized	§1.2.5.1.2
τ	time required to transfer 1 byte	eq.1.2.5.1.1(1)
Θ	usage distribution parameter	§1.3.4
u	usage distribution density	§1.2.5.3
U	usage distribution	eq.1.2.5.3.1(17)

v,w,z	coefficients in size polynomial in t	§1.3.1.1
$\overline{X}_r$	average number of accesses: batch of r requests	§1.3.4.

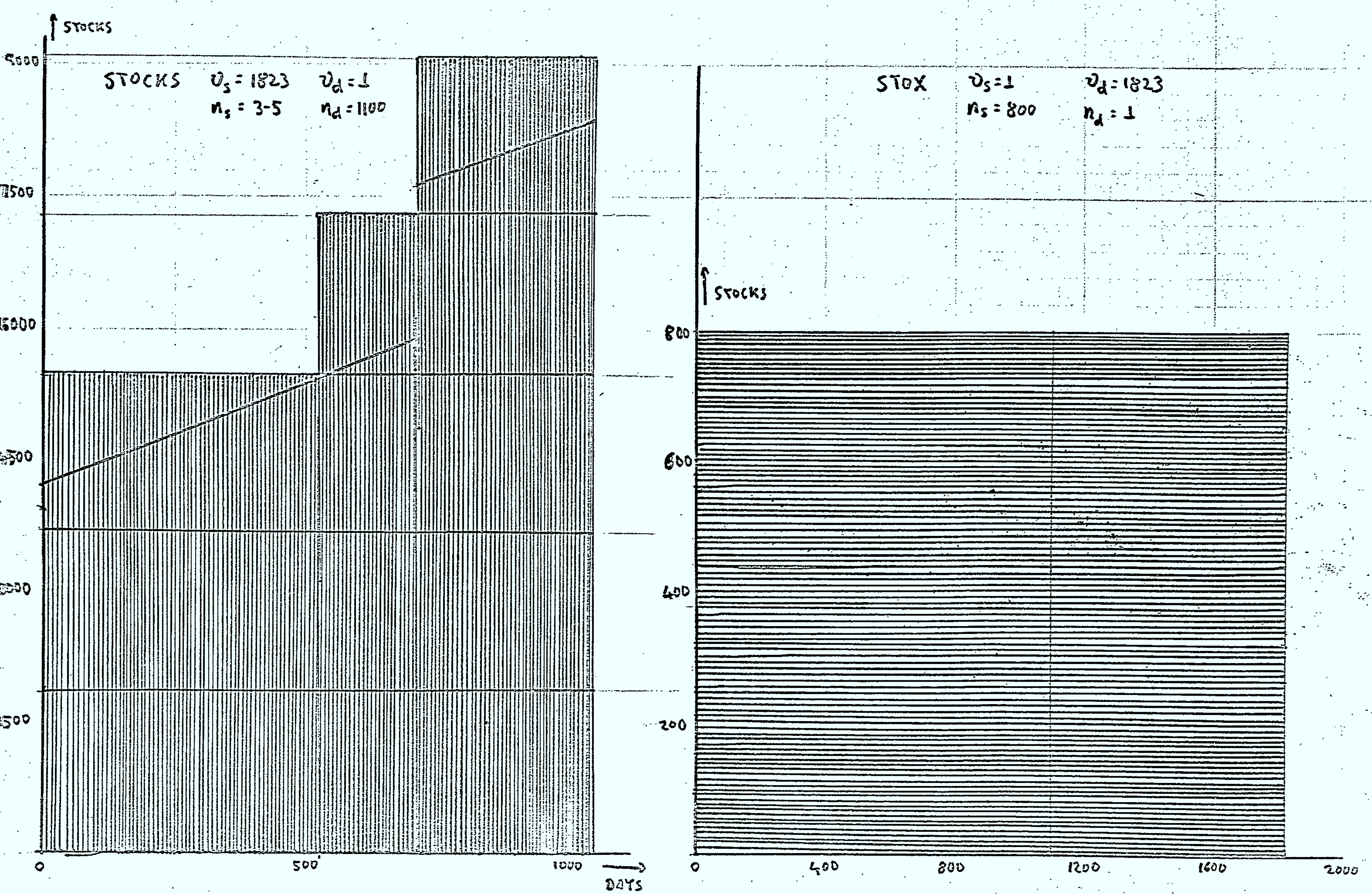


Fig. 21 Track allocation for STOCKS and STOX

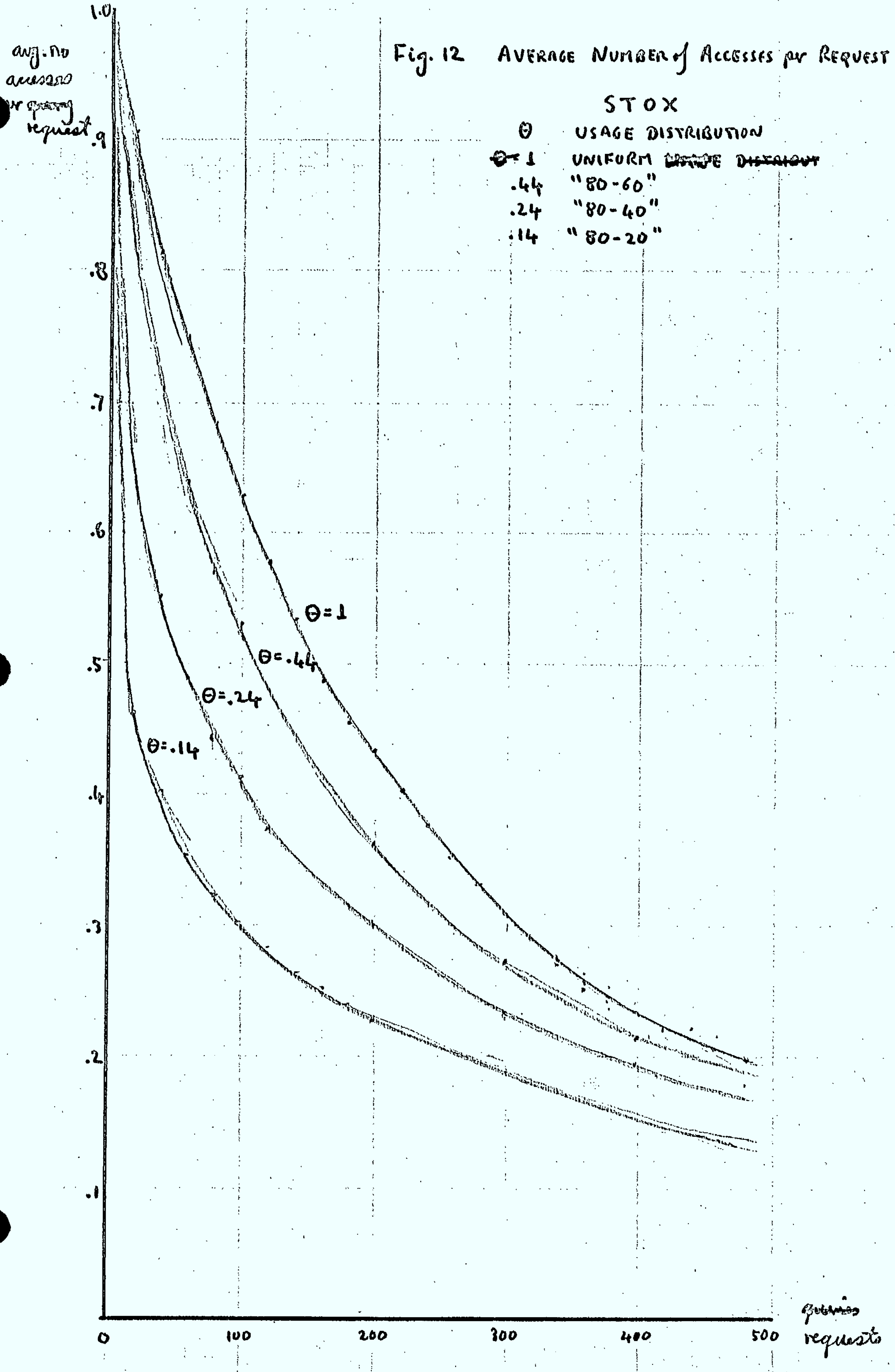
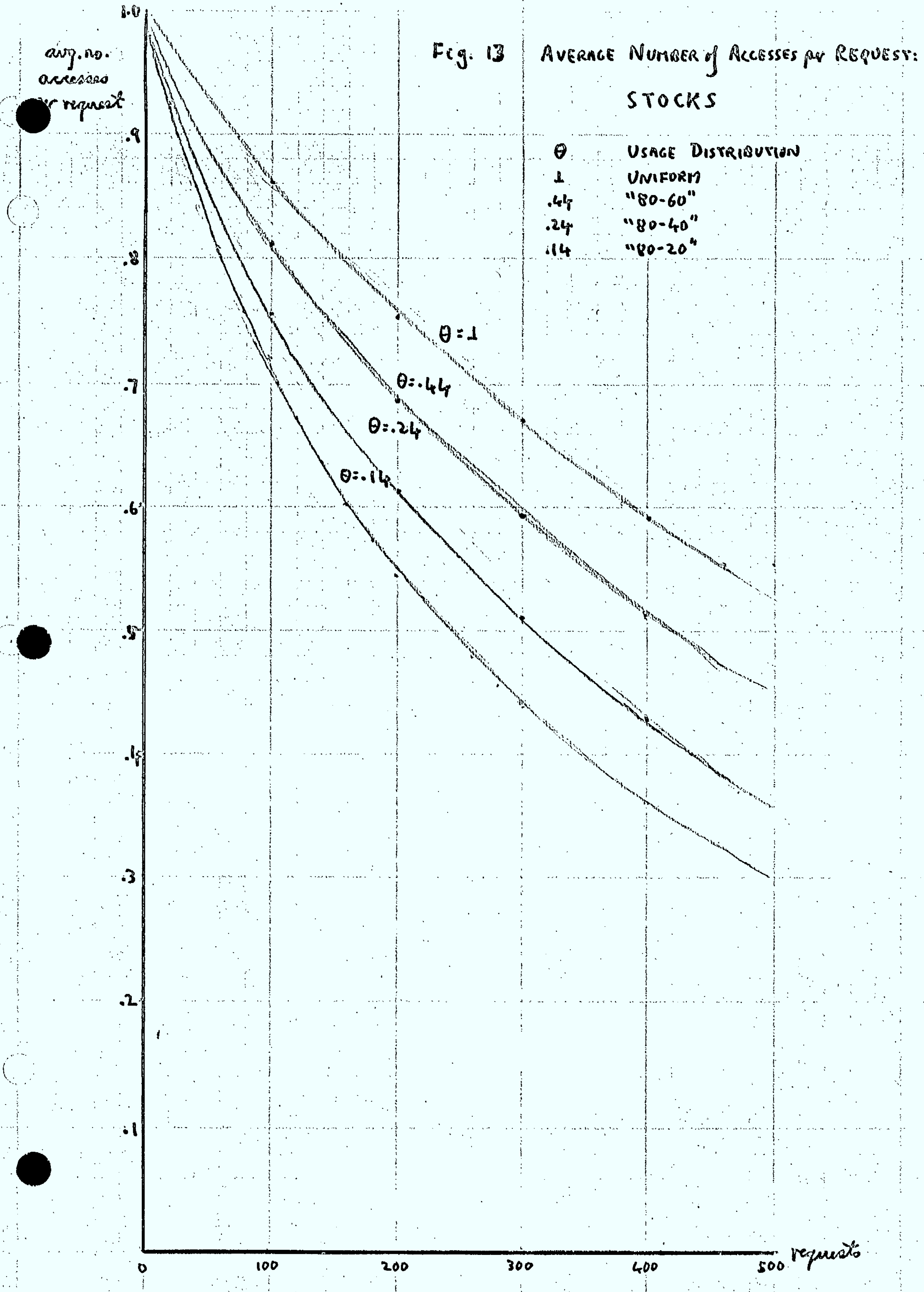


Fig. 12 AVERAGE NUMBER of ACCESSES per REQUEST:



avg. no. accesses per request
Fig. 13 AVERAGE NUMBER of ACCESSES per REQUEST: STOCKS
θ USAGE DISTRIBUTION
1 UNIFORM
.44 "80-60"
.24 "80-40"
.14 "80-20"
θ=1
θ=.44
θ=.24
θ=.14
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
100
200
300
400
500
requests

