MATHEMATICS OF OPERATIONS RESEARCH
Vol. 23, No. 4, November 1998
Printed in U.S.A.

A GENERAL CLASS OF GREEDILY SOLVABLE LINEAR PROGRAMS

MAURICE QUEYRANNE, FRITS SPIEKSMA, AND FABIO TARDELLA

A greedy algorithm solves a dual pair of linear programs where the primal variables are associated to the elements of a sublattice B of a finite product lattice, and the cost coefficients define a submodular function on B. This approach links and generalizes two well-known classes of greedily solvable linear programs. The primal problem generalizes the (ordinary and multi-index) transportation problems satisfying a Monge condition (Hoffman 1963; Bein et al. 1995) to the case of forbidden cells where the nonforbidden cells form a sublattice. The dual problem generalizes to an arbitrary finite product lattice the linear optimization problem over submodular polyhedra (Lovász 1983; Fujishige and Tomizawa 1983), which stemmed from the work of Edmonds (1970) on polymatroids. Our model and results also encompass the dual pairs of linear programs and their greedy solutions defined by Lovász (1983) for the special case of the Boolean algebra, and by Faigle and Kern (1996) for the case of so-called ''rooted forests.'' We also discuss relationships between Monge properties and submodularity, and present a class of problems with submodular costs arising in production and logistics.

1. Introduction. An important class of simple and efficient algorithms for optimizing a function f on a set S is the class of *greedy* (or myopic) algorithms. Since the work of Edmonds (1970, 1971) on matroids and of Hoffman (1963) on transportation problems, numerous authors have studied conditions on f and S which guarantee the convergence of greedy algorithms to optimal solutions. In the case where f is linear and S is a polyhedron, two broad and well-known classes of linear programs have been shown to be optimally solvable by a greedy algorithm. Following the work of Edmonds, the first class includes optimization problems on polymatroids and related submodular polyhedra; see Frank and Tardos (1988) and Fujishige's monograph (Fujishige 1991) for in-depth studies. On the other hand, following the work of Hoffman, the second class includes transportation problems, both ordinary and multi-index, with cost coefficients satisfying some form of a so-called *Monge condition* (see Hoffman 1985, and Bein et al. 1995, for details).

In this paper, we present a dual pair of linear programs, in which the variables are associated with the elements in a sublattice of a discrete product lattice. We show that a greedy algorithm solves both the primal program and the dual program when the cost coefficients in the primal problem (or, equivalently, the right-hand sides in the dual problem) are given by a submodular function on the sublattice. The primal problem generalizes the multi-index transportation problem of Bein et al. (1995) to the case of forbidden arcs, where the nonforbidden cells form a sublattice. The dual problem generalizes the linear optimization problems on submodular polyhedra by Lovász (1983) and Fujishige and Tomizawa (1983) (which are extensions of the polymatroid optimization problem of Edmonds 1971), to a distributive sublattice of a finite product space. Lovász considers a similar pair of dual linear programs in the case of the Boolean algebra, and describes their greedy solutions. A dual pair of linear programs, related to that in the present paper (see §2 below for details), is also presented by Faigle and Kern (1996).

Received July 20, 1993; revised April 21, 1997; January 6, 1998.
AMS 1991 subject classification. Primary: 90C05.
OR/MS subject classification. Primary: Programming/Linear.
Key words. Greedy algorithms, Monge property, multi-index transportation problems, lattices, submodular functions, submodular systems, polymatroids.

0364-765X/98/2304/0892/$05.00

In addition to enlarging the class of linear programs solvable by a greedy algorithm, our work also links heretofore separate streams of research and highlights the duality relationship between (multi-index) transportation problems and linear optimization problems on submodular polyhedra. In particular, we observe that submodularity and the Monge condition are the same concept expressed in different forms (see also Burkard et al. 1996). Indeed, known results on lattices and submodular functions are independently rediscovered in the context of the Monge condition for multi-dimensional arrays. Conversely, new results for Monge matrices also apply to submodular functions.

The content of this paper is as follows. In §2, we define the dual pair of linear programs which is the object of this paper. We show how they generalize multi-index transportation problems and linear optimization problems on submodular polyhedra and relate to the submodular linear programs on a rooted forest by Faigle and Kern. In §3, we present the greedy algorithm and prove that it produces optimal solutions to the primal and the dual problems. In §4, we discuss relations between submodularity, the Monge condition for a matrix and the existence of Monge sequences. Finally, in §5, we exhibit a class of problem instances with a submodular cost function. These problem instances arise in some manufacturing and logistics environments.

2. Lattices, submodular functions, and a dual pair of linear programs.

Let the integer $k \geq 2$ denote the dimension of the product lattice defined below, and $K \doteq \{1, \dots, k\}$. For $i \in K$, let A_i be a totally ordered set (or *chain*) with $m(i) + 1$ elements. For simplicity, we let $A_i \doteq \{0, 1, \dots, m(i)\}$, with the usual total order $0 < 1 < \cdots < m(i)$, for all $i \in K$. The *product space* $A \doteq A_1 \times A_2 \times \cdots \times A_k$ is a distributive lattice with *join* and *meet* operations defined by

$$a \vee b \doteq (\max\{a(1), b(1)\}, \dots, \max\{a(k), b(k)\})$$

and

$$a \wedge b \doteq (\min\{a(1), b(1)\}, \dots, \min\{a(k), b(k)\}),$$

respectively, where a and b are any elements of A. As is well known in lattice theory (see references below), these operations induce the usual *partial order* ‘‘$\leq$’’ on this lattice A by

$$a \leq b \Leftrightarrow a = a \wedge b \qquad (\Leftrightarrow b = a \vee b).$$

The associated *strict* partial order ‘‘$<$’’ is defined by $a < b$ if and only if $a \leq b$ and $a \neq b$. The ‘‘dual’’ partial orders ‘‘$\geq$’’ and ‘‘$>$’’ are defined similarly. Let $0 \doteq (0, \dots, 0)$ and $m \doteq (m(1), \dots, m(k))$ denote the smallest and largest element of A, respectively.

Let B denote any subset of A. For any $i \in K$ and $j \in A_i$ we define the *section* $B(i, j)$ of B at (i, j) as $B(i, j) \doteq \{a \in B\colon a(i) = j\}$. For any $a \in A$ we define the (*lower*) *truncation* B_a of B at a as $B_a \doteq \{b \in B\colon b \leq a\}$. Element $a \in B$ is *minimal* (resp., *maximal*) in B if no element $b \in B$ satisfies $b < a$ (resp., $b > a$). Thus, a is minimal in B if and only if the truncation $B_a = \{a\}$.

A subset B of A is a *sublattice* if for every $a, b \in B$ we have $a \vee b \in B$ and $a \wedge b \in B$. If B is a sublattice then the sections $B(i, j)$ and the truncations B_a are also sublattices, for all $i \in K$, $j \in A_i$ and $a \in A$.

A real-valued function $f\colon B \mapsto \Re$ on a sublattice B is *submodular* if the following *submodular inequality*

$$f(a \vee b) + f(a \wedge b) \leq f(a) + f(b)$$

holds for all $a, b \in B$. It is *strictly submodular* if this inequality is strict whenever $a \vee b \notin \{a, b\}$. See, for example, Birkhoff (1967) for a general exposition of lattice theory, and Topkis (1978), Veinott (1989) and Granot and Veinott (1985), and the references therein, about product lattices, their sublattices, and submodular functions (called ''subadditive'' functions in the latter reference).

Let B be any subset of A and let $B^* \doteq B \setminus \{0\}$. We associate a *cost* $w(a) \in \Re$ with every element $a \in B^*$, and a nonnegative *demand* $d_{ij} \in \Re_+$ with every section $B(i, j)$ where $i \in K$ and $j \in A_i^* \doteq A_i \setminus \{0\}$. The zero element in each chain may be used to define a slack (or artificial) variable for the corresponding demand constraint in the primal problem below. The zero elements are also convenient for modelling special cases of the dual problem, as detailed later in this section.

We now formulate a dual pair of linear programs (P) and (D):

$$(P)\qquad \begin{aligned} \min\quad & \sum_{a \in B^*} w(a) x_a \\ \text{s.t.}\quad & \sum_{a \in B(i,j)} x_a = d_{ij} \quad \text{for } i \in K \text{ and } j \in A_i^*; \\ & x_a \geq 0 \quad \text{for all } a \in B^*; \end{aligned}$$

and

$$(D)\qquad \begin{aligned} \max\quad & \sum_{i \in K} \sum_{j \in A_i^*} d_{ij} y_{ij} \\ \text{s.t.}\quad & \sum_{\substack{i \in K \\ a(i) \neq 0}} y_{i,a(i)} \leq w(a) \quad \text{for all } a \in B^*. \end{aligned}$$

The linear programs (P) and (D) contain the following problems as special cases.

