Discussion Paper No. 131

OPTIMAL INFORMATION STORAGE SYSTEMS

by

V. Balachandran and S. D. Deshmukh

January 1975

Department of Managerial Economics and Decision Sciences
Graduate School of Management
Northwestern University
Evanston, Illinois 60201

ABSTRACT

The problem is to dynamically store different information items in different storage devices in each period so as to minimize the total expected discounted cost over a planning horizon. Each device has a fixed total capacity, each item has a given storage space requirement, while the number of requests for each item per period is changing stochastically through time. Given an allocation, the total cost per period consists of the storage cost (depending on the storage requirements), the access cost (depending on the number of requests) and the transfer cost (depending upon the change of allocation from the previous period). A dynamic programming model is presented to yield optimal strategies. The special case of independent identically distributed demands is completely solved, using a generalized transportation algorithm while a heuristic procedure is indicated for the general problem using parametric analysis.

OPTIMAL INFORMATION STORAGE SYSTEMS

1. Introduction

Complexities of managing modern industrial as well as service oriented operations has necessitated the utilization of sophisticated information storage and retrieval systems. Considering the high costs of operating such systems, the information systems manager is concerned with the efficient utilization of the storage devices. Thus in a computer-based data management system, a piece of information may be stored either in a primary device such as the extended core storage or in an auxiliary device such as a drum, a disc or a tape. The allocation of different pieces of information to different storage devices should be based on the frequency of demand for the information and the associated access costs, the storage requirements, capacities and costs. In addition, the demand patterns for such information are probabilistic in practice, so that periodic review and storage reallocations may be necessary through time, resulting in additional costs of information transfer among the storage devices. Thus the information systems manager is faced with the problem of dynamically allocating information items to the storage devices, so as to minimize the total expected cost over a given planning horizon.

Some related problems in data base management and hierarchical information storage and retrieval have been investigated in the literature. Severence [11] presents a model which can parametrically describe various methods of file organization. Salasin [9] has considered expected time required to access data stored in hierarchical memories, using different file organizations, given the probability distribution of requests. Ramamoorthy and Chandy [7] consider the problem of selecting optimal memory hierarchy so as to minimize the average access time given a budget constraint on the total cost. Kennedy [5] and Sepalla [10] have considered some integer programming models for optimal file partitioning of information records located on random access devices, so as to minimize the data transmitted to the main memory. Babad, Balachandran, Stohr [1] have considered the combined problem of optimal file partitioning and storage among memory hierarchies so as to minimize total access and storage costs.

Most of these models depend upon integer programming formulation of the optimal file partitioning and storage decisions. Since our problem involves a large number of variables, the current integer programming algorithms are generally not capable of handling them. Here we will provide a more efficient algorithm. Furthermore, all of the above models are static and can not be extended to investigate the problem of dynamic storage allocation described earlier in this section. Such a problem arises frequently in practice. Computarized inventory management (in manufacturing, retail distribution centers, spare parts warehouses, etc.) requires dynamically storing the information about different items in different storage devices depending upon the changes in the demand distribution through time. Similar dynamic decisions have to be made in the context of information systems used in the libraries, banks and hospitals. The problem is particularly relevant to those managers who rent the computer storage and time from commercial computer centers.

In the next section we present the dynamic storage allocation model for given file partitions, so as to minimize the total expected discounted cost over a given planning horizon.

2. Model Formulation:

In each one of the applications described in the previous section we can identify various physical units about which information has to be stored. The entire information record can be partitioned into data items according to some file partitioning procedure. In this section we are concerned with the optimal storage of these data items. Suppose there are m data items indexed by $i \in I = \{1,2,\ldots,m\}$, each of which has to be stored in one of the n storage devices indexed by $j \in J = \{1,2,\ldots,n\}$. Each complete data item is required to be stored in one device, since information splitting among devices is not practical. Let e_i be the number of elementary storage units (e.g. bits), abbreviated as esus, required to store the i^{th} data item. Let b_j denote the total capacity in esus of the j^{th} storage device.

