NATIONAL BUREAU OF STANDARDS REPORT

10 146

OPTIMAL DESIGN OF SORTING NETWORKS

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics [2]—Radio Engineering [2]—Time and Frequency [2]—Astrophysics [2]—Cryogenics.[2]

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions:

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:

Engineering Standards—Weights and Measures — Invention and Innovation — Vehicle Systems Research—Product Evaluation—Building Research—Instrument Shops—Measurement Engineering—Electronic Technology—Technical Analysis.

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center consists of the following divisions:

Reactor Radiation—Linac Radiation—Nuclear Radiation—Applied Radiation.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards—Computer Information — Computer Services — Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data—Clearinghouse for Federal Scientific and Technical Information [3]—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.
[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT
4314444

January 1970

NBS REPORT
10 146

OPTIMAL DESIGN OF SORTING NETWORKS

by

W. A. Horn

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

ABSTRACT

The network design problem considered here involves mail which is sorted progressively to a number of final destinations by traveling through a network of sorting machines, each with one input channel and a fixed (for all machines) number of output channels. Cost for sorting is taken to be the sum of individual costs for sorting to each destination, which in turn is the product of mail volume to that destination times a "disutility" per unit of mail for the particular path in the network to that destination's "sink." A simple algorithm and a dynamic program provide complete specification of a network which minimizes cost for certain types of disutility functions.

1. PROBLEM FORMULATION[+]

This paper deals with a class of problems encountered in connection with a study[(1)] of mechanical mail sorting methods. We begin by describing these problems in quite general form, and then specialize to the particular cases for which solution methods will be developed below.

The problems deal with the design of a network of devices to sort material into a given number d of classes, e.g. to perform a "sort by destination" on pieces of mail each addressed to one of d known cities or regions. Problem data include nonnegative numbers $\{v_i\}_1^d$, where

v_i = mean volume of class i material to be sorted per unit time.

The devices available for use in the network constitute a set $\{M_j\}_1^n$ of single-input machines[(2)], each of which separates material into a given number m of categories. A sorting network N with parameters (n, m, d) is defined to be a network of nodes and directed arcs with

(a) a single origin node, with no incoming arc and one outgoing arc,

(b) n sorting nodes, each with one incoming arc and m outgoing arcs, and each containing one of the machines M_j,

---

[+]I am grateful to colleagues A.J. Goldman, L.S. Joel, and J. Levy for helpful discussions.

(1) Computer Symbolics, Inc., Task I Report (6/17/69) on The Operations Research Analysis and Design of Maximally Advantageous Sorting Configurations, under Post Office Department Contract No. RER 28-69.

(2) Here "machine" is used in the abstract or functional sense, without reference to a specific technology; similarly, the "material" to be sorted might consist (say) of information rather than physical objects.

(c) d destination nodes, each with one incoming arc and no outgoing arcs, and a one-to-one association of these nodes with the d classes of material to be separated, and

(d) the topological restriction[3] that there be exactly one directed path in N from the origin node to each destination node, and hence to each sorting node as well. (See Figure 1 for an example of a sorting network.)

In such a sorting network N, let P_i denote the unique path from the origin node to the destination node associated with the i-th class of material. Material in this class is to enter the network at the origin node and (if no errors occur) flow along P_i to the associated destination, which should thus be reached only by class i material. The machine at a given sorting node should receive material only of those classes i such that P_i contains the sorting node; the machine in effect separates this family of classes into m subfamilies for routing along its m outgoing arcs.

Before proceeding further, we recall (op. cit. in footnote 1) that the three parameters (n, m, d) are not independent. To establish this, we count in two ways the arcs of a sorting network: as "incoming arcs," so that there is one arc per sorting node and one per destination node, and as "outgoing arcs," so that there is one arc for the origin node and m for each sorting node. Equating the two counts, we obtain the relation

$$n + d = 1 + nm,$$

or equivalently

$$d = 1 + n(m - 1). \tag{1.1}$$

---

[3] In technical language, the restriction is that N be a "rooted tree."