Multi-index transportation problems. A special case of the primal problem (P) just defined is the following *axial k-index transportation problem with forbidden arcs*. The k sets $A_1^*, \ldots, A_k^*$ may be interpreted as sets of origins, destinations, types of goods, and various related resources. Let B denote the subset of $A' \doteq A_1^* \times \cdots \times A_k^*$ consisting of nonforbidden (permissible) combinations $a \in A'$. With each section $B(i, j)$ we associate a nonnegative ''demand'' (which may be interpreted as a supply when A_i^* is a set of origins, and as a capacity when A_i^* is a set of resources). It is assumed that $\sum_{j \in A_i^*} d_{ij} = D$, a constant for all $i \in K$. With each element $a \in B$, often called an ''arc,'' we associate a cost rate $w(a)$ and a nonnegative decision variable x_a representing the amount of flow which will satisfy the ''demand'' for each origin, destination, type of good, etc., which form element a. The (axial) k-index transportation problem is to determine the amount associated with each permissible arc $a \in B$ so as to satisfy exactly the demand of each section $B(i, j)$ (for all $i \in K$ and $j \in A_i^*$) at minimum total cost. This problem may be formulated precisely as an instance of problem (P).

The case where $B = A'$ (no forbidden arcs) is the axial multi-index transportation problem defined by Haley (p. 376 in Haley 1962; see also Chapter 8 in Yemelichev et al. 1984, and the recent papers by Bein et al. 1995, and Queyranne and Spieksma 1997). The *axial k-index assignment problem* arises when all $m(i)$ are equal to a constant m, all demands are equal to 1, and all variables x_a are restricted to be 0 or 1; see, e.g., Pierskalla (1968) and Bandelt et al. (1994). When, in addition, $k = 3$, we have the much studied (axial) *three-index assignment problem*; see Frieze and Yadegar (1981), Balas and Saltzman (1989), and Crama and Spieksma (1992). The above references describe several practical applications of these different models in such areas as logistics, automated production, statistics, and course scheduling. (For work on nonaxial versions of multi-index transportation problems, see Chapter 8 in Yemelichev et al. 1984, Korsnikov 1988, and the references therein.)

Note that, in addition to offering a concise notation (compared with the above references), our formulation of problem (P) allows us quite naturally to exclude forbidden arcs. Ordinary (i.e., 2-index) transportation problems with forbidden arcs were considered by Shamir and Dietrich (1990) in connection with the existence of Monge sequences; see §4 for details. Note also that we do not need to require in problem (P) that the total demand $\Sigma_{j \in A_i^*} d_{ij}$ be constant for all $i \in K$. Interpreting lattice elements $a \in A^*$ with some (but not all) component $a(i) = 0$ as ''slack variables,'' problem (P) also allows some of the constraints to be inequalities, and also to share common ''slack'' variables.

Submodular polyhedra. When $m(i) = 1$ for all $i \in K$, the lattice A may be identified with the lattice 2^K of all subsets of K. Sublattices B are then (distributive) set lattices, and submodular functions coincide with those now well known in combinatorial optimization (see, e.g., Nemhauser and Wolsey 1988). Assuming $w(\emptyset) = 0$, the constraints of problem (D) are then precisely those which define a *submodular polyhedron* as in Fujishige (1991; see also Frank and Tardos 1988). When $B = A = 2^K$ we have the problem which Lovász (1983) shows to be solvable by a greedy algorithm. These polyhedra are closely related to the *polymatroids* introduced by Edmonds (1971); see the preceding references for details.

Problem (D) properly generalizes these submodular polyhedra by allowing each chain A_i in the lattice to contain any number of elements, giving rise to arbitrary (finite) product lattices. This is akin to extending attention, in integer programming, from binary variables to general integer-valued variables. We refer to the work of Topkis, Veinott, and Granot and Veinott cited above for a description of some of the problems amenable to this broader framework.

Submodular linear programs on forests. Let $P = (E, \preceq)$ be a poset with finite ground set E, and let $\mathcal{A}$ be a set of subsets of E. For each $I \in \mathcal{A}$ denote by I^+ the set of maximal elements of I. In a paper (Faigle and Kern 1996) subsequent to the conference proceedings version (Queyranne et al. 1993) of the present paper, Faigle and Kern define the following linear program:

$$(FK) \qquad \begin{aligned} \max \quad & \sum_{e \in E} c_e z_e \\ \text{s.t.} \quad & \sum_{e \in I^+} z_e \leq f(I) \quad \text{for all } I \in \mathcal{A} \end{aligned}$$

where $c\colon E \mapsto \Re$ is a given weight function, and $f\colon \mathcal{A} \mapsto \Re$ is a given set function. Faigle and Kern observe that, when the sublattice B coincides with the whole lattice A, problem (D) defined above is a special case of (FK), where poset P consists of the union of all the disjoint chains A_i^*. We now show that, conversely, every problem of form (FK) can be transformed into a special case of problem (D).

We first note that in (FK) we can assume without loss of generality that $\mathcal{A}$ is a set of order ideals of P. Recall that $I \subseteq E$ is an *order ideal* of P if $i \preceq j$ and $j \in I$ imply $i \in I$, for all $i, j \in E$. Indeed, if $\mathcal{A}$ is not a set of order ideals of P then, for all $I \in \mathcal{A}$, define $\tilde{I}$ as the smallest order ideal containing I. We may then replace $\mathcal{A}$ and f in (FK) with $\tilde{\mathcal{A}} = \{\tilde{I}\colon I \in \mathcal{A}\}$ and $\tilde{f}(\tilde{I}) \doteq \min\{f(J)\colon J \in \mathcal{A} \text{ and } \tilde{J} = \tilde{I}\}$, respectively. Accordingly, we shall assume henceforth that $\mathcal{A}$ is a set of order ideals of P.

In much of their paper, Faigle and Kern assume that the set $\mathcal{A}$ is a *lattice* of order ideals of P satisfying some additional condition, and that the poset P is a rooted forest. Recall that a poset P is a *rooted forest* if every element in E has at most one upper neighbor (immediate successor).

PROPOSITION 2.1. *For every instance of problem* (FK), *there exists an instance of problem* (D) *such that*:

(i) *for every optimal solution* y^* *to* (D) *the vector* z^* *defined by* $z_e^* = \sum_{i \in K: e \in A_i} y_{ie}^*$ *for all* $e \in E$ *is an optimal solution to* (FK) *and the objective values* dy^* *and* cz^* *coincide*; *and*

(ii) *if* P *is a rooted forest and* $\mathcal{A}$ *is a lattice of order ideals of* P, *then* B *is a sublattice of* A *and*, *furthermore*, w *is submodular if and only if* f *is submodular*;
where K *denotes the index set*, A *the product lattice*, $B \subseteq A$ *the subset of lattice elements*, *and* w *the cost function in the resulting instance of problem* (D).

PROOF. Given an instance of (FK), we first show how to construct the corresponding instance of (D). Add to P an element 0 with $0 \le e$ for all $e \in E$, and let $P_0 = (E_0, \le)$ denote the resulting poset, where $E_0 \doteq E \cup \{0\}$. Let $A_1, \dots, A_k$ denote all the maximal chains in P_0 and $K \doteq \{1, \dots, k\}$. Note that, for every $i \in K$ and $I \in \mathcal{A}$ such that $I^+ \cap A_i$ is nonempty, this intersection consists of a single element, which we denote by $a_I(i)$; if this intersection is empty, we let $a_I(i) = 0$. For every $I \in \mathcal{A}$, we let $a_I \doteq (a_I(1), \dots, a_I(k))$, so $a_I \in A \doteq A_1 \times \cdots \times A_k$. Consider the mapping ϕ: $\mathcal{A} \mapsto A$ defined by $\phi(I) \doteq a_I$. Since I^+, and therefore a_I, uniquely defines the order ideal I, it follows that ϕ is a one-to-one (injective) mapping. Let $B \doteq \phi(\mathcal{A})$ and define w: $B \mapsto \Re$ by $w(a) \doteq f(\phi^{-1}(a))$. Finally, for all $e \in E$ let $d_{ie} \doteq c_e$ for all $i \in K$ such that $e \in A_i$. Problem (FK) is then transformed into problem (D) by setting $z_e = \sum_{i \in K: e \in A_i} y_{ie}$ for all $e \in E$. This implies the validity of statement (i).

For (ii), now assume that P is a rooted forest and $\mathcal{A}$ is a lattice of order ideals of P. For any ideal $I \in \mathcal{A}$, let $I_0 \doteq I \cup \{0\}$. We have $a_I(i) = I_0^+ \cap A_i = (I_0 \cap A_i)^+$ for every $i \in K$, where the last equality holds since P is a rooted forest. Furthermore, $I_0 \cap A_i$ is the subchain $[0, \dots, a_I(i)]_i$ of A_i formed by all elements $e \in A_i$ such that $0 \le e \le a_I(i)$. Hence, for all $i \in K$ and order ideals I, $J \in \mathcal{A}$,

$$
\begin{aligned}
a_{(I \cap J)}(i) &= ((I \cap J)_0 \cap A_i)^+ = ((I_0 \cap A_i) \cap (J_0 \cap A_i))^+ \\
&= ([0, \dots, a_I(i)]_i \cap [0, \dots, a_J(i)]_i)^+ = a_I(i) \wedge a_J(i),
\end{aligned}
$$

that is, $\phi(I \cap J) = \phi(I) \wedge \phi(J)$ for all I, $J \in \mathcal{A}$. In a similar manner one can prove that $\phi(I \cup J) = \phi(I) \vee \phi(J)$ for all I, $J \in \mathcal{A}$. Therefore the mapping ϕ is a lattice homomorphism between $\mathcal{A}$ and A and thus the image B of $\mathcal{A}$ is a sublattice of A. Furthermore, it follows from the definition that the mapping w: $B \mapsto \Re$ defined above is submodular on B whenever f is submodular on $\mathcal{A}$. □

Note that, in the proof of Proposition 2.1, the size of the resulting instance of (D) is polynomially bounded in (in fact, essentially identical to) the size of the given instance of (FK).

The Greedy Algorithm in the next section generalizes the North-West corner rule for ordinary transportation problems and its multi-index extension due to Bein et al. It also generalizes the greedy algorithms in Lovász and in Fujishige and Tomizawa for submodular polyhedra, the latter two being themselves generalizations of that of Edmonds for polymatroids. The greedy algorithm of the next section also applies to problem (FK) under the assumptions of Proposition 2.1(ii).

3. A greedy algorithm. We first describe the input and output of our algorithm for problems (P) and (D). Throughout this section, we assume that B is any sublattice of A with $0 \in B$, and that $w(0) = 0$.

Input:

integer k	the dimension of A;
integers $m(1), \ldots, m(k)$	defining the range of each coordinate of A;
reals $d_{ij} \geq 0$	demands, for all $i \in K$, $j \in A_i^*$;
oracle MAXLE	describing sublattice B (see explanations below);
oracle w	returning the value $w(b)$ for any $b \in B$.

Output:

variable *Status* indicating the status, *Feasible* or *Infeasible*, of problem (P);
and if *Status* = *Feasible*:

list $((b^1, x_{b^1}), \ldots, (b^T, x_{b^T}))$	describing a primal solution (see below);
reals $y_{ij} \geq 0$	describing a dual solution, for all $i \in K$ and $j \in A_i^*$.

The sublattice B might be presented in different ways, such as: A list of all elements in B (the *permissible* elements or cells); a list of all elements in $A \backslash B$ (the *forbidden* elements); collections of conditions (for example, *monotone linear inequalities*, see Veinott 1989) characterizing permissible or forbidden elements; and so forth. However, to achieve sufficient generality and to exploit the intrinsic simplicity of the Greedy Algorithm, we use the following oracle, which we call MAXLE. The input to MAXLE consists of any element $a \in A$. Oracle MAXLE then returns $\vee B_a$, the largest element of $b \in B$ such that $b \leq a$. (Recall that $0 \in B$, so sublattice B_a is nonempty for any $a \in A$.) We leave it to the interested reader which data structures can be used to efficiently implement this oracle for a given representation of the sublattice B, such as one of those outlined above.

The oracle for the function w is straightforward: Its input is any element $b \in B$ and it returns the (finite) value $w(b)$. As the Greedy Algorithm described below only calls this oracle for elements b that are already known to be in B, there is no need to include a check for the validity of $b \in B$. Recall that we assume that $w(0) = 0$.

The output to the Greedy Algorithm exploits the sparseness of the basic solutions to problem (P). Although the primal solution vector x has one component x_b for every element $b \in B^*$ (and hence potentially up to $(\prod_i (m(i) + 1)) - 1$ variables), at most $\Sigma_i \, m(i)$ of these will assume a positive value. Thus the Greedy Algorithm, which will be shown to produce a dual pair of basic solutions to problems (P) and (D), returns a list of T pairs (b, x_b) with $b \in B^*$ and x_b is the value of the corresponding variable in the solution. The number T of such pairs is determined by the algorithm, but will be shown not to exceed $\Sigma_i \, m(i)$. It is understood that $x_b \doteq 0$ for all $b \in B^*$ which do not appear in this list. Similarly, in the dual problem, we may set all variables y_{ij} for which $B(i, j) = \emptyset$ or $d_{ij} = 0$ to small enough, and otherwise arbitrary, values.

The Greedy Algorithm detailed below consists of two phases. The Primal Phase repeats the following step: identify (using the MAXLE oracle) the largest available element $b \in B^*$ and assign the largest possible value to its variable x_b. This step is repeated until either infeasibility is detected, in which case the algorithm halts, or all demands are satisfied. In the latter case, the list of (b, x_b) pairs output by the algorithm defines a feasible primal solution. The Dual Phase then traces back the sequence of elements $b \in B$ recorded in the Primal Phase to construct a dual solution y. Note that the w oracle is used only in the Dual Phase.

GREEDY ALGORITHM:

Primal Phase.

0. (Initialize):
   For all $i \in K$ and $j \in A_i^*$ do $\delta_{ij} := d_{ij}$;
   $t := 0$; $a :=$ MAXLE(m);
1. (Main Loop):
   Repeat {
      if (there exists (i, j) with $j > a(i)$ and $\delta_{ij} > 0$) then
         return (*Status* := *Infeasible*)
      else {
         $\Lambda := \{l \in K: a(l) > 0\}$;
         if $\Lambda \neq \emptyset$ then {
            $t := t + 1$; $a^t := a$;
            $x_{a^t} := \min\{\delta_{l,a^t(l)} : l \in \Lambda\}$; add (a^t, x_{a^t}) to the output list;
            for all $l \in \Lambda$ do $\delta_{l,a^t(l)} := \delta_{l,a^t(l)} - x_{a^t}$;
            $\Lambda' := \{l \in \Lambda : \delta_{l,a^t(l)} = 0\}$;
            for all $l \in \Lambda'$ do {
               $b := a^t$; $b(l) := a^t(l) - 1$;
               $b^{[l]} :=$ MAXLE(b);
               };
            let $a := b^{[i]}$ be any *maximal* element of $\{b^{[l]} : l \in \Lambda'\}$; $\eta(t) := i$;
            }
         }
      }
   until $(\Lambda = \emptyset)$;
2. (Termination):
   Let $T := t$ and $a^{T+1} := 0$; output the list $(a^1, x_{a^1}), \ldots, (a^T, x_{a^T})$;

Dual Phase.

   For all $i \in K$ and $j \in A_i^*$ do $y_{ij} := 0$;
   for $t := T$ down to 1 do
      output $y_{\eta(t),a^t(\eta(t))} := w(a^t) - \Sigma\,\{y_{\eta(u),a^t(\eta(u))}$: all $u > t$ with $a^u(\eta(u)) = a^t(\eta(u))\}$;
   return (*Status* = *Feasible*).

Note that, when applied to an ordinary (two-dimensional) transportation problem, the Primal Phase reduces to the well-known North-West corner rule (with an appropriate geographic orientation of the transportation array). More generally, the Primal Phase reduces to the greedy algorithm of Bein et al. (1995) for multi-index transportation problems without forbidden arcs.

In the case where problem (D) is a linear optimization problem over a submodular polyhedron, (that is, $A_i = \{0, 1\}$ for all $i \in K$), the Primal Phase amounts to sorting the d_{i1} values in a nondecreasing order, *consistent* with the sublattice B in the following sense: if $b(i) = 1$ for all $b \in B$ with $b(h) = 1$, and if $\eta(t) = i$ and $\eta(u) = h$, then $t > u$ (for all $h, i \in K$). Then, in the Dual Phase, the y-variables are sequentially maximized according to this sequence, with $y_{\eta(t),a^t(\eta(t))} := w(a^t) - w(a^{t+1})$ (where $w(a^{T+1}) = w(\emptyset) = 0$). Thus the Greedy Algorithm just presented reduces to that of Lovász (1983) and of Fujishige and Tomizawa (1983) in the case of submodular polyhedra.

If we only seek a primal solution, the Dual Phase may be omitted, and the Primal Phase may be simplified by replacing all the instructions between the definition of Λ' and that of $\eta(t)$ (inclusive) by the simpler instructions

let l be any element of Λ with $\delta_{l,a^t(l)} = 0$; let $a(l) := a^t(l) - 1$; $a :=$ MAXLE(a).

The instructions as originally given in the Primal Phase are required when a dual solution is sought and $B \neq A$. Their purpose is to ensure that the choice of a^{t+1} will be such that there is no element $a \in B$ with $a^t < a < a^{t+1}$. As a result, all relevant dual variables y_{ij} will be assigned a value in the Dual Phase. Any remaining dual variable will be set to zero, thus not affecting any dual constraint.

We now briefly discuss the running time of the Greedy Algorithm. First note that, since at least one component of a decreases at every iteration, the primal phase is finite and terminates after $T \leq \sum_{i=1}^{k} m(i)$ iterations. If we only seek a primal solution and implement the simpler instructions described in the preceding paragraph, each iteration is comprised of a single call to the MAXLE oracle and $O(k)$ additional work. For the original instructions, observe that every index l can belong to the set Λ' for at most $m(l) + 1$ iterations, so that the total work in the primal phase is still $O(k) \sum_{i=1}^{k} m(i)$ operations and at most $\sum_{i=1}^{k} m(i)$ calls to the MAXLE oracle. Note that, in simple cases, such an oracle call may require $O(k)$ time (e.g., when $B = A$), or $O(\sum_{i=1}^{k} \log(m(i) + 1))$ if binary search can be used on each coordinate. By storing sums $\sum \{y_{\eta(u), j}$: all u with $a^u(\eta(u)) = j\}$ of the already assigned y_{ij} values, the dual phase may be implemented to run in $O(k \sum_{i=1}^{k} m(i))$ time. Thus, and ignoring the time needed for the $\sum_{i=1}^{k} m(i)$ MAXLE oracle calls, the running time for the whole algorithm is $O(k \sum_{i=1}^{k} m(i))$. Note that, for a fixed k, this running time is linear in the number of demand data. For $k \geq 2$, and when the number of forbidden cells is, say, at most a fixed fraction of the total number of cells, this time may be considerably less than that required for simply *writing down* the $O(\prod_{i=1}^{k} m(i))$ cost coefficients, and the algorithm is effectively sublinear.

THEOREM 3.1. *Let B be a sublattice of a finite product space A, with $0 \doteq \wedge A \in B$.*

(1) *The Greedy Algorithm returns* Status = Feasible *and outputs a basic feasible solution x if and only if problem (P) is feasible.*

Let $w: B \mapsto \Re$ satisfy $w(0) = 0$.

(2) *The Greedy Algorithm outputs a basic optimal solution to problem (P) for all nonnegative feasible demands d if and only if w is submodular.*

(3) *If problem (P) is feasible and w is submodular, then the Greedy Algorithm outputs a basic optimal solution to problem (D) in the Dual Phase.*

PROOF. First, and as already observed, the algorithm is finite and terminates after at most $\sum_{i=1}^{k} m(i)$ iterations. Next, for any vector $x \in \Re^B$, let $w \cdot x \doteq \sum_{a \in B^*} w_a x_a$ and, for any section $B(i, j)$, let $x(B(i, j)) \doteq \sum_{a \in B(i,j)} x_a$. Our proof uses the following claim.

CLAIM. *Let $a, b \in B$ and let x be a feasible solution to problem (P) with $x_a > 0$ and $x_b > 0$. Define*

$$\mathrm{swap}(x, a, b) \doteq x + \epsilon(e^{a \vee b} + e^{a \wedge b} - e^a - e^b),$$

where $\epsilon \doteq \min\{x_a, x_b\}$ and e^u is the unit vector in $\Re^B$ associated with element $u \in B$ (that is, $e_v^u = 1$ if $v = u$, and 0 otherwise). Then $\mathrm{swap}(x, a, b)$ *is a feasible solution to problem (P) with component* $\mathrm{swap}(x, a, b)_{a \vee b} > x_{a \vee b}$. *Furthermore, if function $w: B \mapsto \Re$ is submodular, then $w \cdot \mathrm{swap}(x, a, b) \leq w \cdot x$.*

PROOF OF THE CLAIM. Under the assumptions of the Claim, let $x' \doteq \mathrm{swap}(x, a, b)$. By the definition of ϵ we have $x' \geq 0$. The only sections $B(i, j)$ where $x'(B(i, j))$ might differ from $x(B(i, j))$ are those where $a(i) \neq b(i)$ and $j \in \{a(i), b(i)\}$. However, we also have

$$(e^{a \vee b} + e^{a \wedge b} - e^a - e^b)(B(i, j)) = 0$$

for $j = a(i)$ and for $j = b(i)$. Hence $x'(B(i, j)) = x(B(i, j)) = d_{ij}$ for all i, j, implying that x' is a feasible solution to (P). We have $x'_{a\vee b} = x_{a\vee b} + \epsilon > x_{a\vee b}$. Finally, the inequality $w\cdot \mathrm{swap}(x, a, b) \le w\cdot x$ follows from the identity $w\cdot e^u = w(u)$ for all $u \in B$ and from the submodularity of w. The proof of the Claim is complete. $\square$

(1) Observe that the equalities $x(B(i, j)) + \delta_{ij} = d_{ij}$ (for all i, j), and the nonnegativity conditions $x \ge 0$ and $\delta \ge 0$, are all maintained at every step of the algorithm. Also observe that if the algorithm returns *Status* = *Feasible*, then all $\delta_{ij} = 0$, and the output x is therefore a feasible solution to problem (P). Conversely, assume that problem (P) is feasible. Given the recursive nature of the Greedy Algorithm, we only need to show that there exists a feasible solution x to problem (P) in which x_{a^1} equals $\xi \doteq \min\{d_{i,a^1(i)}: i \in K\}$, as prescribed by the Greedy Algorithm. Indeed, since problem (P) is feasible, let x be a feasible solution to (P) with the largest possible value of x_{a^1}. If $x_{a^1} < \xi$, then for every $i \in K$ there exists $b^i \in B(i, a^1(i))\setminus\{a^1\}$ with $x_{b^i} > 0$. Let $x^{[1]} \doteq x$ and $c^1 \doteq b^1$. For $i = 2, \ldots, k$, let $x^{[i]} \doteq \mathrm{swap}(x^{[i-1]}, c^{i-1}, b^i)$ and $c^i \doteq c^{i-1} \vee b^i$. Note that $c^k = a^1$ and, by repeated application of the above Claim, $x^{[k]}$ is a feasible solution to (P) with $x^{[k]}_{a^1} > x_{a^1}$, a contradiction with the definition of x. Hence we must have $x_{a^1} = \xi$. It remains to show that the feasible solution output by the Greedy Algorithm is basic. For this, order the constraints of problem (P) as their demands are satisfied during the execution of the primal phase of the Greedy Algorithm. These constraints and the variables x_{a^t} output in the primal phase then form a triangular, nonsingular submatrix of the constraint matrix. This matrix is a basis of the constraint matrix and the greedy primal and dual solutions are the corresponding basic solutions.

(2) First assume that w is submodular. We only need to show that there exists an optimal solution x to problem (P) in which x_{a^1} equals $\xi \doteq \min\{d_{i,a(i)}: i \in K\}$, as prescribed by the Greedy Algorithm. Indeed, let x be an optimal solution to (P) with the largest possible value of x_{a^1}. As in the proof of (1) above, we show that if $x_{a^1} < \xi$, then there exists a feasible solution $x^{[k]}$ to (P) with $x^{[k]}_{a^1} > x_{a^1}$ and, using the last part of the Claim, with $w\cdot x^{[k]} \le w\cdot x$, a contradiction with the definition of x. Hence we must have $x_{a^1} = \xi$, proving that the Greedy Algorithm correctly fixes the value of x_{a^1}.

Conversely, assume that w is an arbitrary cost-function such that the Greedy Algorithm constructs an optimal solution to problem (P) for all $d \ge 0$. For any two $a, b \in B$ distinct from their join and meet, we define a problem instance with $d_{ij} = 1$ if $j \in \{a(i), b(i)\}$, and 0 otherwise, for all $i \in K$. Thus, x' defined by $x'_u = 1$ if $u \in \{a, b\}$, and 0 otherwise, is a feasible solution to (P). However, the Greedy Algorithm applied to this problem instance produces an optimal solution x where $x_u = 1$ if $u \in \{a \vee b, a \wedge b\}$, and 0 otherwise. Therefore, we have

$$w(a \wedge b) + w(a \vee b) = w(x) \le w(x') = w(a) + w(b).$$

Since this inequality holds for any $a, b \in B$, it shows that w is submodular.

(3) Assume that problem (P) is feasible and w is submodular. Let y denote the dual solution constructed in the Dual Phase of the Greedy Algorithm. First, observe that y satisfies the complementary slackness conditions since for each variable x_{a^t} selected in the Primal Phase, the corresponding constraint in problem (D) is satisfied with equality in the Dual Phase. Since the primal solution x is optimal, it suffices to show that y is a feasible solution to (D).

Let the *path* $P \doteq (a^1, a^2, \ldots, a^T, a^{T+1} = 0)$ denote the sequence of lattice elements found in the Primal Phase of the Greedy Algorithm. Note that $a^t > a^{t+1}$ for all $t = 1, \ldots, T$, that is, the path P is a totally ordered subset of B. For any $a \in B$, the path element $a^v \doteq \wedge\{a^t: a^t \ge a\}$ is well defined (since $a^1 = \vee B \ge a$). We may thus define $\mu(a) \doteq \mathrm{dist}(a, a^v)$, where $\mathrm{dist}(a, b)$ denotes the Manhattan (or rectilinear) distance between lattice elements a and b, that is, $\mathrm{dist}(a, b) \doteq \sum_{i=1}^k |a(i) - b(i)|$. Thus $\mu(a)$ is a nonnegative integer, bounded above by $\Sigma_i\ m(i)$, and such that $\mu(a) = 0$ if and only if $a \in P$. We shall now prove by induction on $\mu(a)$ that y satisfies the constraint of (D) corresponding to every element $a \in B^*$. Indeed, we have already observed that these constraints $y(a) \leq w(a)$ hold with equality for every $a \in B$ with $\mu(a) = 0$, that is, for every $a \in P$. (For $a = 0$, this reduces to the identity $0 = w(0)$ assumed throughout.) Therefore, assume that, for any integer $r \geq 1$, all constraints of (D) corresponding to elements $a' \in B$ with $\mu(a') \leq r - 1$ hold, and consider any $a \in B$ with $\mu(a) = r \geq 1$. If we can find an index $t \in \{1, \ldots, T+1\}$ such that $\mu(a \wedge a^t) \leq r - 1$ and $\mu(a \vee a^t) \leq r - 1$, then we have, by the inductive assumption,

$$
\begin{aligned}
y(a) &= y(a \wedge a^t) + y(a \vee a^t) - y(a^t) \\
&\leq w(a \wedge a^t) + w(a \vee a^t) - w(a^t) \\
&\leq w(a)
\end{aligned}
$$

(where the last inequality follows from submodularity of w), and the inductive proof is complete. In the rest of this proof, we establish, by contradiction, the existence of such an index t for any lattice element $a \in B$ with $\mu(a) = r$.

Thus, let a be a *minimal* element of B such that $\mu(a) = r \geq 1$ and for all $t = 1, \ldots, T+1$ either $\mu(a \wedge a^t) \geq r$ or $\mu(a \vee a^t) \geq r$. Since $a^{T+1} = 0 < a < \vee B = a^1$, the path elements $a^u \doteq \vee\{a^t\colon a^t < a\}$ and $a^v \doteq \wedge\{a^t\colon a^t > a\}$ are well defined and $\mu(a) = \text{dist}(a, a^v)$. Recall that there is no lattice element $b \in B$ such that $a^{v+1} < b < a^v$, for this would contradict the definition of a^{v+1} as a *maximal* element of $\{b^{[l]}\colon l \in \Lambda'\}$ in the Primal Phase. Thus, we must have $u > v + 1$, since otherwise we would have $a^{v+1} = a^u < a < a^v$. Therefore, we have $a^u < a^{v+1} < a^v$, and a and a^{v+1} are incomparable. Since $a < a^v$ and $a^{v+1} < a^v$, we have $a \vee a^{v+1} \leq a^v$. Since $a^{v+1} < a \vee a^{v+1}$, we must then have $a \vee a^{v+1} = a^v$ (for otherwise we would again have an element $b = a \vee a^{v+1}$ such that $a^{v+1} < b < a^v$). Now note the property of the Manhattan distance that $\text{dist}(a \wedge a', a') = \text{dist}(a, a \vee a')$ for any lattice elements a and a' in B. Thus $\text{dist}(a \wedge a^{v+1}, a^{v+1}) = \text{dist}(a, a^v) = \mu(a) = r$. Therefore we have $\mu(a \wedge a^{v+1}) \leq \text{dist}(a \wedge a^{v+1}, a^{v+1}) = r$ and $\mu(a \vee a^{v+1}) = \mu(a^v) = 0 < r$. Now, whether $\mu(a \wedge a^{v+1}) \leq r - 1$ or $\mu(a \wedge a^{v+1}) = r$, we have a contradiction with the definition of a (in the latter case, with the *minimality* of a). Hence our inductive proof is complete. We have thus established that y is a feasible, and therefore optimal, solution to the dual problem (D). Furthermore, as shown in the proof of part (1) above, y is basic. The proof of the theorem is complete. □

We now briefly describe some consequences of Theorem 1. First note that, for a given feasible demand vector d, the primal solution constructed by the Greedy Algorithm does not depend on the cost function w, provided it is submodular. See Adler et al. (1990) and Adler and Shamir (1990) for a study of a similar property in the context of ordinary transportation and minimum cost network flow problems.

Recall (Hoffman 1974, and Edmonds and Giles 1974) that a system of linear inequalities $Cy \leq w$ is *totally dual integral* (*TDI*) if, for all integral d such that the maximum in $\max_y\{dy\colon Cy \leq w\}$ is finite, the dual problem $\min_x\{wx\colon C^T x = d, x \geq 0\}$ has an integral optimal solution. A consequence of Theorem 1 is that, when w is submodular, the inequalities of problem (D) form a *totally dual integral* (*TDI*) linear inequality system. See Hoffman, Edmonds and Giles, and Nemhauser and Wolsey (1988) for a discussion of *TDI* systems and their properties.

4. Submodular costs and Monge properties. The main purpose of this section is to point out and exploit the equivalence between submodularity of a function defined on a product of k chains, and the Monge condition of a k-dimensional array. We also introduce the concept of submodular sequences, and discuss its relationship with that of Monge sequences in two-dimensional arrays. In particular, we show that, for any *strictly* submodular two-dimensional array, the class of submodular sequences coincides with that of Monge sequences. Finally, we show a simple example of a two-dimensional array that admits a Monge sequence, but does not satisfy the Monge condition for any permutations of its rows and columns.

A (two-dimensional) $n \times m$ array C is called a (*strict*) *Monge array* if it satisfies the following (*strict*) *Monge condition*: For all i, j, k, l with $i < k$ and $j < l$,

$$c[i, j] + c[k, l] \leq c[i, l] + c[k, j].$$

This definition is trivially equivalent to the (strict) submodularity of the function defined by C on the product lattice $A = \{1, \ldots, n\} \times \{1, \ldots, m\}$. Furthermore, it can be straightforwardly extended to the case where the entries $c[i, j]$ of C are defined only when (i, j) belongs to a sublattice B of A. The inequality on which the above definition is based was already exploited by the French mathematician Gaspard Monge (1781) who, in the eighteenth century, observed that, in moving materials such as earth for building roads or military facilities, the paths followed by different shipments should not cross (as do the diagonal of a convex quadrangle) because the total distance travelled, and the total effort expended, would then be larger than necessary.

The Monge condition has been reintroduced with various names in different contexts: A square matrix satisfying the Monge condition is called a *distribution matrix* in Gilmore et al. (1985), when the only defined entries in the matrix are above the diagonal (thus forming a sublattice of the lattice A of matrix cells), this condition is also called the (concave) quadrangle inequality (e.g., Yao 1980), and (as if to add to the confusion) functions defined by such an array are sometimes called concave functions (e.g., Larmore and Schieber 1991). Besides the above equivalent definitions, there are numerous closely related and often weaker concepts, including totally monotone matrices (e.g., Klawe 1992) and the Gilmore-Gomory and Demidenko conditions for travelling salesman problems (Park 1991); see Aggarwal and Park (1989), Bein et al. (1995) and Burkard et al. (1996) for details. The computer science community has seen a flurry of activity on such concepts during the past few years. This activity was motivated in part by the seminal paper of Aggarwal et al. (1987) on matrix searching, and also by a wide variety of applications to problems in such areas as computational geometry (e.g., Aggarwal et al. 1987 and Aggarwal and Klawe 1990); VLSI channel routing (e.g., Aggarwal et al. 1987 and Aggarwal and Park 1989); signal quantization (Wu 1991); molecular biology (e.g., Sankoff and Kruskal, Eds 1983, and Larmore and Schieber 1991); dynamic lot sizing (the Wagner-Whitin problem; see Aggarwal and Park 1993); flow shop scheduling (van der Veen and van Dal 1991); and the travelling salesman problem (Park 1991). Because the field is now so vast, we have only given here a few indicative references; further references can be found therein and in Burkard et al. (1996).

The concept of Monge arrays has recently been extended to k-dimensional arrays by Aggarwal and Park (1989a, 1989b): A k-dimensional array C satisfies the *Monge condition* if every two-dimensional plane of C corresponding to fixed values of $k - 2$ coordinates satisfies the Monge condition.

The Monge condition just defined for the k-dimensional case is equivalent to the submodularity of the function $c\colon A \mapsto \Re$ defined by the array C. Indeed, the following result (compare Proposition 2.4 in Aggarwal and Park 1989a) is a direct rephrasing of Theorems 3.1 and 3.2 in Topkis (1978) for the case where A is a product of a finite number of chains, as is assumed throughout the present paper:

THEOREM 4.1. *A function $c\colon A \mapsto \Re$ is submodular if and only if it is submodular on every two-dimensional sublattice* (*plane*) *corresponding to fixed values of $k - 2$ coordinates*.

As a consequence, many results on k-dimensional arrays satisfying the Monge condition (such as in §2 of Aggarwal and Park 1989a, and in §2 in Bein et al. 1995) can be directly derived from known results on submodular functions. In addition, the deep theory of parametric lattice programming developed in Topkis (1978) also applies to problems involving k-dimensional arrays satisfying the Monge condition. Conversely, the rich computer science literature on Monge and related arrays, in particular the fast algorithms developed for a variety of such problems, may also be exploited to study submodular functions on discrete product lattices. (A case in point is the dynamic lot-sizing problem considered in detail in Topkis (1978), and for which fast algorithms were derived in Aggarwal and Park 1993, using the Monge condition.)

We now discuss some relations between the Monge condition for an array and the existence of a *Monge sequence*. The concept of Monge sequences was introduced for two-dimensional arrays (matrices) by Hoffman in 1963 in order to describe classes of transportation problems that are greedily solvable. A *Monge sequence* for a two-dimensional $n \times m$ array $C = (c[i, j])$ is a total ordering of the nm pairs (i, j) such that, whenever pair (i, j) precedes both pair (i, l) and pair (k, j),

$$c[i, j] + c[k, l] \leq c[i, l] + c[k, j].$$

Monge sequences have been fruitfully employed for the construction of greedy solutions which are either feasible or optimal for two-dimensional transportation problems, among others, in Adler et al. (1993), Adler and Shamir (1993), and Shamir and Dietrich (1990). They also have an interesting antimatroid interpretation (Dietrich 1990) and are closely related to notions of *greedoids* (see Korte et al. 1991).

Some authors (e.g., Bein et al. 1991, and Rudolf 1992) have recently investigated the possibility of extending this concept to higher dimensions. However, at present no approach seems to clearly dominate. Hence we restrict our discussion of relations between Monge condition and the existence of Monge sequences to the two-dimensional case. We shall employ the following notion which arises naturally in the lattice framework: A *submodular sequence* for a two-dimensional $n \times m$ array $C = (c[i, j])$ is a total ordering of the nm pairs (i, j) such that, for any i, j, k and l with $(i, j) \wedge (k, l) \notin \{(i, j), (k, l)\}$, at least one of the pairs $(i, j) \wedge (k, l)$ or $(i, j) \vee (k, l)$ precedes both pairs (i, j) and (k, l). For instance, the lexicographic sequence (in which (i, j) precedes (k, l) if $i < k$ or if $i = k$ and $j < l$) is a submodular sequence. Note that the notions of submodular sequence and of Monge sequence directly extend to the case where the entries of the array C are defined only when the indices (i, j) belong to a sublattice B of A, i.e., when some cells of C are *forbidden*. In this general context, the following proposition shows that submodular sequences and Monge sequences coincide when C satisfies the Monge condition.

PROPOSITION 4.2. *A sublattice of a two-dimensional array satisfies the Monge condition if and only if every submodular sequence is a Monge sequence.*

PROOF. *Necessity*: Assume that C is an array whose elements $c[i, j]$ satisfy the Monge condition when their indices (i, j) belong to a sublattice B of the lattice A of all pairs of indices. Consider any submodular sequence. To prove that it is a Monge sequence, consider any $i \neq k$ and $j \neq l$ such that (i, j), (i, l), (k, j) and (k, l) belong to B and (i, j) precedes both (i, l) and (k, j) in the sequence. Since neither (i, l) nor (k, j) precedes (i, j), we cannot have $\{(i, l), (k, j)\} = \{(i, j) \wedge (k, l), (i, j) \vee (k, l)\}$. Hence we must have $\{(i, j), (k, l)\} = \{(i, l), \wedge (k, j), (i, l) \vee (k, j)\}$ and, since the array C is Monge, $c[i, j] + c[k, l] \leq c[i, l] + c[k, j]$. This shows that the sequence is a Monge sequence for array C.

Sufficiency: This follows immediately by noting that a sublattice of an array is Monge if and only if the lexicographic sequence (which is a submodular sequence) is Monge. □

It is not true that, in a Monge array, every Monge sequence is a submodular sequence. For example, consider a constant array C (where all entries $c[i, j]$ have the same value): All sequences of the pairs (i, j), including those that are not submodular, are Monge sequences. However, the next result shows that Monge sequences coincide with submodular sequences when C is a *strict* Monge array:

PROPOSITION 4.3. *If a sublattice of an array C satisfies the strict Monge condition then a total ordering of the pairs of indices (i, j) of the elements of the sublattice is a submodular sequence if and only if it is a Monge sequence.*

PROOF. Assume that there exists a Monge sequence which is not a submodular sequence, that is, in which neither (i, j) nor (k, l) precedes both (i, l) and (k, j), for some $i \leq k$ and $j \leq l$. Then at least one of the pairs (i, l) and (k, j) precedes both (i, j) and (k, l) in this Monge sequence. Hence, by the definition of Monge sequences, we have $c[i, l] + c[k, j] \leq c[i, j] + c[k, l]$, contradicting the strict Monge condition for array C. Hence every Monge sequence must be submodular. The converse follows directly from Proposition 4.2. □

Note that the Monge condition may or may not be satisfied by a given array C depending on the ordering of its rows and columns. On the other hand, the existence of a Monge sequence for C is independent of such orderings. Hence, if there exists an ordering of the rows and columns under which C satisfies the Monge condition, then there exists a Monge sequence for C. However, the existence of a Monge sequence for C does not imply the existence of an ordering of its rows and columns under which C is Monge. This is shown in the following example: For $n = m = 3$, let

$$C = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 3 \\ 1 & 0 & 0 \end{pmatrix}.$$

A Monge sequence is given by $((2, 2), (1, 1), (2, 1), (0, 2), (0, 0), (1, 0), (1, 2), (2, 0), (0, 1))$. However, as can be verified by enumeration, no ordering of its rows and columns makes C Monge.

5. A class of problems with submodular costs. We now consider a class of problems where the elements of each chain A_i^* are resources of a certain type, all located in some given space where distances are defined, a *metric space*. The lattice elements $a \in A' = A_1^* \times \cdots \times A_k^*$ correspond to sets, or *clusters*, of chain elements, one from each chain, and the cost of each cluster is determined by some function of the distances between the cluster elements. For example, in a physical distribution context, the resources may include customer sites, warehouses, trucks, truck depots, plants, etc. In a flexible manufacturing application (Crama and Spieksma 1992), the resources are the part types, the feeder slots, and the insertion points on the circuit board. Note that not all clusters need be feasible. Indeed, in our model we can also deal with the case where the feasible clusters form a sublattice B of A'.

The cost rate $w(a)$ of cluster a reflects various motions and shipment activities within the cluster, and is determined jointly by the pattern of such activities within the cluster and by the distances between cluster elements. For example, w may be the total length of the *shortest spanning tree*, of the *shortest Hamiltonian path*, or of the *shortest Hamiltonian cycle* (''travelling salesman tour''), between all points in a cluster, modelling various ways in which these activities may be conducted. Two other important examples are the *weighted sum* of all distances in a cluster, and the *diameter*, i.e., the longest distance between any two elements of a cluster. This last example may reflect the minimum time to perform all the within-cluster activities simultaneously. See Bandelt et al. (1994) for a discussion of multi-index assignment problems where the points are located in a general metric space, and Queyranne and Spieksma (1997) for extensions to multi-index transportation problems.

In the following models we consider two cases in which all resources of each chain A_i^* are located on a line L_i in the Euclidean space $\Re^q$: First when all lines L_i are identical, and next when all L_i's are parallel.

In the case where all resources are located on a *single line* we let f_i be the function that associates with each element $h \in A_i^*$ its *abscissa* $f_i(h)$ and assume, w.l.o.g., that each f_i is isotone, i.e., $h < l$ implies $f_i(h) \leq f_i(l)$. Therefore the mapping $f\colon A' \to \Re^k$ defined by $f(a) \doteq (f_1(a(1)), \ldots, f_k(a(k)))$ satisfies $f(a \vee b) = f(a) \vee f(b)$ and $f(a \wedge b) = f(a) \wedge f(b)$ and hence is a lattice homomorphism from A' into $\Re^k$. This implies that S is a sublattice of $\Re^k$ iff $f^{-1}(S)$ is a sublattice of A'. Furthermore, any submodular function g on $\Re^k$ induces a submodular function $w = g \circ f$ on A' (or a sublattice thereof) defined by $w(a) = g(f(a))$.

The next proposition establishes submodularity of interesting special cases of such functions g including the diameter and weighted sum mentioned above.

PROPOSITION 5.1. *The following functions are submodular on* $\Re^k$:
(i) $g_1(x) = \max_{i \in K} x_i$,
(ii) $g_2(x) = \max_{i,j \in K} |x_i - x_j|$,
(iii) $g_3(x) = \sum_{i,j \in K} \varphi_{ij}(x_i - x_j)$, *where the* φ_{ij}*'s are convex functions from* $\Re$ *to* $\Re$.

PROOF. (i) Given any $x, y \in \Re^k$, assume w.l.o.g. that $g_1(x \vee y) = (x \vee y)_i = x_i$. Hence, $g_1(x) = x_i$. Since trivially $g_1(x \wedge y) \leq g_1(y)$, it follows that $g_1(x \vee y) + g_1(x \wedge y) \leq g_1(x) + g_1(y)$.

(ii) Note that $g_2(x) = g_1(x) + g_1(-x)$. Hence the submodularity of $g_2(x)$ follows from (i) and the simple fact that a function $g(x)$ is submodular iff $g(-x)$ is submodular.

(iii) Since the sum of submodular functions is submodular we only need to prove that $\gamma(x) = \varphi(x_i - x_j)$ is submodular when i, j are a given pair of indices and φ is convex. Given $x, y \in \Re^k$ we can assume w.l.o.g. that $x_i < y_i$ and $x_j > y_j$ (note that the case $x_i < y_i$ and $x_j < y_j$ is trivial). We then have $\gamma(x \wedge y) + \gamma(x \vee y) = \varphi(x_i - y_j) + \varphi(y_i - x_j)$ and $\gamma(x) + \gamma(y) = \varphi(x_i - x_j) + \varphi(y_i - y_j)$. Define $a = x_i - y_j$, $b = y_i - x_j$, $c = x_i - x_j$ and $d = y_i - y_j$ and observe that $a + b = c + d$. Furthermore, the assumptions $x_i < y_i$ and $x_j > y_j$ imply $c < a < d$ and $c < b < d$. Hence, from the convexity of φ we deduce $\varphi(a) + \varphi(b) \leq \varphi(c) + \varphi(d)$ which concludes the proof. □

Note that, when all resources are located on a line, function g_2 above corresponds to a cost function $w_2(a) = g_2(f(a))$ associated with the diameter of cluster a, while function g_3 can be specialized to the case of a sum of weighted distances of the resources of a cluster a by setting $\varphi_{ij}(x) = r_{ij}|x_i - x_j|$, with $r_{ij} \geq 0$. The submodularity of the sum of the weighted distances above was also proved in Tamir (1993). Function g_3 can also be applied to define cost functions which depend nonlinearly on the distances of the elements of the cluster, e.g., $\varphi_{ij}(x) = r_{ij}|x_i - x_j|^p$, with $p \geq 1$.

Now consider the case where all the resources $h \in A_i^*$ are located on (not necessarily distinct) *parallel lines* in a Euclidean space $\Re^q$ as follows: Given points $u^1, \ldots, u^k \in \Re^q$ and a common direction $v \in \Re^q$ the location of $h \in A_i^*$ is $f_i(h) = u^i + t_{ih}v$, where $t_{ih} \in \Re$. As before we assume w.l.o.g. that these locations are in the same order as the elements of A_i^*, i.e., $h < l$ implies $t_{ih} \leq t_{il}$. Consider the cost rate $w(a)$ defined by a weighted sum of the Euclidean distances $d(f_i(a(i)), f_j(a(j)))$ between points $f_i(a(i))$ and $f_j(a(j))$. The submodularity of $w(a)$ can be proved by showing that the functions

$w_{ij}(a) = d(f_i(a(i)), f_j(a(j)))$ are submodular. This latter statement is a consequence of a simple quadrangle inequality in the plane already observed by Monge (1781) (see also Aggarwal and Klawe 1990, and Aggarwal et al. 1987). Indeed, consider $a, b \in A'$ and assume $a(i) < b(i)$ and $a(j) > b(j)$ (the case $a(i) < b(i)$ and $a(j) < b(j)$ is trivial). Since $a(i) < b(i)$ and $a(j) > b(j)$ belong to parallel lines L_i and L_j, the line segment joining $f_i(a(i))$ and $f_j(a(j))$ intersects that joining $f_i(b(i))$ and $f_j(b(j))$. Hence, $w_{ij}(a \wedge b) + w_{ij}(a \vee b) = d(f_i(a(i)), f_j(b(j))) + d(f_i(b(i)), f_j(a(j))) \leq d(f_i(a(i)), f_j(a(j))) + d(f_i(b(i)), f_j(b(j))) = w_{ij}(a) + w_{ij}(b)$.

The results and methods in previous sections apply to *constrained problems* where some clusters may be forbidden, provided the allowable clusters form a sublattice of A'. Through the lattice homomorphism f mentioned above, this is the case in particular when the resources are located on a single line and the location of the resources that form allowable clusters are determined by distance (or precedence) constraints of the form $f_i(a(i)) - f_j(a(j)) \leq \delta_{ij}$, for $i, j \in K$ and $\delta_{ij} \in \Re$; see Veinott (1989) and also Tamir (1993).

When function w in the above models is submodular, the corresponding problems (P) and (D) may be solved by the Greedy Algorithm of §3. In the case where all resources are located on a single line this algorithm proceeds along the line (say) left to right in the Primal Phase, greedily saturating the primal variables, and then right to left in the Dual Phase, tracing back the sequence followed in the Primal Phase and also greedily saturating the dual variables. This algorithm takes a particularly simple form for the k-index assignment problem: If the resources in each chain i are sorted from left to right on the line then, for $j = 1, \ldots, m$, the jth cluster in an optimal solution simply consists of the jth resource of each type.

Acknowledgments. We thank Professors Ron Shamir and Ulrich Faigle for pointing out the papers by Bein et al. (1995) and Faigle and Kern (1996), respectively, and for interesting discussions. This paper also benefited from useful remarks and perceptive comments by the referees. The work of the first author was supported in part by a research grant from the National Sciences and Engineering Research Council (NSERC) of Canada and by a Visiting Professor grant from the Italian National Council of Research (CNR) at the Department of Computer Science, University of Pisa, Italy, 1992.

References

Adler, I., A. J. Hoffman, R. Shamir. 1993. Monge and feasibility sequences in general flow problems. *Discrete Appl. Math.* **44** 21–38.

———, R. Shamir. 1993. Greedily solvable transportation networks and edge-guided vertex elimination. D.-Z. Du, P. M. Pardalos, eds. *Network Optimization Problems: Algorithms, Applications and Complexity.* World Scientific, 1–22.

Aggarwal, A., M. M. Klawe. 1990. Applications of generalized matrix searching to geometric algorithms. *Discrete Appl. math.* **27** 3–23.

———, ———, S. Moran, P. Shor, R. Wilber. 1987. Geometric applications of a matrix-searching algorithm. *Algorithmica* **2** 195–208.

———, J. K. Park. 1989a. Sequential searching in multi-dimensional monotone arrays. Research Report RC 15128, IBM T. J. Watson Research Center, Yorktown Heights.

———, ———. 1989b. Parallel searching in multi-dimensional monotone arrays. Research Report RC 14826, IBM T. J. Watson Research Center, Yorktown Heights.

———, ———. 1993. Improved algorithms for economic lot-size problems. *Oper. Res.* **41** 549–571.

Balas, E., M. J. Saltzman. 1989. Facets of the three-index assignment polytope. *Discrete Appl. Math.* **23** 201–229.

Bandelt, H. J., Y. Crama, F. C. R. Spieksma. 1994. Approximation algorithms for multi-dimensional assignment problems with decomposable costs. *Discrete Appl. Math.* **49** 25–50.

Bein, W. W., P. Brucker, P. K. Pathak. 1991. Monge properties in higher dimensions. Preprint 134, Universität Osnabrück.

———, ———, J. K. Park, P. K. Pathak. 1995. A Monge property for the d-dimensional transportation problem. *Discrete Appl. Math.* **58** 97–109.

Birkhoff, G. 1967. *Lattice Theory*, 3rd ed. Amer. Math. Soc. Colloq. Publ. Vol 25, American Mathematical Society, Providence, RI.
Burkard, R. E., B. Klinz, R. Rudolf. 1996. Perspectives of Monge properties in optimization. *Discrete Appl. Math.* **70** 95–161.
Crama, Y., F. C. R. Spieksma. 1992. Approximation algorithms for three-dimensional assignment problems with triangle inequalities. *European J. Operational Res.* **60** 273–279.
Dietrich, B. L. 1990. Monge sequences, antimatroids, and the transportation problem with forbidden arcs. *Linear Algebra Appl.* **39** 133–145.
Edmonds, J. 1970. Submodular functions, matroids and certain polyhedra. R. Guy et al., eds. *Combinatorial Structures and Their Applications*. Gordon and Breach, New York, 69–87.
———. 1971. Matroids and the greedy algorithm. *Math. Programming* **1** 127–136.
———, R. Giles. 1974. A min-max relation for submodular functions on graphs. *Ann. Discrete Math.* **1** 185–204.
Faigle, U., W. Kern. 1996. Submodular linear programs on forests. *Math. Programming* **72** 195–206. (Original manuscript: Memorandum No. 1163, Faculty of Applied Mathematics, University of Twente, September 1993.)
Frank, A., É. Tardos. 1988. Generalized polymatroids and submodular flows. *Math. Programming* **42** 489–563.
Frieze, A. M., J. Yadegar. 1981. An algorithm for solving 3-dimensional assignment problems with applications to scheduling a teaching practice. *J. Operational Res. Soc.* **32** 989–995.
Fujishige, S. 1991. *Submodular Functions and Optimization*. North-Holland, Amsterdam.
———, N. Tomizawa. 1983. A note on submodular functions on distributive lattices. *J. Oper. Res. Soc. Japan* **26** 309–318.
Gilmore, P. C., E. L. Lawler, D. B. Shmoys. 1985. Well-solved special cases. E. L. Lawler et al., eds. *The Traveling Salesman Problem*. Wiley, Chichester.
Granot, F., A. F. Veinott, Jr. 1985. Substitutes, complements and ripples in network flows. *Math. Oper. Res.* **10** 471–497.
Haley, K. B. 1962. The multi-index problem. *Oper. Res.* **11** 368–379.
Hoffman, A. J. 1963. On simple linear programming problems. V. Klee, ed. *Convexity: Proceedings of the Seventh Symposium in Pure Mathematics*, Proceedings of Symposia in Pure Mathematics, Vol. 7. American Mathematical Society, Providence, RI, 317–327.
———. 1974. A generalization of max-flow min-cut. *Math. Programming* **6** 352–359.
———. 1985. On greedy algorithms that succeed. I. Anderson, ed. *Surveys in Combinatorics*. Cambridge University Press, Cambridge, 97–112.
Klawe, M. M. 1992. Superlinear bounds for matrix searching problems. *J. Algorithms* **13** 55–78.
Korte, B., L. Lovász, R. Schrader. 1991. *Greedoids*. Springer, Berlin.
Korsnikov, A. D. 1988. Planar three-index transportation problems with dominating index. *ZOR—Zeitschrift für Oper. Res.* **32** 29–33.
Larmore, L. L., B. Schieber. 1991. On-line dynamic programming with applications to the prediction of RNA secondary structure. *J. Algorithms* **12** 490–515.
Lovász, L. 1983. Submodular functions and convexity. A. Bachem et al., eds. *Mathematical Programming—The State of The Art*. Springer, Berlin, 235–257.
Monge, G. 1781. Mémoire sur la théorie des déblais et remblais. *Mémoires de Mathematiques et de Physique*, 666–704.
Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley, Chichester.
Park, J. K. 1991. A special case of the n-vertex traveling salesman problem that can be solved in $O(n)$ time. *Information Processing Letters* **40** 247–254.
Pierskalla, W. P. 1968. The multidimensional assignment problem. *Oper. Res.* **16** 422–431.
Queyranne, M., F. C. R. Spieksma. 1997. Approximation algorithms for multi-index transportation problems with decomposable costs. *Discrete Appl. Math.* **76** 239–253.
———, ———, F. Tardella. 1993. A general class of greedily solvable linear programs. G. Rinaldi and L. Wolsey, eds. *Integer Programming and Combinatorial Optimization* 3, Proceedings of the Third IPCO Conference, CORE, Louvain-la-Neuve, May 1993, 385–399.
Rudolf, R. 1992. Personal communication.
Sankoff, D., J. B. Kruskal, eds. 1983. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA.
Shamir, R., B. L. Dietrich. 1990. Characterization and algorithms for greedily solvable transportation problems. *Proc. 1st ACM/SIAM Symposium on Discrete Algorithms*, 358–366.
Tamir, A. 1993. The least element property of center location on tree networks with applications to distance and precedence constrained problems. *Math. Programming* **62** 475–496.
Topkis, D. M. 1978. Minimizing a submodular function on a lattice. *Oper. Res.* **26** 305–321.
van der Veen, J. A., R. van Dal. 1991. Solvable cases of the no-wait flow-shop scheduling problem. *J. Operational Res. Soc.* **42** 971–980.

Veinott, A. F., Jr. 1989. Representation of general and polyhedral subsemilattices and sublattices of product spaces. *Linear Algebra Appl*. **114/115** 681–704.

Wu, X. 1991. Optimal quantization by matrix searching. *J. Algorithms* **12** 663–673.

Yao, F. F. 1980. Efficient dynamic programming using quadrangle inequalities. *Proc*. 12*th ACM Symposium on Theory of Computing*, 429–435.

Yemelichev, V. A., M. M. Kovalev, M. K. Kratsov. 1984. *Polytopes, Graphs and Optimisation*. Cambridge University Press, Cambridge.

M. Queyranne: Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, Canada V6T 1Z2; e-mail: Maurice.Queyranne@commerce.ubc.ca

F. Spieksma: Department of Mathematics, Maastricht University, P.O. Box 616, NL-6200 MD Maastricht, The Netherlands; e-mail: spieksma@math.unimaas.nl

F. Tardella: Department of Mathematics, Faculty of Economics, Universitá degli Studi di Roma ‘‘La Sapienza’’, Via del Castro Laurenziano 9, 00161 Roma, Italy; e-mail: tardella@imc.pi.cnr.it