The problem is that of dynamic allocation of the data items to storage devices over a given planning horizon of duration $T \leq \infty$. For simplicity, the allocation decisions are assumed to be made at equally spaced discrete points in time denoted by $t = 0,1,2,\ldots,T$. Thus, for example, every Monday morning, based on the past week's number of requests for information contained in a data item, we may decide to move it from the drum to the tape or vice versa, depending upon the predicted demand for the present week. Let $x_{ij}(t)$ be the binary decision variable with value 1 if the data item i is stored in the storage device j in period t and 0 otherwise. Let $X(t)$ be the $m \times n$ matrix of allocation decisions for period t. Denote by $d_i(t)$ the random variable representing the number of requests for the i^{th} item in period t, thus taking values in $S = \{0,1,2,\ldots\}$.

Let $D(t)$ be the random vector of demands at time t having a given multivariate distribution. For simplicity, we will assume that $\{D(t):t=0,1,2,...\}$ is a Markov process with the state space S^m, given transition probabilities which are time invariant and a given initial distribution of $D(0)$.

The costs which are relevant for optimal decisions $\{X^*(t):t=0,1,2,...\}$ are the costs of storing, accessing and moving information among storage devices. Let CS_{ij} be the cost of storing the i^{th} item in the j^{th} device per period. This storage cost includes the physical cost of storing e_i esus (e.g. the rental cost) as well as the opportunity cost due to limited storage capacity of the j^{th} device. Let CA_{ij} be the cost of accessing the i^{th} item at the j^{th} device and shifting it to the core. This access cost includes a fixed and a variable component. The fixed cost is due to the time delay required for calculating the address, moving the arm to that address and other possible rotational delays. The variable component of CA_{ij} is the cost of bringing e_i esus from the j^{th} device to the core. Finally, let CT_{ikj} be the cost of transferring e_i esus of the i^{th} item in device k to device j ($CT_{ijj} = 0$ for all $j \in J$). This transfer cost consists of fixed and variable components similar to the access cost, the only difference being that the transfer takes place between the storage devices rather than from a device to the core. The costs will be assumed to be discounted by a factor $\beta \in [0,1)$ per period, i.e. β is the present worth of one dollar spent in the next period.

The problem of minimizing the total expected discounted cost over the infinite time horizon can be formulated in the stochastic dynamic programming framework (see Ross [8]) as follows. Denote by $V(X(t-1), D(t-1))$

the total expected discounted cost from period t onwards following the optimal policy, given that the storage allocation decision in the previous period (t-1) was X(t-1) and the vector of demands occurring in that period was D(t-1). Note that the decision X(t) has to be made at the beginning of period t, while the random demand D(t) takes place during that period. Let α denote a stationary policy for choosing the decisions X(t) dynamically, so that $\alpha(X(t-1), D(t-1))$ yields X(t) for all values of X(t-1) and D(t-1), $t \geq 1$. It is well known (Ross [8]) that in search of an optimal policy we may confine ourselves only to stationary policies and the optimal stationary policy specifies that X(t) which yields the minimum in the following recurrence relation.

$$(1) \quad V(X(t-1),D(t-1)) = \min_{X(t)\in F} \left\{ \sum_{j=1}^{n} \sum_{i=1}^{m} \left[CS_{ij}x_{ij}(t) + CA_{ij}E[D_i(t)\,|\,D(t-1)]\,x_{ij}(t) + \sum_{k=1}^{m} x_{ij}(t-1)CT_{ikj}\; x_{ij}(t) \right] + \beta\, E[V(X(t),D(t)\,|\,D(t-1)] \right\}$$

where

$$(2) \quad F = \left\{ x : \sum_{j=1}^{n} x_{ij} = 1,\ i \in I; \sum_{i=1}^{m} e_i x_{ij} \leq b_j,\ j \in J;\ x_{ij} = 0 \text{ or } 1,\ i \in I,\ j \in J \right\}$$

denotes the feasible set of decision variables. In the expression on the right hand side of (1) the first three terms give the storage cost, the expected access cost and the transfer cost in the t^{th} period, while the last term denotes the total optimal expected discounted cost from the period (t+1) onwards. Note that the summation in the transfer cost term yields at most one positive contribution, namely CT_{ikj} when $k \neq j$ and $x_{ik}(t-1) = x_{ij}(t) = 1$.