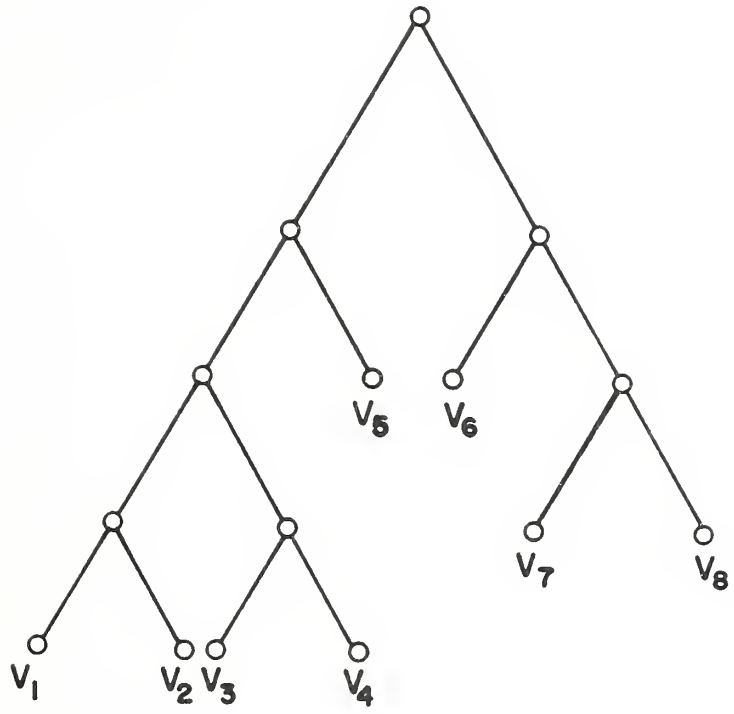

Figure 1. Sorting network with $m=2, n=7, d=8$

The design of a sorting network, for given values[4] (n, m) can be conceptually divided into three parts:

(a) specification of topology, subject to (d) above,

(b) associating the d destination nodes one-to-one with the d classes of material, and

(c) associating the n sorting nodes one-to-one with the n machines $\{M_j\}_1^n$, i.e. "assigning the machines to the sorting nodes."

For an optimal design problem, we must specify an "objective function" to be maximized or minimized. This will first be done in a quite general way to indicate a class of problems which may be of interest for future work. We then specialize to the type of problems for which the methods of this paper give a complete solution.

For any sequence (j(1), j(2), - - - j(L)) of distinct integers from the set {1, 2, - - -, n}, let $g_i[j(1), - - - j(L)]$ = "score" per unit flow of class i material through the sequence of machines $M_{j(1)}, - - -, M_{j(L)}$. Furthermore, for a sorting network N, let $S_i(N)$ = sequence of machines (specified by their indices) along the path P_i. Then the function to be extremized by a proper choice of N is the mean total score per unit time,

$$f(N) = \sum_1^d v_i \, g_i \, [S_i(N)]. \qquad (1.2)$$

In considering how (1.2) might be specialized to tractable forms of practical interest, it is useful to note three aspects of the generality of the quantities $g_i[S_i(N)]$ appearing in (1.2). The first is the subscript

[4] Given m and d, the n-value determined from (1.1) might be non-integral; in this case we can think of d as increased by the adjunction of enough "dummy" classes with no members to raise n to the next higher integer value.

on g_i, which implies that different classes of material should receive different scores for passage through a given sequence of machines. The scores we have in mind refer to the time or cost of such a passage, or the likelihood of damage or misrouting or of rejection as "unprocessable"; these are "disutilities" rather than "utilities," so that we will be concerned with minimizing f(N) rather than maximizing it. In this context, the subscript-dependence of g_i might be plausible if we were classifying objects by size or messages by length or pieces of mail by the degree of machine-readability of their addresses. But our motivating interest is the classification of mail by destination, which appears to be at most coincidentally correlated with classification by score in the sense indicated above. Thus we replace (1.2) by

$$f(N) = \sum_{1}^{d} v_i \, g\,[S_i(N)]. \qquad (1.3)$$

Second, at present the score $g[S_i(N)]$ in (1.3) depends not only on what set of machines is encountered in movement along path P_i, but also the order in which they are encountered. Such generality, while perhaps needed for some applications, does not seem especially relevant for our present purposes and will be dropped.

Third, no mention has been made of what properties of the machines influence the scores. It will be assumed that the relevant properties of M_j can be represented by a single positive number

a_j = indicator of disutility of passage through M_j.

As a consequence of these three steps, the scoring functions $g_i[j(1), j(2), \ldots j(L)]$ have been specialized to a function $g\,[a_{j(1)}, a_{j(2)}, \ldots, a_{j(L)}]$, which is unchanged under permutations of its positive-number arguments.

Before specializing further, three examples will be given to help indicate the minimum degree of generality we want our final mathematical model to possess. Suppose first that a_j represents the time required for an object to pass through M_j and on to the next node, and that our objective is to minimize total time spent in the network by all objects. Then

$$g[a_{j(1)}, a_{j(2)}, \cdots a_{j(L)}] = a_{j(1)} + a_{j(2)} + \cdots + a_{j(L)},$$

or more compactly

$$g[S_i(N)] = \Sigma \{a_j : M_j \varepsilon S_i(N)\}, \tag{1.4}$$

is the appropriate scoring function[5]. Next, assume a_j is the probability that M_j routes an object correctly (or, passes an object undamaged). If the objective is to maximize the expected number of objects per unit time which arrive at the correct destination node (or, which survive passage through the network undamaged), then the appropriate scoring function is

$$g[a_{j(1)}, a_{j(2)}, \cdots, a_{j(L)}] = -a_{j(1)}\, a_{j(2)} \cdots a_{j(L)},$$

or more compactly

$$g[S_i(N)] = -\Pi \{a_j : M_j \varepsilon S_i(N)\}, \tag{1.5}$$

the minus sign arising because our convention calls for a minimization rather than a maximization. Third, the score may depend only on how many machine-handlings an object receives in the network. If we define the length of a path to be the number of sorting nodes it contains, and employ the notation

$$L_i(N) = \text{length of } P_i, \tag{1.6}$$

---

[5] The special subcase with all $a_j=1$ is the one solved in the cited reference, footnote (1).