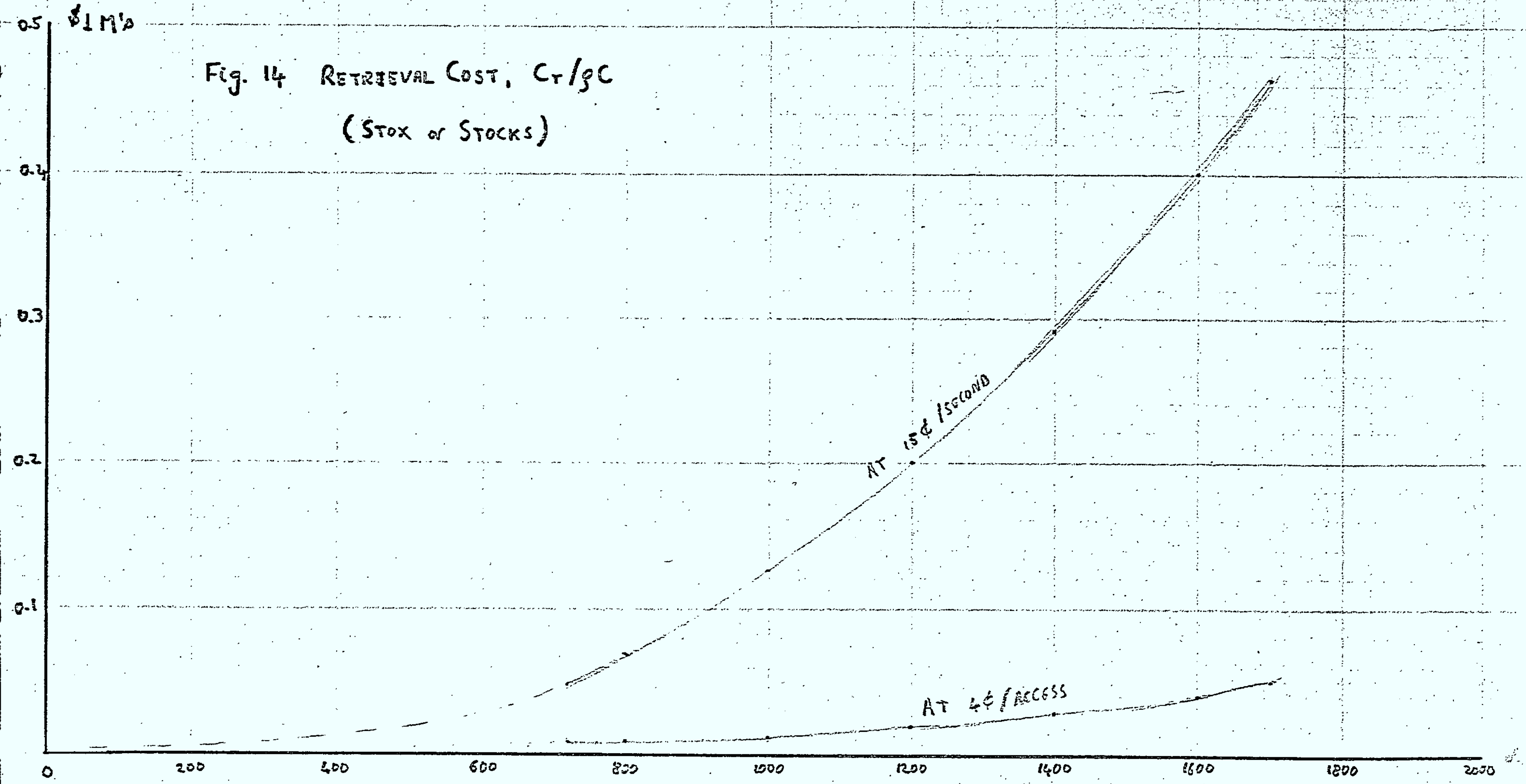


Fig. 14 RETRIEVAL COST, Cr/gC
(STOX or STOCKS)