In principle, the solution of the functional equation (1) completely solves the multiperiod information storage problem. However, in practice, there are thousands of data items to be stored in probably few devices but resulting in a high dimensional state matrix X(t) and vector D(t), which is unsolvable with the current state of art in dynamic programming. Similarly, currently available integer programming codes may not be adequate to handle the "curse of dimensionality" existing even in the single period version of this problem. Further complication arises due to the stochastic nature of the requests for data items. In the remainder of the paper we provide optimal strategies under certain conditions and a heuristic procedure for the general problem.

3. Independent Identically Distributed Demands

In this section we assume that the multivariate distribution of D(t) is independent of t for all $t \geq 1$. Note that the random variable $D_i(t)$ and $D_j(t)$, $i \neq j$, need not be independent, so that we take into consideration the possibility that a request may require a simultaneous access to a block of interrelated data items. In this case of D(t) independent of D(t-1), the state variable in the functional equation (1) reduces to X(t-1), with the same definition of F as in (2), thus yielding

$$(3) \qquad V(X(t-1)) = \underset{X(t)\in F}{\mathrm{Min}} \left\{ \sum_{j=1}^{n} \sum_{i=1}^{m} \left[CS_{ij}x_{ij}(t) + CA_{ij}\, E(D_i(t))x_{ij}(t) + \sum_{k=1}^{n} x_{ik}(t-1)\, CT_{ikj}x_{ij}(t) \right] + \beta\, V(X(t)) \right\}$$

Let us consider the following auxiliary time independent problem:

(4) Minimize $\sum_{j=1}^{n} \sum_{i=1}^{m} C_{ij} x_{ij}$

$$\text{s.t.} \quad \sum_{i=1}^{m} e_i \, x_{ij} \le b_j \qquad j \in J$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i \in I$$

$$x_{ij} = 0 \text{ or } 1, \; i \in I, \; j \in J, \qquad \text{where}$$

(5) $C_{ij} = CS_{ij} + CA_{ij} \, E(D_i)$

The auxiliary problem (4) is a generalized transportation problem [4, 5] with the added 0-1 constraints. Let X^* be the optimal solution to (4) and let C^* be the corresponding optimal value. The relevance of considering this problem is exhibited in the following

Theorem 1: The optimal strategy for the dynamic problem (3) is $X(t) = X^*$ independent of t and $V(X^*) = C^*/(1-\beta)$.

Proof: Consider the version of problem (3) truncated at $T < \infty$. For this finite planning horizon problem the functional equation (3) remains the same with $V_t(X(t-1))$ and $V_{t+1}(X(t))$ replacing $V(X(t-1))$ and $V(X(t))$ respectively and $V_{T+1}(X(T)) \equiv 0$. Note that the original problem and the functional equation (3) are obtained by letting $T \to \infty$. We prove that if any period t starts with the storage configuration $X(t-1) = X^*$, then the optimal decision is not to change it, i.e. $\alpha^*(X^*) = X^*$. Then we prove that the best configuration with which to start any period t is $X(t-1) = X^*$. These two facts together yield the optimal policy as being the one which selects $X(0) = X^*$ and yields for any $t \ge 1$ $\alpha^*(X(t-1)) = X^*$.