then for an appropriate strictly monotone function[5a] h the scoring function would be given by

$$g[S_i(N)] = h[L_i(N)]. \tag{1.7}$$

In this case the machines M_j are in effect assumed identical.[6]

To encompass these three cases while retaining the stipulations given previously on g, and to give a basis for developing a solution algorithm, we consider a continuous mathematical operation "*" which converts an ordered pair (x,y) of numbers from some interval I of real numbers into a number $x*y$ of I. (Here the interval I can be finite or infinite, open or half-open or closed.) The operation is assumed to satisfy the associative law

$$(x*y)*z = x*(y*z)$$

(in algebra, the "semi-group" property), and to have the further property that

$$x*y = x*z \text{ or } y*x = z*x \text{ implies } y = z.$$

We then define the scoring function as

$$g[a_{j(1)}, a_{j(2)}, \ldots a_{j(L)}] = h\,[a_{j(1)}*a_{j(2)}* \ldots *a_{j(L)}], \tag{1.8}$$

where h is continuous and strictly monotone. (With "*" as addition and $h(x) = x$ we get special case (1.4) or, if all $a_j=1$, special case (1.7). With "*" as multiplication and $h(x) = -x$, we get (1.5).)

Now we note that the above definition of "*" implies[7] that there exists a continuous, strictly monotonically increasing function ϕ defined on a

---

(5a) Function h is called strictly monotone if either $h(x)<h(y)$ whenever $x<y$, or $h(x)>h(y)$ whenever $x<y$.

(6) With $h(x)\equiv x$, this again reduces to the case treated in the cited reference.

(7) See, for example, Aczel, J., Lectures on Functional Equations and their Application, Academic Press, 1966, pp. 253 ff.

subset of $(-\infty,\infty)$ and with range I, such that[7a]

$$x*y = \phi\,(\phi^{-1}(x) + \phi^{-1}(y)). \tag{1.9}$$

Thus the scoring function may also be written as[7b]

$$g[a_{j(1)}, a_{j(2)}, \ldots a_{j(L)}] = h\phi[\phi^{-1}(a_{j(1)}) + \phi^{-1}(a_{j(2)}) + \ldots + \phi^{-1}(a_{j(L)})]$$

$$= h\phi[\sum_{i=1}^{L} \phi^{-1}(a_{j(i)})]. \tag{1.10}$$

Since h and ϕ are strictly monotone, so is $\bar{h} = h\phi$. Thus by relabeling machine j with the quantity $\bar{a}_j = \phi^{-1}(a_j)$, we may assume that the function g has the form

$$g[a_{j(1)}, a_{j(2)}, \ldots, a_{j(L)}] = \bar{h}\,[\sum_{i=1}^{L} \bar{a}_{j(i)}], \tag{1.11}$$

where $\bar{h}$ is strictly monotone.

Although the problem has not been solved in this generality, a simple algorithm and a dynamic program give the solution for two special cases of interest, as detailed in sections 2 and 3 respectively. The first two examples given (cf. equations (1.4) and (1.5)) are solvable by the algorithm of Section 2, while the third (1.7) is solvable by the dynamic program. Section 4 discusses two specific examples of the type problem of Section 3 which are of seeming practical importance.

[7a] $\phi^{-1}(x)$ is the unique number u such that $\phi(u) = x$.

[7b] Here $h\phi$ is defined by $h\phi(u) = h\,[\phi(u)]$.

2. THE ALGORITHM

We consider in this section the class of functions h given previously, further restricted so that

$$h(x*a) - h(y*a) = \Theta(a)[h(x) - h(y)], \tag{2.1}$$

where Θ is a positive continuous function, monotone in the direction opposite to that of h. Applying the transformation described in the previous section, (2.1) becomes

$$h\phi(\phi^{-1}(x)+\phi^{-1}(a)) - h\phi(\phi^{-1}(y) +\phi^{-1}(a)) =\Theta\phi(\phi^{-1}(a)) [h\phi(\phi^{-1}(x))- h\phi(\phi^{-1}(y))],$$

or,

$$\bar{h}(\bar{x} + \bar{a}) - \bar{h}(\bar{y}+\bar{a}) = \bar{\Theta}(\bar{a})[\bar{h}(\bar{x}) - \bar{h}(\bar{y})], \tag{2.2}$$

where $\bar{\Theta}$ is still monotone in the direction opposite to that of $\bar{h}$. We use the functions $\bar{h}$ and $\bar{\Theta}$ and the new labels in the discussion that follows.