It suffices to show these results by induction in the truncated version of the problem. Suppose $X(T-1) = X^*$, the r in the T^{th} period

$$V_T(X^*) = \underset{X\in F}{\text{Min}}\left\{\sum_{j=1}^{n}\sum_{i=1}^{m}\left[C_{ij}x_{ij} + \sum_{k=1}^{n} x^*_{ik}\ CT_{ikj}\ x_{ij}\right]\geq C^*\right.$$

since

$$\sum_{k=1}^{n} x^*_{ik}\ CT_{ikj}\ x_{ij} \geq 0$$

On the other hand, since X^* is a particular feasible solution in F, the right hand side of the above equality is less than or equal to C^*, since $\sum_{k=1}^{n} x^*_{ik}\ CT_{ikj}\ x^*_{ij} = 0$. Thus, $V_T(X^*) = C^*$ and $\alpha_T(X^*) = X^*$. Also note that for any $X(T-1)\in F$, $V_T(X(T-1)) \geq \underset{X\in F}{\text{Min}}\left\{\sum_{j=1}^{n}\sum_{i=1}^{m} C_{ij}x_{ij}\right\} + \underset{X\in F}{\text{Min}}\left\{\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{k=1}^{n} x_{ik}(T-1)\ CT_{ikj}x_{ij}\right\}$

$= C^* + 0$, i.e. $V_T(X(T-1)) \geq C^* = V_T(X^*)$.

Suppose $V_{t+1}(X(t)) \geq V_{t+1}(X^*) = C^* \cdot \frac{1-\beta^{T-t-1}}{1-\beta}$ and $\alpha^*_{t+1}(X^*) = X^*$.

$$V_t(X^*) = \underset{X\in F}{\text{Min}}\left\{\sum_{j=1}^{n}\sum_{i=1}^{m}\left[C_{ij}x_{ij} + \sum_{k=1}^{n} x^*_{ik}\ CT_{ikj}\ x_{ij}\right] + \beta\ V_{t+1}(X)\right\}$$

$$\geq \underset{X\in F}{\text{Min}}\left\{\sum_{j=1}^{n}\sum_{i=1}^{m} C_{ij}x_{ij}\right\} + \underset{X\in F}{\text{Min}}\left\{\sum_{k=1}^{n} x^*_{ik}\ CT_{ikj}x_{ij}\right\} + \beta\ \underset{X\in F}{\text{Min}}\ V_{t+1}(X)$$

$$= C^* + \beta \cdot V_{t+1}(X^*) = C^*\ \frac{1-\beta^{T-t}}{1-\beta}\ .$$

But

$$V_t(X^*) \leq \sum_{j=1}^{n}\sum_{i=1}^{m}\left[C_{ij}x^*_{ij} + \sum_{k=1}^{n} x^*_{ik}\ CT_{ikj}\ x^*_{ij}\right] + \beta\ V_{t+1}(X^*) = C^*\ \frac{1-\beta^{T-t}}{1-\beta}$$

Thus $V_t(X^*) = C^* \dfrac{1-\beta^{T-t}}{1-\beta}$ and $\alpha_t(X^*) = X^*$. Also

$$V_t(X(t-1)) = \underset{X\in F}{\text{Min}} \left\{ \sum_{j=1}^{n} \sum_{i=1}^{m} \left[C_{ij} x_{ij} + \sum_{k=1}^{n} x_{ik}(t-1)\ CT_{ikj}\ x_{ij} \right] + \beta\ V_{t+1}(X) \right\}$$

$$\geq \underset{X\in F}{\text{Min}} \left\{ \sum_{j=1}^{n} \sum_{i=1}^{m} C_{ij} x_{ij} \right\} + \beta \underset{X\in F}{\text{Min}} \{V_{t+1}(X)\} = C^* + \beta\ V_{t+1}(X^*) = V_t(X^*).$$

Finally, as $T \to \infty$ we get

$$V(X) \geq V(X^*) \quad \text{for all } X \in F$$

$$V(X^*) = C^* + \beta\ V(X^*), \text{ i.e. } V(X^*) = \frac{C^*}{1-\beta} \text{ and } \alpha(X^*) = X^*$$

Q.E.D.

Thus the optimal storage policy is to choose X^* in the first period and to stay there forever. Next, we deal with the procedure for solving the auxiliary problem (4) yielding X^* and C^*.

To obtain X^* and C^* we first convert problem (4) into the standard format of the Generalized Transportation Problem (GTP). Let us introduce a fictitious device (n+1) with an abundant capacity $b_{n+1} = M_1$ (a large positive constant) and also a fictitious data item (m+1) to be stored among the devices to fill up any unused capacity available. Let $I' = I \cup \{m+1\}$ and $J' = J \cup \{n+1\}$. Set $C_{i,n+1} = M_2$ (a high penalty cost), $i \in I$, so that no item will be assigned to the fictitious device in the optimal solution, let $C_{m+1,n+1} = 0$ and $e_{m+1} = 1$. With these additional definitions the problem (4) is equivalent to the following zero-one Generalized Transportation problem.

(5) $$\text{Min} \sum_{i=1}^{m+1} \sum_{j=1}^{n+1} C_{ij}\ x_{ij}\ .$$

$$(6) \qquad \text{s.t.} \quad \sum_{i=1}^{m+1} e_{ij}\, x_{ij} = b_j \qquad j \in J'$$

$$(7) \qquad \sum_{j=1}^{n+1} x_{ij} = 1 \qquad i \in I$$

$$(8) \qquad x_{ij} \geq 0 \qquad i \in I',\ j \in J'$$

$$(9) \qquad x_{ij} = 0 \text{ or } 1 \qquad i \in I,\ j \in J'$$

(It is to be noted that $x_{m+1,j}$ can be fractional and $\sum_{j=1}^{n+1} x_{m+1,j}$ need not be equal to 1.)

Let us call the problem (5)-(8) as a GTP and (5)-(9) as IGTP (Integer GTP). The GTP can be solved by the generalized stepping stone method [4, 5] or by the Four index algorithm [3]. In the GTP format the (m+1) rows correspond to the data items and the (n+1) columns represent the storage devices, thus yielding a matrix of cells (i,j), $i \in I'$, $j \in J'$. A basis B of the GTP is a collection of (m+n+1) linearly independent cells. A solution $X = \{x_{ij}\}$ is basic if $x_{ij} = 0$ for $(i,j) \notin B$. A basic solution X is feasible if it satisfies constraints (6)-(8). Due to the creation of the fictitious device (n+1) there always exists a basic feasible solution and hence a basic optimal solution. The original problem (4), even without the zero-one constraints, has no feasible solution if and only if in the optimal solution of the GTP at least one cell (i,n+1) is in the basis at a positive level, i.e. $x_{i,n+1} > 0$, [3].

In the special case where e_i is the same for all i, the GTP reduces to an ordinary transportation problem and, due to total unimodularity, the zero-one constraints (9) are automatically satisfied. Then the IGTP and GTP are identical. Hereafter we will assume that e_i's are different.

Since the number of basic cells is (m+n+1) and since there are (m+1) rows, each row having at least one cell in the basis, the total number of rows that can have more than one cell in the basis is at most (m+n+1) - (m+1) = n. For example, if the number of data items m = 1000 and the number of storage devices are n = 3 (drum, disc and tape), then at most 3 items will be stored in more than one device each, so that the zero-one constraint (9) will be violated for at most 3 variables $\{x_{ij}\}$. Therefore, a heuristic procedure for constructing a good solution to the IGTP from the optimal solution to the GTP would be to increase (if necessary) the capacity of a storage device (in practice, this is easy to do in case of tapes) and assign those data items (at most n) which were originally split among devices to the device with increased capacity. In the remainder of this section we provide a branch and bound algorithm for obtaining an optimal solution to the IGTP

Algorithm for IGTP

Step 1: Let P_1 denote the GTP given by (5)-(8). Obtain an optimal solution to P_1 according to the procedures in [3, 4, 5]. If the solution satisfies constraint (9), then we are done. Otherwise go to step (2).