First, however, we prove that equation (2.2) implies that $\bar{h}$ and $\bar{\Theta}$ have the form

$$\bar{\Theta}(x) = e^{cx} \qquad -\infty<x<\infty, \tag{2.3}$$

and

$$\bar{h}(x) = \begin{cases} k_1 e^{cx}+k_2 & (c\neq 0) \quad (2.4) \\ k_1 x +k_2 & (c=0) \quad (2.5) \end{cases}$$

which implies that

$$\Theta(x) = e^{c\psi(x)} \tag{2.6}$$

and

$$h(x) = \begin{cases} k_1 e^{c\psi(x)}+k_2 & (c\neq 0) \quad (2.7) \\ k_1 \psi(x) +k_2 & (c=0) \quad (2.8) \end{cases}$$

where $\psi=\phi^{-1}$. These relationships are shown as follows: First, it is clear that $\bar{\Theta}(ma) = (\bar{\Theta}(a))^m$ for any positive integer m.

For,

$$\bar{\theta}(ma)[\bar{h}(x)-\bar{h}(y)] = \bar{h}(x+ma) - \bar{h}(y+ma) = \bar{\theta}((m-1)a)\bar{\theta}(a)[\bar{h}(x)-\bar{h}(y)]$$

$$= [\bar{\theta}(a)]^m[\bar{h}(x)-\bar{h}(y)]$$

if $\bar{\theta}((m-1)a) = (\bar{\theta}(a))^{m-1}$. Thus a simple inductive argument shows that $\bar{\theta}(ma) = [\bar{\theta}(a)]^m$, provided we take $\bar{h}(x) = \bar{h}(y)$ in the above equation. Therefore $\bar{\theta}(a/m) = [\bar{\theta}(a)]^{1/m}$, for each positive integer m, so that if $\bar{\theta}(1) = e^c$ then $\bar{\theta}(j/m) = e^{cj/m}$. Thus $\bar{\theta}(x) = e^{cx}$ holds for all positive rational numbers $x = j/m$; by continuity, it holds for all $x \geq 0$.

Similarly, we have

$$\bar{\theta}(-ma)[\bar{h}(x)-\bar{h}(y)] = \bar{h}(x-ma) - \bar{h}(y-ma)$$

$$= \bar{\theta}(-(m+1)a)\bar{\theta}(a)[\bar{h}(x)-\bar{h}(y)],$$

or

$$\bar{\theta}(-(m+1)a) = [\bar{\theta}(a)]^{-1}\,\bar{\theta}(-ma),$$

which leads to $\bar{\theta}(x) = e^{cx}$ for $x \leq 0$, as above.

To show that $\bar{h}$ is of the form (2.4) or (2.5), we note that if some $\bar{h}$ satisfies (2.2), then so does $K_1\bar{h} + K_2$ for any constants K_1 and K_2. Let $\bar{h}(x)$ be any function satisfying (2.2) for $\bar{\theta} = e^{cx}$, $c \neq 0$. Then, by the last remark, we may assume that $\bar{h}(0) = 0$ and $\bar{h}(1) \neq e^c - 1$. Now for any x let k be defined by

$$\bar{h}(x) = k(e^{cx} - 1).$$

By induction, it may be shown that

$$\bar{h}(mx) = k(e^{cmx} - 1)$$

for any positive integer m, since if this is true for m-1 we have, by (2.1),

$$\bar{h}(mx) - \bar{h}((m-1)x) = e^{c(m-1)x}[\bar{h}(x) - \bar{h}(0)],$$

or

$$\bar{h}(mx) = k(e^{c(m-1)x} - 1) + e^{c(m-1)x}[k(e^{cx}-1)] = k\,(e^{cmx}-1).$$

Letting $x = 1/m$, we get $k=1$ so that

$$\bar{h}(1/m) = e^{c/m} - 1,$$

and by the previous argument,

$$\bar{h}(p/m) = e^{cpm} - 1$$

for all positive integers p and m. Thus by the continuity of $\bar{h}$,

$$\bar{h}(x) = e^{cx} - 1,$$

for all $x \geq 0$. This result is easily extended to negative numbers by showing that

$$\bar{h}\,(-mx) = e^{-cmx} - 1$$

in a fashion similar to the above. Thus (2.4) is proved.

If $c = 0$, it may be shown that setting $\bar{h}(0) = 0$ and $\bar{h}(1) = 1$ results in $\bar{h}(x) = x$, by an argument analogous to the above. Thus (2.5) is also proved.

We now develop the basis for the algorithm. In what follows, the scoring function will be given by

$$g[a_{j(1)}, a_{j(2)}, \cdots a_{j(L)}] = \bar{h}\left(\sum_{i=1}^{L} \bar{a}_{j(i)}\right),$$

where $\bar{h}$ satisfies (2.2) and $\bar{a}_{j(i)} = \phi^{-1}(a_{j(i)})$. For convenience, let

the numbering of classes and machines be such that

$$v_1 \leq v_2 \leq \cdots \leq v_d \tag{2.9}$$

and

$$\bar{a}_1 \leq \bar{a}_2 \leq \cdots \leq \bar{a}_n \tag{2.10}$$

if $\bar{h}$ is increasing, while

$$\bar{a}_1 \geq \bar{a}_2 \geq \cdots \geq \bar{a}_n \tag{2.10'}$$

if $\bar{h}$ is decreasing.

We also introduce the following two definitions. In any sorting network N, a final sorting node is one whose outgoing arcs all terminate in destination nodes, and a chain is a path from the origin node to a final sorting node.

The following lemma is based solely on the numbering (2.9), (2.10) or (2.10'), and the form of the function f(N).

LEMMA 1. There is an optimal network for which some final sorting node is labeled n and the final destinations which it sorts to are labeled 1, 2, ..., m.

PROOF. Let N be any optimal network. Choose a destination node corresponding to some class i, such that $g[S_i(N)]$ is maximum. Let x be the final sorting node preceding this destination node. If x does not sort to classes 1, 2, ..., m, then there exist integers p and q, with $1 \leq q \leq m < p$, such that x sorts to class p but not to class q. Consider the new network N' formed by interchanging destination classes p and q in N. We have

$$
\begin{aligned}
f(N) - f(N') &= (v_p g[S_p(N)] + v_q g[S_q(N)]) \\
&\quad - (v_p g[S_q(N)] + v_q g[S_p(N)]) \\
&= (v_p - v_q)(g[S_p(N)] - g[S_q(N)]) \\
&= (v_p - v_q)(g[S_i(N)] - g[S_q(N)]) \\
&\geq 0,
\end{aligned}
$$

by (2.9) and the fact that $g[S_i(N)]$ is maximal. Thus interchanging classes p and q in N does not destroy optimality. By an inductive process, we may move all of the destinations 1, 2, . . ., m to final sorting node x without destroying optimality. Therefore, in the rest of the proof we assume that x sorts to nodes 1, 2, . . ., m.

Suppose that node x contains machine M_k where $k \neq n$. Let

$$Q = \{i : n \in S_i(N)\}.$$

Then Q contains at least m members. Let R be any subset of Q of cardinality m, and let distinct numbers $j(i) \in R$ be defined so that

$$v_i \leq v_{j(i)}, \qquad i = 1, 2, \ldots, m. \tag{2.11}$$

Clearly this is possible, by (2.9).

Consider the network N' formed by interchanging machines k and n in N. In the network N let

$$u_r = \Sigma\{\bar{a}_s : s \in S_r(N),\ s \neq n\}$$

for $r \in Q$, and let

$$u = \Sigma\{\bar{a}_s : s \in S_1(N),\ s \neq k\}.$$

Then we have

$$f(N') - f(N) = [\bar{h}(u + \bar{a}_n) - \bar{h}(u+\bar{a}_k)] \sum_{i=1}^{m} v_i$$

$$+ \sum_{r \in Q} v_r[\bar{h}(u_r+\bar{a}_k) - \bar{h}(u_r+\bar{a}_n)]. \qquad (2.12)$$

Now suppose that $\bar{h}$ is increasing and $\bar{\theta}$ decreasing in what follows. (A similar argument applies in the opposite case.) Since $\bar{a}_k \leq \bar{a}_n$, by (2.10), we have

$$\bar{h}(u_r+\bar{a}_k) - \bar{h}\ (u_r+\bar{a}_n) \leq 0,$$

for each $r \in Q$-R, so that

$$f(N') - f(N) \leq [\bar{h}(u+\bar{a}_n) - \bar{h}(u+\bar{a}_k)] \sum_{i=1}^{m} v_i$$

$$+ \sum_{r \in R} v_r[\bar{h}(u_r+\bar{a}_k) - \bar{h}(u_r+\bar{a}_n)]$$

$$= \sum_{i=1}^{m} \{v_i[\bar{h}(u+\bar{a}_n) - \bar{h}(u+\bar{a}_k)]$$

$$+ v_{j(i)}[\bar{h}(u_{j(i)}+\bar{a}_k) - \bar{h}\ (u_{j(i)} + \bar{a}_n)]\}. \qquad (2.13)$$

If $f(N') - f(N) \leq 0$, then the lemma has been proved since N' is also optimal. Thus it is sufficient to show, because of (2.13), that

$$v_i[\bar{h}(u+\bar{a}_n) - \bar{h}(u+\bar{a}_k)] + v_{j(i)}\ [\bar{h}(u_{j(i)} + \bar{a}_k) - \bar{h}(u_{j(i)} + \bar{a}_n)]$$

$$\leq 0, \quad i = 1, 2, \ldots, m.$$

Because $\bar{a}_n \geq \bar{a}_k$, $\bar{h}(u_{j(i)}+\bar{a}_k) \leq \bar{h}(u_{j(i)} + \bar{a}_n)$, as noted above.