Step 2: (Initialization) Let $SI_1 = \emptyset$ denote the set of cells (i,j) with $x_{ij} = 1$ in the optimal solution of IGTP. Let $SZ_1 = \emptyset$ denote the set of cells (i,j) with $x_{ij} = 0$ in the optimal IGTP. Let X_1 be the optimal solution to P_1 with optimal basis B_1 and cost Z_1. Let $\mathcal{P} = \{1\}$ denote the index set of problems under consideration and let $\ell = 1$ denote the total number of problems generated so far.

Step 3: Choose that problem P_k for which Z_k is the smallest for $k \in \theta$. If X_k satisfies zero-one constraint (9), go to step 9. Else go to step 4. (This enables one to branch along the current minimum cost node).

Step 4: Find the set of rows $I^* \subset I$ such that the basis B_k has more than one basic cell in each row $i \in I^*$ (I^* contains at most n rows). For each $i \in I^*$ find two basic cells (i,j) and (i,k) with the smallest and the second smallest unit costs. Let $\Delta_i = (C_{ik} - C_{ij})x_{ij}$ and choose $r \in I^*$ with $\Delta_r = \underset{i \in I^*}{\text{Max}}\ \Delta_i$. Select the basic cell (r,s) with $C_{rs} = \underset{(r,j) \in B_k}{\text{Min}}\ C_{rj}$ for branching. (This branch selection rule is similar to the entry criterion heuristic used in the Simplex method.)

Step 5: Let $P_{\ell+1}$ be the problem obtained from P_k by fixing $x_{rs} = 1$ for the IGTP. Set $SI_{\ell+1} = SI_k \cup \{(r,s)\}$ and $SZ_{\ell+1} = SZ_k$. $P_{\ell+1}$ is obtained from P_k by eliminating the row r and setting $b_s = b_s - e_r$. If this $b_s < 0$, eliminate this branch $\ell+1$ and any further branching from $\ell+1$. (This step forces the most promising cell to be in the basis of the IGTP, thus reducing the size of the GTP).

Step 6: Let $P_{\ell+2}$ be the problem obtained from P_k by fixing $x_{rs} = 0$ for the IGTP. Set $SZ_{\ell+2} = SZ_k \cup \{(r,s)\}$ and $SI_{\ell+2} = SI_k$. $P_{\ell+2}$ is obtained from P_k by setting $C_{rs} = M_1$ (a high penalty cost). (This forces x_{rs} to be = 0).

Step 7: Solve the GTPs $P_{\ell+1}$ and $P_{\ell+2}$ obtained in steps 5 and 6, yielding the optimal solutions $X_{\ell+1}$ and $X_{\ell+2}$ with corresponding bases $B_{\ell+1}$ and $B_{\ell+2}$. Let $Z_{\ell+1}$ = (Optimal cost of $P_{\ell+1}$) + $\sum_{(i,j) \in SI_{\ell+1}} C_{ij}$ and

$$Z_{\ell+2} = (\text{Optimal cost of } P_{\ell+2}) - \sum_{(i,j)\in SI_{\ell+2}} C_{ij}.$$ (The solution of $P_{\ell+1}$ and $P_{\ell+2}$ can be generated from that of P_k parametrically, instead of resolving, using the operator theory developed elsewhere [3].

Step 8: If in step 5 $b_s > 0$ then set $\theta = \theta \cup \{(\ell+1, \ell+2)\} - \{k\}$ and $\ell = \ell+2$; else $\theta = \theta \cup \{\ell+2\} - \{k\}$ and $\ell = \ell+1$. Go to step 3.

Step 9: The optimal solution of IGTP is X_k and $x_{ij} = 1$ for $(i,j) \in SI_k$ with the total cost = (Optimal cost of P_k) + $\sum_{(i,j)\in SI_k} C_{ij}$. Stop.

The storage requirement of this algorithm is not high since it is not needed to store all P_k's $k \in \theta$. Only SI_k and SZ_k need be stored for all $k \in \theta$. Any P_k can be constructed from the original GTP by setting $C_{ij} = M, (i,j) \in SZ_k$, eliminating row i and reducing b_j by e_i for $(i,j) \in SI_k$.