Furthermore, $v_{j(i)} \geq v_i$, so that

$$v_i[\bar{h}(u+\bar{a}_n) - \bar{h}(u+\bar{a}_k)] + v_{j(i)}[\bar{h}(u_{j(i)}+\bar{a}_k) - \bar{h}(u_{j(i)} +\bar{a}_n)]$$

$$\leq v_i[\bar{h}(u+\bar{a}_n) - \bar{h}(u+\bar{a}_k)] + v_i[\bar{h}(u_{j(i)} +\bar{a}_k) - \bar{h}(u_{j(i)} + \bar{a}_n)]$$

$$= v_i[\bar{h}(u+\bar{a}_n) - \bar{h}(u+\bar{a}_k) + \bar{h}(u_{j(i)}+\bar{a}_k) - \bar{h}(u_{j(i)}+\bar{a}_n)]$$

$$= v_i\{\bar{\theta}(u)[\bar{h}(\bar{a}_n) - \bar{h}(\bar{a}_k)] - \bar{\theta}(u_{j(i)})[\bar{h}(\bar{a}_n) - \bar{h}(\bar{a}_k)]\}$$

$$= v_i[\bar{\theta}(u) - \bar{\theta}(u_{j(i)})][\bar{h}(\bar{a}_n) - \bar{h}(\bar{a}_k)].$$

But $u+\bar{a}_k \geq u_{j(i)} + \bar{a}_n$, by the maximality of $g[S_1(N)]$, while $\bar{a}_k \leq \bar{a}_n$, implying $u \geq u_{j(i)}$.

Thus

$$\bar{\theta}(u) \leq \bar{\theta}(u_{j(i)}),$$

while

$$\bar{h}(\bar{a}_n) \geq \bar{h}(\bar{a}_k),$$

implying the desired inequality. This completes the proof of the lemma.

We now state the special solution algorithm which applies when condition (2.1) is satisfied. It is assumed that we are dealing with the transformed functions $\bar{h}$ and $\bar{\theta}$, together with transformed a_i's, in this first version

of the algorithm. Such transformation will be shown to be unnecessary in the corollary to theorem 1.

The algorithm is described (for fixed m) by recursion on the number n of sorting nodes, with d given by (1.1). Since the solution for n=1 is obvious, we give the recursion step from n-1 to n.

SPECIAL ALGORITHM. Number the classes and machines so that (2.9) and (2.10 - 2.10') hold. Consider a new problem with the d-m original classes m+1, m+2, . . ., d plus a new class d+1 with

$$v_{d+1} = \bar{\theta}(\bar{a}_n) \sum_1^m v_i, \tag{2.14}$$

and with machine-set $\{M_j\}_1^{n-1}$ and associated $\{\bar{a}_j\}_1^{n-1}$. By the recursion hypothesis, an optimal network N_{n-1}° for this new problem is available. In this network, "expand" the destination node corresponding to class d+1 to a sorting node occupied by M_n with outgoing arcs terminating in destination nodes associated with classes 1, 2, . . ., m. This yields a sorting network N_n° for the original problem; N_n° is an optimal solution to this problem.

Less formally, the algorithm's instructions are as follows. Choose m classes with the smallest v_i's, and a machine M_j with the largest or smallest $\bar{a}_j$ according as $\bar{h}$ is increasing or decreasing. Destination nodes corresponding to the chosen classes will terminate arcs from a final sorting node occupied by the chosen machine. Now apply the same instruction to a situation involving all the remaining machines, and all the remaining classes plus an artificial one whose "v_i" is $\bar{\theta}(\bar{a}_j)$ times the sum of the m smallest v_i's; the node occupied by M_j is then identified with the "new" destination node corresponding to the artificial class. Repeat the process (reducing the number of classes by m-1 each time) until one class is left.

THEOREM 1. The special algorithm produces an optimal sorting network when h satisfies (2.1).

PROOF. The proof is by induction on n, where d is given by (1.1). Let N be any sorting network for the n-node problem, and let x be any final sorting node with disutility a which sorts to destinations j(1), j(2), ...j(m). Let N' be another sorting network with n-1 sorting nodes derived from N by replacing destinations j(1), j(2), . . ., j(m) with a destination d+1 having volume $\bar{\theta}(a) \sum_{i=1}^{m} v_{j(i)}$ and replacing node x with a destination node labeled d+1. Then it is clear that if N and N' have the same scoring function $\bar{h}$, we have

$$f(N) - f(N') = \left(\sum_{i=1}^{m} v_{j(i)}\right) \bar{h}(u+a) - \bar{\theta}(a)\left(\sum_{i=1}^{m} v_{j(i)}\right) \bar{h}(u)$$

$$= \left(\sum_{i=1}^{m} v_{j(i)}\right)[\bar{h}(u+a) - \bar{h}(a)] - \bar{\theta}(a)\left(\sum_{i=1}^{m} v_{j(i)}\right)[\bar{h}(u) - \bar{h}(0)]$$
$$+ \bar{h}(a) \sum_{i=1}^{m} v_{j(i)} - \bar{\theta}(a)\bar{h}(0) \sum_{i=1}^{m} v_{j(i)}$$
$$= [\bar{h}(a) - \bar{\theta}(a)\, \bar{h}(0)] \sum_{i=1}^{m} v_{j(i)}, \text{ by (2.2).}$$

Now let N_1 be any optimal network on the original n sorting nodes. By lemma 1, we may assume that there is a final sorting node in N labeled n which sorts to destinations 1 through m. Let N_1' be the network on n-1 sorting nodes, with destination m+1, m+2, . . ., d, d+1 and sorting disutilities $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{n-1}$ derived from N_1 by replacing sorting node n and its destination nodes by a destination node d+1, where

$$v_{d+1} = \bar{\theta}(\bar{a}_n) \sum_{i=1}^{m} v_i.$$

By the above remarks, we have

$$f(N_1) - f(N_1') = [h(\bar{a}_n) - \bar{\theta}(\bar{a}_n)\, h(0)] \sum_{i=1}^{m} v_i. \tag{2.15}$$

But if N°_{n-1} and N°_n are, respectively the networks produced by the algorithm operating on the reduced set of machines and destinations and on the original set, then it is also clear that

$$f(N^\circ_n) - f(N^\circ_{n-1}) = [h(\bar{a}_n) - \bar{\theta}(\bar{a}_n)\, h\,(0)] \sum_{i=1}^{m} v_i, \tag{2.16}$$

from the construction given in the algorithm. But since N°_{n-1} is optimal, by the induction assumption, it follows that $f(N^\circ_{n-1}) \leq f(N_1')$, and hence from (2.15) and (2.16) that $f(N^\circ_n) \leq f(N_1)$. Thus N°_n is optimal and the theorem is proven.

Next we show that the conversion function ϕ of (1.9) is really unnecessary.

COROLLARY. In the algorithm, let the function θ be substituted for $\bar{\theta}$ and the original values of the a_i be substituted for their transformed values. Then the algorithm still gives the optimal solution.

PROOF. Clearly

$$\bar{\theta}(\bar{a}_n) = \theta\phi(\phi^{-1}(a_n)) = \theta(a_n), \tag{2.17}$$

while

$$a_j \leq a_k \text{ if and only if } \bar{a}_j \leq \bar{a}_k,$$

and θ is monotone increasing (decreasing) if and only if $\bar{\theta}$ is monotone increasing (decreasing), by the fact that ϕ is monotone increasing. Thus the ordering of subscripts for the a's is not changed, so that M_n is used

in the recursion step of the algorithm, while by (2.17) the v_{d+1} defined by either method is the same. Thus each application of the algorithm gives the same result under the original as under the transformed system, and so the final network is the same.

3. A DYNAMIC PROGRAM

In this section we obtain a dynamic programming formulation for the problem of minimizing

$$f(N) = \sum_{1}^{d} v_i \, h[L_i(N)], \tag{3.1}$$

where as before $L_i(N)$ is the number of sorting nodes, in N, in the path from the origin node to the destination node associated with class i. Note that the quantities $\{a_j\}_1^n$ do not appear explicitly in this problem. The function h is assumed monotone increasing. Two examples will be discussed in Section 4.

As in Section 2, we adopt the numbering convention

$$v_1 \leq v_2 \leq \cdots \leq v_d. \tag{3.2}$$

Using a standard theorem[8] on "rearrangements," we find that there must be an optimal N with

$$L_1(N) \geq L_2(N) \geq \cdots \geq L_d(N). \tag{3.3}$$

Define N to be an M-stage network if it satisfies (3.3) and has

$$L_1(N) = \max_i L_i(N) = M. \tag{3.4}$$

This clearly requires $M \leq n$.