The computational time required for solving IGTP with m = 1000 and n = 4 using this algorithm is about 3 seconds of CPU time on CDC 6400.

4. The General Problem

Suppose the demands $D_i(t)$ are neither independent nor identically distributed. Then the dynamic multiperiod problem posed in section 2 becomes that of solving the functional equation (1) for the optimal policy yielding X(t) which maximizes the right-hand side of (1). The difficulties involved have already been outlined in section 2. Since the problem in its entirety can not be solved, a heuristic procedure is to solve a sequence of single period problems. In period t, given X(t-1) and D(t-1), the problem is

$$\text{(10)} \qquad \text{Min} \sum_{j=1}^{n+1} \sum_{i=1}^{m+1} \left[CS_{ij} + \overline{CA}_{ij}(t) + \overline{CT}_{ij}(t) \right] x_{ij}(t)$$

$$\text{s.t.} \quad X(t) \in F$$

where the constants $\overline{CA}_{ij}(t) = CA_{ij}\ E[D_i(t)|D(t-1)]$ and

$$\text{(11)} \qquad \overline{CT}_{ij}(t) = \sum_{k=1}^{m} x_{ik}(t-1)\ CT_{ikj}$$

are computed with known values of D(t-1) and X(t-1). Solving (10) by the algorithm indicated in section 3 yields X(t), then D(t) is observed in the t^{th} period and the problem for the period (t+1) is the same as (10) with new coefficients $\overline{CA}_{ij}(t+1)$ and $\overline{CT}_{ij}(t+1)$ computed according to (11). The problem for period (t+1) can then be solved parametrically from the optimal solution X(t) for period t. Efficient procedures for obtaining the optimal X(t+1) from the optimal X(t) when the cost coefficients change using the "area cost operator theory" are available in Balachandran and Thompson [3].

Further research is required in order to solve the complete problem (1) optimally, rather than heuristically, involving the dynamics, the high dimensionality and the integrality requirements. However, with the current state of the art in the areas of dynamic and integer programming, the task appears formidable.

References

[1] Babad, J., Balachandran, V., Stohr, E.A., "Management of Operating System Subprograms", Report 7438. Center for Mathematical Studies in Business and Economics, University of Chicago, Chicago, Ill., October 1974.

[2] Balachandran, V., "An Integer Generalized Transportation Model for Optimal Job Assignment in Computer Networks", Discussion Paper No. 58. Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, Ill., October 1973.

[3] Balachandran, V. and Thompson, G.L., "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - I: Basic Theory and II: Rim, Cost and Bound Operators", Naval Research Logistics Quarterly, Vol. 22, No. 1, March 1975.

[4] Balas, E. and Ivanescu, L.P., "On the Generalized Transportation Problem", Management Science, 11, pp. 188-202 (1964).

[5] Eiseman, K., "The Generalized Stepping Stone Method for the Machine Loading Model", Management Science, 11 pp. 154-176 (1964).

[6] Kennedy, S.R., Mathematical Models of Computer File Organization, Ph.D. dissertation, Department of Operations Research, Cornell University, June 1973.

[7] Ramamoorthy, C.V. and Chandy, K.M., "Optimization of Memory Hierarchies in Multiprogrammed Systems", Journal of the ACM, 17, 3, pp. 426-445, July 1970.

[8] Ross, S.M., Applied Probability Models with Optimization Applications, Holden-Day 1970.

[9] Salasin, J., "Hierarchical Storage in Information Retrieval", Communication of ACM, 16, 5, pp. 291-295, May 1973.

[10] Seppalla, Y., "Definition of Extraction Files and Their Optimization by Zero-One Programming", BIT, Vol. 7, No. 3, pp. 206-215.

[11] Severance, D.G., Some Generalized Modeling Structures for Use in Design of File Organizations, Ph.D. dissertation, University of Michigan, 1972.