In terms of the problem data $\{v_i\}_1^d$, define an M-stage network N' to be of type (s,z) if it satisfies (3.3), has s sorting nodes ($s \leq n$) and therefore

---

[8] See Chapter 10 of Inequalities, Hardy, Littlewood and Polya, Cambridge University Press (1952).

$$D(s) = s(m-1) + 1 = d - (n-s)(m-1)$$

destination nodes, and has $D(s)$ classes with data $\{v'_{d+1-k}\}_{k=1}^{D(s)}$

such that

$$v'_{d+1-k} = v_{d+1-k} \qquad \text{for } 1 \leq k \leq D(s) - z, \tag{3.5}$$

$$v'_{d+1-k} = 0 \qquad \text{for } D(s) - z < k \leq D(s), \tag{3.6}$$

$$v'_{d+1-k} = 0 \qquad \text{implies } L_{d+1-k}(N') = M. \tag{3.7}$$

The original problem is to find an optimal network of type $(n,0)$. For this it suffices to find, for each $M \leq n$, an M-stage network which is optimal in the class of M-stage type $(n,0)$ networks. To employ a dynamic programming approach, we imbed this in the following class of problems: For each triple (M,s,z) with $M \leq s \leq n$ and $z \leq D(s)$ for which an M-stage type (s,z) network exists, find an optimal M-stage network of type (s,z). Let

$f_M(s,z)$ = minimum value of $f(N')$ over all M-stage networks N' of type (s,z).

Clearly f_1 is defined only for $s=1$ and $z \leq m$, and is given by

$$f_1(1,z) = \left(\sum_{k=1}^{m-z} v_{d+1-k}\right) h(1).$$

To complete the dynamic programming formulation, we develop a recursion expressing f_{M+1} in terms of f_M.

Let N'_{M+1} be an optimal (M+1)-stage type (s,z) network. In N'_{M+1}, consider all final nodes of chains of length M+1, and all destination nodes which follow them; by (3.6)-(3.7), these include <u>at least</u> the destination nodes associated with the first z classes of N'_{M+1}, that is, the classes labeled D(s)-z+1 through D(s), whose volume is 0 as stated in (3.6). Suppose these final sorting nodes are q in number so that $qm \geq z$. Collapse each one of these sorting nodes, together with the m destination nodes which follow it, to a single new destination node associated with a new class having "$v'_i = 0$". This yields an M-stage network N'_M, of type (s-q,q), such that

$$f(N'_{M+1}) - f(N'_M) = h(M+1) \sum_{k=D(s)-qm}^{D(s)-z} v_{d+1-k}$$

Moreover the construction is reversible, in the following sense: Given any M-stage type (s-q,q) network N_M and a value of z with $z \leq qm$ and $z \leq D(s)$, the destination nodes of N_M corresponding to its first q classes can each be "expanded" to a sorting node followed by m destination nodes, in such a way that an (M+1)-stage type (s,z) network results. From these considerations we obtain the desired recursion,

$$f_{M+1}(s,z) = \min_q \{ f_M(s-q,q) + \sum_{k=D(s)-qm}^{D(s)-z} v_{d+1-k} \}, \qquad (3.8)$$

where the range for q can be limited by the conditions $z \leq qm \leq D(s)$.

4. EXAMPLES FOR SECTION 3

Two practical problems which are amenable to the dynamic program solution of section 3 are as follows. In the first instance, it is desired to minimize a cost of sorting, which is composed of a cost for each unit of material sorted by each machine plus a cost for material which is lost (or misdirected) at a given sorting. If we assume that a fraction $q \leq 1$ of the input to a sort remains after sorting, that the cost per unit per sort is a, and that the cost for a unit lost in the sorting process is L, then the function h of section 3 is given by

$$h(M) = a(1+q+\ldots+q^{M-1}) + L(1-q^{M}),$$

and so

$$h(M+1) - h(M) = aq^{M} + L(q^{M} - q^{M+1})$$

$$= q^{M}(a+L(1-q)) > 0$$

Thus h is monotonic, and the method of section 3 applies.

The second problem is as follows. Part of the material being sorted is lost at each sort, but everything lost in the process of sorting is reinserted at the initial node for another sort, and this process is repeated until everything has come through the sorting network. This might apply, for instance, to the sorting of mail. The objective is to minimize the total cost of sorting, where the cost per sort per machine is a and a fraction q of the input to a sorting node comes through without loss.

For a single sort, if an amount v_i is inserted at the origin bound for destination i, and if the length of the path P_i is M, then the cost is

$$v_i a(1+q+\ldots+q^{M-1}).$$

However, since $q^M v_i$ comes through the process, an amount pv_i, where $p=1-q^M$, is lost and must be put through again. On the third time, $p^2 v_i$ is inserted, and so on. The total cost for an initial amount v_i is therefore

$$\begin{aligned}
&(1+p+p^2+\ldots)\ v_i a(1+q+\ldots+q^{M-1}) \\
&= \left(\frac{1}{1-p}\right)\ v_i a\ \left(\frac{1-q^M}{1-q}\right) \\
&= v_i\ a \sum_{j=1}^{M} q^{-j}.
\end{aligned}$$

Thus

$$h(M) = a \sum_{j=1}^{M} q^{-j}$$

and

$$h(M+1) - h(M) = aq^{-M-1} > 0,$$

proving monotonicity of h, so that Section 3 applies.


DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS