INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again – beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

University Microfilms International
300 North Zeeb Road
Ann Arbor, Michigan 48106 USA
St. John's Road, Tyler's Green
High Wycombe, Bucks, England HP10 8HR

7816132

KORNBLUM, DEBORAH FREEDMAN
"GREEDY" ALGORITHMS FOR SOME OPTIMIZATION
PROBLEMS ON A LATTICE POLYHEDRON.

CITY UNIVERSITY OF NEW YORK, PH.D., 1978

University
Microfilms
International 300 N. ZEEB ROAD, ANN ARBOR, MI 48106

"GREEDY" ALGORITHMS
FOR SOME
OPTIMIZATION PROBLEMS
ON A
LATTICE POLYHEDRON

by

Deborah Freedman Kornblum

A dissertation submitted to the
Graduate Faculty in Mathematics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy,
The City University of New York

1978

This manuscript has been read and accepted for the Graduate Faculty in Mathematics in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

May 4, 1978
date

Alan J Hoffman
Alan J. Hoffman,
Chairman of Examining Committee

date

Alphonse Thomas Vasquez
Alphonse T. Vasquez,
Executive Officer

Supervisory Committee

The City University of New York

Abstract

"GREEDY" ALGORITHMS
FOR SOME
OPTIMIZATION PROBLEMS
ON A
LATTICE POLYHEDRON

by

Deborah Freedman Kornblum

Adviser: Dr. Alan J. Hoffman

Three "greedy" algorithms, which generalize results of Edmonds and Johnson, are presented to solve linear programming problems on certain "lattice polyhedra."

To My Favorite Physical Chemist,
Dearest Zvi,
"Yakarta Va'anee Ahavteecha"

ACKNOWLEDGMENTS

With heartfelt appreciation, I would like to acknowledge the assistance and guidance that was given to me by Dr. Alan J. Hoffman throughout the course of my graduate work. He is truly a warm, wonderful, and "peachy" person. He always made me feel that I could call upon him whenever I needed advice. It has been a privilege to have conducted this research under such an accomplished and energetic mathematician. I would also like to express my sincere appreciation to the City University of New York Graduate School, to Brooklyn College, and to their faculties and staffs for the educational opportunity that they provided. May the City University always strive to provide an excellent education for the inhabitants of this great city. I am deeply grateful to my husband, Zvi, without whose encouragement I would never have been able to overcome the many hurdles that I encountered during the course of my graduate studies. He is a constant source of support.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.a "Greediness"

Several "greedy" algorithms will be presented to solve linear programming problems on certain "lattice polyhedra" defined by Hoffman and Schwartz [12]. Since the definition of a "lattice polyhedron" is lengthy, it will be deferred to section 1.f, while we first discuss "greediness" and a history of some "greedy" algorithms.

By a "greedy" algorithm we mean the following: Suppose we have a set E with n elements and a nonnegative function w on E, and we wish to maximize (or minimize) $\sum_{e \in E} w(e)x(e)$ subject to various constraints on the variables $x(e)$. We arrange the n elements of E so that $w(e_1) \geqq w(e_2) \geqq \ldots \geqq w(e_n)$. We make $x(e_1)$ as large (or as small) as possible, subject to the constraints involving $x(e_1)$, while insuring feasibility of the remaining variables. We then do the same for $x(e_2)$, assuming $x(e_1)$ to be fixed at the assigned value, and so on, until we assign a value to $x(e_n)$.

1.b Kruskal's Algorithm

In order to look at the first algorithm entitled "greedy," we need to introduce several definitions. A graph consists of a set of nodes and a set of edges. An edge consists of two nodes, which are called its endpoints. A path between nodes a and b is a sequence of edges of the form (a,v_1), (v_1,v_2), ..., (v_{k-1}, v_k), (v_k,b), where a, v_1, v_2, ..., v_k,b are distinct nodes. A graph is said to be connected if for every two nodes a and b there is a path between a and b. A path from a to a is called a cycle. A graph which is connected and cycle-free is called a tree. A graph G' is a subgraph of a graph G if the nodes of G' are a subset of the nodes of G and the edges of G' are a subset of the edges of G. A subgraph that includes all the nodes of G and is a tree is called a spanning tree of G.

Suppose that we are given a connected graph G with n nodes and that each edge of G has an assigned nonnegative weight. The maximal spanning tree problem is to find a maximum weight spanning tree. Kruskal [14] proposed the following algorithm, to which Edmonds [3] first applied the term "greedy algorithm." Let w(e) represent the weight of edge e. Order the edges e_1, e_2, ..., e_n so that $w(e_1) \geqq w(e_2) \geqq \ldots \geqq w(e_n)$. Let T represent the set of edges in the solution. Put e_1 in T. Put e_2 in T, unless it forms a cycle with the edges already in T, and so on, until we have one less edge in T than the number of nodes in G. We stop then, because a cycle-free graph on $|V|$ nodes can have at most $|V| - 1$ edges [7].

Kruskal's algorithm is "greedy," as we have defined it, in the following way. Let $x(e_k) = 0$ if $e_k \notin T$ and $x(e_k) = 1$

if $e_k \in T$. We will say an edge (v_i, v_j) is contained in a set of nodes U if $\{v_i, v_j\} \subseteq U$. Since a cycle-free graph with $|U|$ nodes can have at most $|U| - 1$ edges, we must require for any subset U of the nodes V of G that the number of edges of T contained in U is at most $|U| - 1$. Thus, the maximal spanning tree problem can be expressed as

$$\max \sum_{e_k \in E} w(e_k)x(e_k)$$

subject to

$$(1.1) \qquad \sum_{e_k \text{ contained in } U} x(e_k) \leqq |U| - 1 \qquad \forall\, U \subseteq V$$

$$(1.2) \qquad x(e_k) = 0 \text{ or } 1 \qquad \forall\, e_k \in E.$$

Since $e_k = (v_i, v_j)$ is contained in $U = \{v_i, v_j\}$, (1.1) implies that $x(e_k)$ will be at most 1 and, clearly, always integral. Therefore, (1.2) can be replaced by $x(e_k) \geqq 0$ for all $e_k \in E$. Kruskal's algorithm consists of making $x(e_k)$ equal to 1 if no cycles are formed with $\{e_i | i = 1, \ldots, k-1, e_i \in T\}$, i.e., if none of the inequalities are violated. Otherwise, $x(e_k)$ will be zero.

1.c The "Greedy" Algorithm for Matroids

The "greedy" algorithm for matroids [3] is a generalization of Kruskal's algorithm. A <u>matroid</u> consists of a finite set E and a family $\mathcal{I}$ of subsets of E, called <u>independent</u> sets, with the properties

$$\text{(i)} \quad I \in \mathcal{I} \quad \text{and} \quad I' \subsetneq I \rightarrow I' \in \mathcal{I} ,$$

and

(ii) $\forall$ S $\subseteq$ E, all maximal independent sets contained in S have the same cardinality, which is called the <u>rank</u> of S.

A minimal set not in $\mathcal{I}$ is called a <u>circuit</u>. We can see that a graph is a matroid with E as the set of edges and $\mathcal{I}$ the family of cycle-free subsets of E. The generalization of the maximal spanning tree problem is to find the maximum weight independent set. Let w(e) represent the nonnegative weight of element e in E. The matroid "greedy" algorithm is the following. Order the elements of E so that $w(e_1) \geqq w(e_2) \geqq \ldots \geqq w(e_n)$. Let I represent the set of elements in the solution. For each k = 1,2, ..., n, put e_k in I, unless it forms a circuit with $\{e_i | i = 1,2, \ldots, k-1, e_i \varepsilon I\}$.

If we know the rank of S, r(S), for all S $\subsetneq$ E, the problem can be stated as

$$\max \sum_{e_k \varepsilon E} w(e_k)x(e_k)$$

subject to

$$\sum_{e_k \varepsilon S} x(e_k) \leqq r(S) \qquad \forall\ S \subseteq E$$

$$x(e_k) \geqq 0 \qquad \forall\ e_k \varepsilon E \quad .$$

Since $r(\{e_k\}) \leqq 1$ for $S = \{e_k\}$, by definition of the rank, it follows that $x(e_k)$ will equal at most 1. Thus, $x(e_k)$ will equal 1 if $e_k \varepsilon I$ and $x(e_k)$ will equal zero if $e_k \not\varepsilon I$. The algorithm consists of assigning a value of 1 to $x(e_k)$ for each $k = 1, 2, \ldots, n$, unless one of the inequalities is violated, in which case e_k is given a value of zero. Therefore, the algorithm is "greedy" as we have defined it.

We should note that the function $r(S)$ has certain properties:

(1.3) $\qquad S \subseteq T \rightarrow r(S) \leqq r(T) \quad ,$

(1.4) $\qquad r(S) \geqq 0 \qquad \forall\ S \subseteq E \quad ,$

and

(1.5) $\qquad r(S) + r(T) \geqq r(S \cup T) + r(S \cap T) \qquad \forall\ S, T \subseteq E \quad .$

If a function has property (1.5), it is called <u>submodular</u>. These properties can easily be proven [16].

1.d The Polymatroid "Greedy" Algorithm

The maximum independent set problem on a matroid was generalized to a maximization problem on a polymatroid [4]. Let E be a finite set and let r be any submodular nonnegative function on 2^E. Let $\bar{a} \in (R \cup \pm \infty)^E$. The polyhedron $\{\bar{x} = (x_e | e \in E) \in R^E | 0 \leqq \bar{x} \leqq \bar{a};\ \sum_{e \in S} x_e \leqq r(S) \quad \forall\ S \subseteq E\}$ is called a polymatroid. The linear program

$$\max \sum_{e \in E} c(e)x(e)$$

subject to

$$\sum_{e \in S} x(e) \leqq r(S) \qquad \forall\ S \subseteq E$$

$$0 \leqq x(e) \leqq a(e) \qquad \forall\ e \in E$$

and its dual

$$\min \sum_{S \subseteq E} r(S)y(S) + \sum_{e \in E} a(e)z(e)$$

subject to

$$\sum_{e|e\varepsilon S} y(S) + z(e) \geqq c(e) \qquad \forall\, e \,\varepsilon\, E$$

$$y(S) \geqq 0 \qquad \forall\, S \subseteq E$$

$$z(e) \geqq 0 \qquad \forall\, e \,\varepsilon\, E$$

can be solved by a "greedy" algorithm. Edmonds [4], [5] considered only the case where the vector $\bar{a}$ is all infinite (i.e., there are no upper bounds on the variables $x(e)$) and $r(S)$ satisfies property (1.3) in order to prove the "Greedy Algorithm Theorem," though since this algorithm is a special case of one of the algorithms presented here, we will see that $\bar{a}$ need not be infinite and that (1.3) need not be satisfied in order to solve the problem with a "greedy" algorithm. However, these requirements simplify Edmonds' algorithm, which we will now examine.

Greedy Algorithm Theorem [4], [5]

If the elements of E are arranged so that $c(e_1) \geqq c(e_2) \geqq \ldots \geqq c(e_n) \geqq 0 \geqq c(e_{n+1}) \geqq \ldots \geqq c(e_{|E|})$, then the following vectors $x^0 = (x^0(e_k)|k=1, \ldots, |E|)$ and $y^0 = (y^0(S)|S\subseteq E)$ are optimum solutions to the primal and dual problems, respectively: Let $S_k = \{e_1, e_2, \ldots, e_k\}$. Let $x^0(e_1) = r(S_1)$, $x^0(e_k) = r(S_k) - r(S_{k-1})$ for $k = 2, \ldots, n$; and $x^0(e_k) = 0$ for $k = n+1, \ldots, |E|$. Let $y^0(S_k) = c(e_k) - c(e_{k+1})$ for $k = 1, \ldots, n-1$; $y^0(S_n) = c(e_n)$, and $y^0(S) = 0$ for all other $S \subsetneq E$.

We will now see why this algorithm is "greedy" on the primal. We must show that for each k, $x(e_k)$ is made as large as possible, subject to the constraints. Since $x(e_1) \leqq r(S)$ must hold for all S such that $e_1 \varepsilon S$, we first let $x(e_1) = \min_{S | e_1 \varepsilon S} r(S)$. Since $\{e_1\} \subseteq S$ for all S such that $e_1 \varepsilon S$, $r(S_1) \leqq r(S)$, so that $x(e_1) = r(S_1)$. In general, we let

$$x(e_k) = \min_{S | e_k \varepsilon S} [r(S) - \sum_{\substack{e_\ell \varepsilon S \\ \ell<k}} x(e_\ell)] \quad .$$

Assume for k = 1, ..., p-1 that

$$(1.6) \qquad \min_{S | e_k \varepsilon S} [r(S) - \sum_{\substack{e_\ell \varepsilon S \\ \ell<k}} x(e_\ell)] = r(S_k) - \sum_{\ell=1}^{k-1} x(e_\ell) =$$

$$r(S_k) - r(S_{k-1}) \quad .$$

We would like to prove that (1.6) is true for k=p. For all $S \subsetneq E$ such that $e_p \varepsilon S$, let $S' = \{e_k | e_k \varepsilon S, k \leqq p\}$. $S' \subseteq S$, so that by (1.3) $r(s') \leqq r(S)$. Therefore,

$$(1.7) \qquad r(S_p) - r(S_{p-1}) \leqq r(S') - \sum_{\substack{e_\ell \varepsilon S' \\ \ell<p}} x(e_\ell) \rightarrow$$

$$r(S_p) - r(S_{p-1}) \leqq r(S) - \sum_{\substack{e_\ell \varepsilon S \\ \ell<p}} x(e_\ell) \quad .$$

By (1.5),

$$(1.8) \qquad r(S_p) + r(S' - \{e_p\}) \leqq r(S') + r(S_{p-1}) \quad ,$$

since $S' \cup S_{p-1} = S_p$ and $S' \cap S_{p-1} = S' - \{e_p\}$. (1.8) implies

$$(1.9) \qquad r(S_p) - r(S_{p-1}) \leqq r(S') - r(S' - \{e_p\}) \quad .$$

Now, since we are assuming that the values assigned to $x(e_1)$, $x(e_2)$, ..., $x(e_{p-1})$ satisfy the constraints, we have

$$\sum_{e_\ell \varepsilon S' - \{e_p\}} x(e_\ell) \leqq r\ (S' - \{e_p\}) \quad ,$$

which, with (1.9), implies

$$(1.10) \qquad r(S_p) - r(S_{p-1}) \leqq r(S') - \sum_{e_\ell \varepsilon S' - \{e_p\}} x(e_\ell) \quad .$$

By the induction hypothesis,

$$(1.11) \qquad r(S_{p-1}) = \sum_{\ell=1}^{p-2} x(e_\ell) + x(e_{p-1}) = \sum_{\ell=1}^{p-1} x(e_\ell) \quad .$$

(1.10) and (1.11) imply

$$(1.12) \qquad r(S_p) - r(S_{p-1}) = r(S_p) - \sum_{\ell=1}^{p-1} x(e_\ell) \leqq r(S') - \sum_{e_\ell \varepsilon S' - \{e_p\}} x(e_\ell) \quad .$$

From (1.12) and (1.7) we have

$$r(S_p) - \sum_{\ell=1}^{p-1} x(e_\ell) \leqq r(S) - \sum_{\substack{e_\ell \in S \\ \ell<p}} x(e_\ell) \quad ,$$

for all S such that $e_p \varepsilon S$. Therefore, (1.6) is true for $k=p$, so that the algorithm satisfies our definition of "greedy."

1.e Johnson's Algorithm

The algorithms we have discussed thus far are all "greedy" on their respective primal problems. We will now look at an algorithm which is "greedy" on the dual. Johnson [13] formulated the following problem and a "greedy" algorithm, which generalize the problem of finding the shortest path between two given nodes in a graph and Dijkstra's [2] algorithm to solve that problem.

Let a graph G have node set V and edge set E. Let G be directed, i.e., each edge e is an ordered pair (i,j) of nodes in V. We say e is out of i and into j. For a node i, let w(i) represent the set of edges out of i, and for a subset U of the nodes V, let w(U) be the edges (i,j) out of some node i in U such that $j \not\in U$. A source is a node such that no edge is directed into it. Let a node s in G be distinguished as the source. Let r be a nonnegative function on the subsets U of V. We assume r has the following properties:

(1.13) $\quad r(S) + r(T) \leqq r(S \cup T) + r(S \cap T),$

that is, r is <u>supermodular</u>, and

(1.14) $\quad r(V) = 0.$

(1.13) and (1.14) imply

(1.15) $\quad S \subsetneq T \rightarrow r(S) \geqq r(T) \quad .$

This can be seen if we consider the sets T and $V - (T-S)$. Then $r(T) \leqq r(T) + r\left(V-(T-S)\right) \leqq r\left(T \cup \left(V-(T-S)\right)\right) + r\left(T \cap \left(V-(T-S)\right)\right) = r(V) + r(S) = r(S) + 0$. The problem is

<u>primal</u> $\quad \min \sum_{e \in E} c(e)x(e)$

subject to

$$\sum_{e \in w(U)} x(e) \geqq r(U) \qquad \forall \quad U \subsetneq V \text{ with } s \in U$$

$$x(e) \geqq 0 \qquad \forall \quad e \in E \quad .$$

<u>dual</u> $\quad \max \sum_{U \subsetneq V} r(U)y(U)$

subject to

$$\sum_{U \mid e \varepsilon w(U)} y(U) \leqq c(e) \qquad \forall \quad e \varepsilon E$$

$$y(U) \geqq 0 \qquad \forall \quad U \subsetneq V \quad .$$

Johnson [13] proves that the following algorithm solves the problem:

Step 0. Let $U_0 = \{s\}$; let $c^0(e) = c(e) \quad \forall \quad e \varepsilon E$; let $k = 0$.

Step 1. Let $y(U_k) = \min_{e \mid e \varepsilon w(U_k)} \left[c^k(e) \right]$. Let e_k be the $e \varepsilon w(U_k)$ giving this minimum. Let $c^{k+1}(e) = c^k(e) - y(U_k)$ for all $e \varepsilon w(U_k)$; let $c^{k+1}(e) = c^k(e)$, otherwise. If $|U_k| = |V| - 1$, go to step 2. Otherwise, let $U_{k+1} = U_k \cup \{j\}$, where $e_k = (i,j)$, replace k with k+1, and repeat step 1.

Step 2. Let $x(e_k) = r(U_k) - \sum_{\substack{e_\ell \varepsilon w(U_k) \\ \ell \geqq k+1}} x(e_\ell)$. If $k = 0$, go to step 3. Otherwise, replace k with k-1 and repeat step 2.

Step 3. Let $y(U) = 0$ for all other $U \subsetneq V$ and $x(e) = 0$ for all other $e \varepsilon E$. Stop.

We would now like to show why this algorithm is "greedy" on the dual. Since we require that $\{s\} \subseteq U$ for all $U \subseteq V$, $r(\{s\}) \geqq r(U)$ for all $U \subseteq V$, by (1.15). Therefore, we would first make $y(\{s\}) = y(U_0)$ as large as possible, subject to the constraints. $y(U_0) \leqq c(e)$ must hold for all $e \varepsilon w(U_0)$, so that we can let $y(U_0) = \min\limits_{e|e\varepsilon w(U_0)} c(e)$. The e giving the minimum is e_1. Let $e_1 = (s,j)$. Since the algorithm is "greedy," we would now make $y(U)$ as large as possible for the U such that $r(U)$ is the greatest, after $r(U_0)$. However, if $e_1 \varepsilon w(U)$, $y(U)$ cannot be greater than zero. Therefore, $\{s,j\} \subseteq U$ must hold if $y(U)$ is to be greater than zero. Since $\{s,j\} \subseteq U$ implies $r(\{s,j\}) \geqq r(U)$, we next assign a value to $\{s,j\} = U_1$. In the same way, in general, we can make

$$y(U_k) = \min_{e|e\varepsilon w(U_k)} [c(e) - \sum_{\substack{e\varepsilon w(U_\ell) \\ \ell<k}} y(U_\ell)] =$$

$$\min_{e|e\varepsilon w(U_k)} c^k(e) \quad .$$

Thus, the algorithm is "greedy" on the dual. The values for the primal are assigned so that the duality equation [1] $\sum\limits_{e\varepsilon E} c(e)x(e) = \sum\limits_{U\subseteq V} r(U)y(U)$ is satisfied and feasibility is preserved to insure optimality of the solutions.

1.f Lattice Polyhedra

If r is an integral function, both Edmonds' and Johnson's algorithms will produce integral optimal primal solutions, so that by Lemma 2.1 of [8] the polymatroid and the polyhedron defined by Johnson's algorithm would be integer polyhedra. Hoffman and Schwartz [12] introduced a class of integer polyhedra, which they called "lattice polyhedra." All of the preceding problems are special cases of linear programming problems defined on these polyhedra, which we will now define.

We assume that the following are given: a partially ordered set L, two functions on $L \times L \rightarrow L$, "$\vee$" and "$\wedge$", that satisfy $a \vee b = b \vee a \geqq a,b$ and $a \wedge b = b \wedge a \leqq a,b$. We also assume $a<b$ implies $a \wedge b = a$ and $a \vee b = b$. A finite set U and a mapping $f: L \rightarrow 2^U - \phi$ that satisfies

$$(1.16) \qquad a < b < c \rightarrow f(a) \wedge f(c) \subset f(b)$$

are also assumed to be given. If f has the property

$$(1.17) \qquad u \in f(a), \quad a \leqq b \rightarrow u \in f(b) \quad ,$$

then f is called <u>upper</u>; if f has the property

$$(1.18) \qquad u \in f(a), \quad a \geqq b \rightarrow u \in f(b) \quad ,$$

then f is called lower. Each of (1.17) and (1.18) implies (1.16). If f satisfies (1.16),

(1.19) $$f(a) \cap f(b) \subset f(a \vee b) \cap f(a \wedge b) \quad ,$$

and

(1.20) $$f(a) \cup f(b) \subset f(a \vee b) \cup f(a \wedge b) \quad ,$$

f is called submodular. If f satisfies (1.16),

(1.21) $$f(a) \cap f(b) \supset f(a \vee b) \cap f(a \wedge b) \quad ,$$

and

(1.22) $$f(a) \cup f(b) \supset f(a \vee b) \cup f(a \wedge b) \quad ,$$

f is called supermodular. We also assume that a nonnegative function r is defined on L. r is said to be submodular if

(1.23) $$r(a \vee b) + r(a \wedge b) \leqq r(a) + r(b) \quad ,$$

and supermodular if

(1.24) $$r(a \vee b) + r(a \wedge b) \geqq r(a) + r(b) \quad .$$

Lastly, we assume we are given values $d(u) \geqq 0$ for all $u \in U$. Each of the following polyhedra is called a "lattice polyhedron":

$$\{\bar{x} \mid 0 \leqq \bar{x}(u) \leqq \bar{d}(u),\ \sum_{u \in f(a)} x(u) \leqq r(a),\ r(a) \text{ and } f(a) \text{ submodular}\}$$

and

$$\{\bar{x} \mid 0 \leqq \bar{x}(u) \leqq \bar{d}(u),\ \sum_{u \in f(a)} x(u) \geqq r(a),\ r(a) \text{ and } f(a) \text{ supermodular}\} \quad .$$

We consider the following two linear programs and their duals:

Submodular Problem

Let both r and f be submodular.

Primal $\quad \max \sum_{u \in U} c(u)x(u)$

subject to

(1.25) $$\sum_{u \in f(a)} x(u) \leqq r(a) \qquad \forall\ a \in L$$

(1.26) $$0 \leqq x(u) \leqq d(u) \qquad \forall\ u \in U \quad .$$

Dual $\min \sum_{a\varepsilon L} r(a)y(a) + \sum_{u\varepsilon U} d(u)z(u)$

subject to

(1.27) $\sum_{a|u\varepsilon f(a)} y(a) + z(u) \geqq c(u) \quad \forall\ u\ \varepsilon\ U$

(1.28) $y(a) \geqq 0 \quad \forall\ a\ \varepsilon\ L$

(1.29) $z(u) \geqq 0 \quad \forall\ u\ \varepsilon\ U$.

Supermodular Problem

Let both r and f be supermodular and let $c(u) \geqq 0$ for all $u\ \varepsilon\ U$.

Primal $\min \sum_{u\varepsilon U} c(u)x(u)$

subject to

(1.30) $\sum_{u|u\varepsilon f(a)} x(u) \geqq r(a) \quad \forall\ a\ \varepsilon\ L$

(1.31) $0 \leqq x(u) \leqq d(u) \quad \forall\ u\ \varepsilon\ U$.

Dual $\max \sum_{a\varepsilon L} r(a)y(a) - \sum_{u\varepsilon U} d(u)z(u)$

subject to

(1.32) $$\sum_{a | u \varepsilon f(a)} y(a) - z(u) \leqq c(u) \qquad \forall\, u \,\varepsilon\, U$$

(1.33) $$y(a) \geqq 0 \qquad \forall\, a \,\varepsilon\, L$$

(1.34) $$z(u) \geqq 0 \qquad \forall\, u \,\varepsilon\, U \;.$$

Hoffman and Schwartz [12], [10], [11] studied these linear programming problems in order to derive some known and some new combinatorial theorems. However, no "nice" algorithms existed to solve the problems. The purpose of this research is to present algorithms to solve these problems.

In Chapter 2 we present an algorithm that solves the submodular problem if f is either upper or lower. In Chapter 3 we present an algorithm that solves the supermodular problem if f is either upper or lower. In Chapter 4 we present an algorithm that solves the supermodular problem, without upper bounds, under the assumptions that r is monotonic, i.e., either

(1.35) $$a < b \rightarrow r(a) \leqq r(b)$$

or

(1.36) $$a > b \rightarrow r(a) \leqq r(b) \quad ,$$

and that L is a lattice.

Hoffman and Schwartz [12] proved that if r(a), c(u), and d(u) are integral, then there exists at least one integral optimal solution, which, by lemma 2.1 of [8] implies that the lattice polyhedra are integer polyhedra. The algorithms to be presented produce integral solutions if r(a), c(u), and d(u) are integral, thus providing an alternate proof. The algorithms in Chapters 2 and 3 are "greedy" on their respective primal problems, while the algorithm in Chapter 4 is "greedy" on its dual.

Several other algorithms which are "greedy" in approach can be found in references [6], [9], and [15]. However, we are still far from a comprehensive theory as to when and why the "greedy" approach will work.

CHAPTER 2

AN ALGORITHM FOR THE SUBMODULAR PROBLEM

We now present the algorithm for the submodular problem, in the case where f is either upper or lower. Since each of (1.17) and (1.18) clearly implies both (1.16) and (1.20), we need not assume (1.16) or (1.20).

Theorem 1. The following algorithm solves the submodular problem if f is either upper or lower:

Step 0. Order the elements of U so that $c(u_1) \geqq c(u_2) \geqq \ldots \geqq c(u_n) \geqq 0 \geqq c(u_{n+1}) \geqq \ldots \geqq c(u_{n+r})$.

Let $r^0(a) = r(a)$ for all a in L.

Let $k = 1$; let $j = 0$.

Step 1. If $u_k \varepsilon f(a_\ell)$ for some $\ell \leqq j$, let $x(u_k) = 0$ and go to (c). Otherwise:

(a) Let $x(u_k) = \min\left(\min_{a|u_k \varepsilon f(a)} r^{k-1}(a),\ d(u_k)\right)$. Increase j by 1. Let $k(j) = k$, where $k(j)$ represents the value of k corresponding to j.

(b) If $x(u_k) = \min_{a|u_k \varepsilon f(a)} r^{k-1}(a)$, let the a in L giving the minimum be a_j. If more than one a in L gives this minimum, then choose the maximum one in L to be a_j if f is upper, and the minimum one in L if f is lower. If $x(u_k) = d(u_k)$, no a is to be subscripted with j.

(c) Let $r^k(a) = r^{k-1}(a) - x(u_k)$ for all a such that $u_k \varepsilon f(a)$. Let $r^k(a) = r^{k-1}(a)$, otherwise. If k = n, let $x(u_k) = 0$ for $k \geqq n+1$, and go to step 2. Otherwise increase k by 1 and repeat step 1.

Step 2. Let m = j, i.e., m will represent the maximum value of j.

Let $y(a_m) = c(u_{k(m)})$, if m is the subscript of some a in L.

Let $z(u_{k(m)}) = c(u_{k(m)})$, if not.

Step 3. Decrease j by 1.

Let $y(a_j) = c(u_{k(j)}) - c(u_{k(j+1)}) + z(u_{k(j+1)})$, if j is the subscript of some a in L.

Let $z(u_{k(j)}) = c(u_{k(j)}) - c(u_{k(j+1)}) + z(u_{k(j+1)})$, if not.

If j = 1, go to step 4. Otherwise, repeat step 3.

Step 4. Let y(a) = 0 for all other a in L.

Let z(u) = 0 for all other u in U. Stop.

Proof: The method of proof will be to show that the solutions to the primal and dual problems are feasible, and then to show that they satisfy the complementary slackness optimality criterion [1], to prove optimality. If we reverse the order of L, a lower function becomes an upper one. Thus, it suffices to prove the theorem for an upper f. We assume, then, that f is upper.

We must first show that if more than one a in L gives a minimum value in step 1 of the algorithm, then there is a maximum such a. Let a_s and a_t be two elements in L such that $u_k \varepsilon f(a_s)$, $u_k \varepsilon f(a_t)$, and $r^{k-1}(a_s) = r^{k-1}(a_t) = \min_{a|u_k \varepsilon f(a)} r^{k-1}(a)$. It will be shown that $r^{k-1}(a_s \vee a_t) = r^{k-1}(a_s)$. If $r^{k-1}(a_s \vee a_t) < r^{k-1}(a_s)$, then $r^{k-1}(a_s)$ would not have been the minimum value. Suppose $r^{k-1}(a_s \vee a_t) > r^{k-1}(a_s)$. According to (2.5), which is proven below,

$$r(a) = r^{k-1}(a) + \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leq k-1}} x(u_\ell) \quad , \qquad \forall \; a \; \varepsilon \; L \; .$$

Thus, according to (1.23),

$$(2.1) \qquad r^{k-1}(a_s \vee a_t) + \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \leq k-1}} x(u_\ell) + r^{k-1}(a_s \wedge a_t) + \sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \leq k-1}} x(u_\ell) \leqq r^{k-1}(a_s) + \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \leq k-1}} x(u_\ell) +$$

$$r^{k-1}(a_t) + \sum_{\substack{u_\ell \in f(a_t) \\ \ell \leqq k-1}} x(u_\ell) \quad .$$

Now, by properties (1.19) and (1.17),

$$(2.2) \qquad \sum_{\substack{u_\ell \in f(a_s \vee a_t) \\ \ell \leqq k-1}} x(u_\ell) + \sum_{\substack{u_\ell \in f(a_s \wedge a_t) \\ \ell \leqq k-1}} x(u_\ell) \geqq$$

$$\sum_{\substack{u_\ell \in f(a_s)-f(a_t) \\ \ell \leqq k-1}} x(u_\ell) + \sum_{\substack{u_\ell \in f(a_t)-f(a_s) \\ \ell \leqq k-1}} x(u_\ell) +$$

$$2\sum_{\substack{u_\ell \in f(a_s) \cap f(a_t) \\ \ell \leqq k-1}} x(u_\ell) = \sum_{\substack{u_\ell \in f(a_s) \\ \ell \leqq k-1}} x(u_\ell) + \sum_{\substack{u_\ell \in f(a_t) \\ \ell \leqq k-1}} x(u_\ell) \quad .$$

Thus, (2.1) leads to

$$(2.3) \qquad r^{k-1}(a_s \vee a_t) + r^{k-1}(a_s \wedge a_t) \leqq r^{k-1}(a_s) + r^{k-1}(a_t) \quad .$$

If $r^{k-1}(a_s \vee a_t) > r^{k-1}(a_s)$ were true, then $r^{k-1}(a_s \wedge a_t) < r^{k-1}(a_t) = r^{k-1}(a_s)$ would hold. But since $u_k \in f(a_s \wedge a_t)$ by property (1.19), $a_s \wedge a_t$ would have given the minimum value, rather than a_s. Thus, $r^{k-1}(a_s \vee a_t) = r^{k-1}(a_s)$, so that there is a maximum a in L giving the minimum value $r^{k-1}(a_s)$.

Feasibility of the primal solution will now be proven.

In order to prove that $x(u_k) \geqq 0$ for all k, we will prove by induction on k that $r^k(a) \geqq 0$ for all $a \varepsilon L$. $r^0(a) \geqq 0$ for all $a \varepsilon L$, since $r^0(a) = r(a) \geqq 0$. Assume that $r^p(a) \geqq 0$ for all $a \varepsilon L$. Now, by step 1, $x(u_{p+1}) \leqq r^p(a)$ for all a such that $u_{p+1} \varepsilon f(a)$, and $r^{p+1}(a) = r^p(a) - x(u_{p+1})$ for all a such that $u_{p+1} \varepsilon f(a)$. Therefore, $r^{p+1}(a) \geqq 0$ for all a such that $u_{p+1} \varepsilon f(a)$. Also, $r^{p+1}(a) = r^p(a) \geqq 0$, otherwise. Now, either $x(u_k) = \min_{a|u_k \varepsilon f(a)} r^{k-1}(a) \geqq 0$, $x(u_k) = d(u_k) \geqq 0$, or $x(u_k) = 0$, so that $x(u_k) \geqq 0$ for all k. It is quite clear from step 1 that $x(u_k) \leqq d(u_k)$ for all k.

The proof that all the inequalities (1.25) hold will be by induction on k. Consider all a in L with the property that 1 is the largest value of k such that $u_k \varepsilon f(a)$. Then $\sum_{u \varepsilon f(a)} x(u) = x(u_1) \leqq r^0(a) = r(a)$ for all a with the above property, by definition of $x(u_1)$. Assume now for all a in L with the property that p is the largest value of k such that $u_k \varepsilon f(a)$ that $\sum_{u \varepsilon f(a)} x(u) \leqq r(a)$. Consider any $\hat{a}$ in L with the property that p+1 is the largest value of k such that $u_k \varepsilon f(\hat{a})$. Then

$$\sum_{u \varepsilon f(\hat{a})} x(u) = x(u_{p+1}) + \sum_{\substack{u_k \varepsilon f(\hat{a}) \\ k \leqq p}} x(u_k) \leqq$$

$$\sum_{\substack{u_k \varepsilon f(\hat{a}) \\ k \leqq p}} x(u_k) + \min_{a|u_{p+1} \varepsilon f(a)} r^p(a) \leqq \sum_{\substack{u_k \varepsilon f(\hat{a}) \\ k \leqq p}} x(u_k) +$$

$$r^p(a)$$

for all a such that $u_{p+1} \varepsilon f(a)$. Therefore, in particular,

$$(2.4) \qquad \sum_{u \varepsilon f(\hat{a})} x(u_k) \leqq \sum_{\substack{u_k \varepsilon f(\hat{a}) \\ k \leqq p}} x(u_k) + r^p(\hat{a}) \quad .$$

To complete the proof, it will be shown that

$$(2.5) \qquad r^k(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq k}} x(u_\ell), \qquad \forall \quad a \varepsilon L \quad .$$

By definition of $r^0(a)$, $r^0(a) = r(a)$ for all $a \varepsilon L$.

$$r^1(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq 1}} x(u_\ell)$$

is true for all a, since if $u_1 \not\varepsilon f(a)$, then $r^1(a) = r^0(a) = r(a)$, and if $u_1 \varepsilon f(a)$, then $r^1(a) = r^0(a) - x(u_1) = r(a) - x(u_1)$. Assume that

$$r^p(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq p}} x(u_\ell) \quad , \qquad \forall \quad a.$$

$r^{p+1}(a) = r^p(a) - x(u_{p+1})$, if $u_{p+1} \varepsilon f(a)$, and $r^{p+1}(a) = r^p(a)$, otherwise. Therefore,

$$r^{p+1}(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq p+1}} x(u_\ell) \quad ,$$

which completes the proof of (2.5). It follows from (2.4) and (2.5) that

$$\sum_{u \varepsilon f(\hat{a})} x(u) \leqq \sum_{\substack{u_k \varepsilon f(\hat{a}) \\ k \leqq p}} x(u_k) + r(\hat{a}) - \sum_{\substack{u_k \varepsilon f(\hat{a}) \\ k \leqq p}} x(u_k)$$

$$= r(\hat{a}) \quad ,$$

which concludes the proof of feasibility of the solution to the primal.

To prove feasibility of the solution to the dual, the following lemma must be proven:

(2.6) <u>Lemma</u>. $s < t \rightarrow a_s < a_t$.

<u>Proof</u>: Let s be the subscript of some a in L, and let t be the next subscript after s. Suppose $a_t \leqq a_s$. Then since f is upper, if $u_k \varepsilon f(a_t)$, $u_k \varepsilon f(a_s)$. Then there would be no $u_k \varepsilon f(a_t)$ such that $u_k \not\varepsilon f(a_s)$, so that a_t could not have been subscripted in step 1 of the algorithm. Therefore $a_t \not\leqq a_s$. We must show that a_s and a_t are not incomparable. Suppose they were. There are two cases to consider:

Case I. $u_{k(s)} \varepsilon f(a_t)$.

If $u_{k(s)} \varepsilon f(a_t)$, by property (1.19), $u_{k(s)} \varepsilon f(a_s \wedge a_t)$, since $u_{k(s)} \varepsilon f(a_s)$. Therefore, according to step 1 of the algorithm,

$$(2.7) \qquad r^{k(s)-1}(a_s) \leqq r^{k(s)-1}(a_s \wedge a_t) \quad .$$

As a result of step 1 of the algortihm,

$$(2.8) \qquad r^{k(t)-1}(a_t) = r^{k(s)}(a_t) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_t)}} d(u_{k(j)}) \quad ,$$

since for j, $s<j<t$, $x(u_{k(j)}) = d(u_{k(j)})$, and for all ℓ such that $\ell \neq k(j)$ for some j and $k(s)<\ell<k(t)$, $x(u_\ell) = 0$. Now, since $u_{k(s)} \varepsilon f(a_t)$,

$$(2.9) \qquad r^{k(s)}(a_t) = r^{k(s)-1}(a_t) - x(u_{k(s)}) =$$

$$= r^{k(s)-1}(a_t) - r^{k(s)-1}(a_s) \quad .$$

(2.8) and (2.9) imply that

$$(2.10) \qquad r^{k(t)-1}(a_t) = r^{k(s)-1}(a_t) - r^{k(s)-1}(a_s) -$$

$$\sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_t)}} d(u_{k(j)}) \quad .$$

Similarly,

$$(2.11) \qquad r^{k(t)-1}(a_s \vee a_t) = r^{k(s)}(a_s \vee a_t) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_s \vee a_t)}} d(u_{k(j)}) = r^{k(s)-1}(a_s \vee a_t) -$$

$$x(u_{k(s)}) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_s \vee a_t)}} d(u_{k(j)}) =$$

$$r^{k(s)-1}(a_s \vee a_t) - r^{k(s)-1}(a_s) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_s \vee a_t)}} d(u_{k(j)}) \ .$$

By (2.3), $r^{k(s)-1}(a_s \vee a_t) + r^{k(s)-1}(a_s \wedge a_t) \leqq r^{k(s)-1}(a_s) + r^{k(s)-1}(a_t)$, which implies

$$(2.12) \qquad r^{k(s)-1}(a_s \vee a_t) - r^{k(s)-1}(a_s) \leqq r^{k(s)-1}(a_t) - r^{k(s)-1}(a_s \wedge a_t) \ .$$

(2.7) and (2.12) imply

$$(2.13)\qquad r^{k(s)-1}(a_s \vee a_t) - r^{k(s)-1}(a_s) \leqq r^{k(s)-1}(a_t) - r^{k(s)-1}(a_s) \quad .$$

By property (1.17),

$$(2.14)\qquad \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_s \vee a_t)}} d(u_{k(j)}) \geqq \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_t)}} d(u_{k(j)}) \quad .$$

From (2.13) and (2.14) we can conclude

$$r^{k(s)-1}(a_s \vee a_t) - r^{k(s)-1}(a_s) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_s \vee a_t)}} d(u_{k(j)}) \leqq r^{k(s)-1}(a_t) - r^{k(s)-1}(a_s) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_t)}} d(u_{k(j)}) \quad ,$$

which implies that $r^{k(t)-1}(a_s \vee a_t) \leqq r^{k(t)-1}(a_t)$, by (2.10) and (2.11). Then $a_s \vee a_t$ would have been subscripted instead of a_t in step 1 of the algorithm. Thus, $a_s < a_t$ must be true.

Case II. $u_{k(s)} \notin f(a_t)$.

As in Case I, (2.8) is true. But, since $u_{k(s)} \notin f(a_t)$, $r^{k(s)}(a_t) = r^{k(s)-1}(a_t)$, so that

$$(2.15) \qquad r^{k(t)-1}(a_t) = r^{k(s)-1}(a_t) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_t)}} d(u_{k(j)}) \quad .$$

(2.11) is true in this case, also. Since $r^{k(s)-1}(a_s \wedge a_t) > 0$, (2.12) implies

$$(2.16) \qquad r^{k(s)-1}(a_s \vee a_t) - r^{k(s)-1}(a_s) \leqq r^{k(s)-1}(a_t) \quad .$$

From (2.16) and (2.14) we have

$$r^{k(s)-1}(a_s \vee a_t) - r^{k(s)-1}(a_s) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_s \vee a_t)}} d(u_{k(j)}) \leqq r^{k(s)-1}(a_t) - \sum_{\substack{s<j<t \\ u_{k(j)} \varepsilon f(a_t)}} d(u_{k(j)}) \quad ,$$

which implies that $r^{k(t)-1}(a_s \vee a_t) \leqq r^{k(t)-1}(a_t)$, by (2.15) and (2.11). Then $a_s \vee a_t$ would have been subscripted instead of a_t in step 1 of the algorithm. Thus $a_s < a_t$ must be true. By transitivity of the partial ordering of L, it follows that $a_s < a_t$ for any s and t such that $s < t$, so that the lemma is proven.

The feasibility of the dual solution will now be proven. $z(u_\ell) = 0$ unless $\ell = k(j)$ for some j and j is not the subscript of any a in L. We must prove that in the latter case $z(u_{k(j)}) \geqq 0$. If $j=m$, $z(u_{k(m)}) = c(u_{k(m)}) \geqq 0$, since $k(m) \leqq n$. Assume for all $j \geqq p+1$ that $z(u_{k(j)}) \geqq 0$. For $j=p$, if p is not the subscript of any a in L, $z(u_{k(p)}) = c(u_{k(p)}) - c(u_{k(p+1)}) + z(u_{k(p+1)}) \geqq c(u_{k(p)}) - c(u_{k(p+1)})$, since, by the induction hypothesis, $z(u_{k(p+1)}) \geqq 0$. $c(u_{k(p)}) - c(u_{k(p+1)}) \geqq 0$, by step 0 of the algorithm. If p is the subscript of some a, let $p-r$ be the greatest $j<p$ such that j is not the subscript of some a. Then $z(u_{k(p-r)}) = c(u_{k(p-r)}) - c(u_{k(p-r+1)}) + z(u_{k(p-r+1)}) = c(u_{k(p-r)}) - c(u_{k(p-r+1)})$, since $z(u_{k(p-r+1)}) = 0$. $c(u_{k(p-r)}) - c(u_{k(p-r+1)} \geqq 0$ by step 0, so that $z(u) \geqq 0$ for all $u \varepsilon U$.

Since $y(a) = 0$ for all non-subscripted elements of L, we must show that for all j, if j is the subscript of some a in L, then $y(a_j) \geqq 0$. This is an immediate result of steps 2 and 3 of the algorithm and (1.29), which was just proven. If m, the maximum value of j, is the subscript of some a, then $y(a_m) = c(u_{k(m)}) \geqq 0$. For any other value of j that is the subscript of some a, $y(a_j) = c(u_{k(j)}) - c(u_{k(j+1)}) + z(u_{k(j+1)}) \geqq c(u_{k(j)}) - c(u_{k(j+1)}) \geqq 0$, using (1.29) and step 0 of the algorithm.

Before proving that the inequalities (1.27) hold, we will prove

$$\sum_{\substack{j \geqq \ell \\ a_j | u_{k(\ell)} \varepsilon f(a_j)}} y(a_j) + z(u_{k(\ell)}) = c(u_{k(\ell)}), \tag{2.17}$$

$$\ell = 1, \ldots, m,$$

which will be used in the proof that (1.27) holds, as well as in the proof of optimality. When ℓ=m, if m is the subscript of some a in L, $y(a_m) = c(u_{k(m)})$ and $z(u_{k(m)}) = 0$, so that (2.17) is true. If m is not the subscript of some a, $z(u_{k(m)}) = c(u_{k(m)})$, so that (2.17) is true. Assume that (2.17) is true for $\ell \geqq p+1$. For ℓ=p, by step 3 of the algorithm, either

$$y(a_p) = c(u_{k(p)}) - c(u_{k(p+1)} + z(u_{k(p+1)}) \quad , \tag{2.18}$$

if p is the subscript of some a, or

$$z(u_{k(p)}) = c(u_{k(p)}) - c(u_{k(p+1)}) + z(u_{k(p+1)}) \quad , \tag{2.19}$$

if not. Now, by the induction hypothesis,

$$c(u_{k(p)}) - c(u_{k(p+1)}) + z(u_{k(p+1)}) = c(u_{k(p)}) -$$

$$\sum_{\substack{j \geq p+1 \\ a_j | u_{k(p+1)} \varepsilon f(a_j)}} y(a_j) - z(u_{k(p+1)}) + z(u_{k(p+1)}) =$$

$$c(u_{k(p)}) - \sum_{\substack{j \geq p+1 \\ a_j | u_{k(p+1)} \varepsilon f(a_j)}} y(a_j) = c(u_{k(p)}) -$$

$$\sum_{\substack{j \geq p+1 \\ a_j | u_{k(p)} \varepsilon f(a_j)}} y(a_j) \quad ,$$

by (2.6) and (1.17). Thus, if p is the subscript of some a, (2.18) results in

$$\sum_{\substack{j \geq p \\ a_j | u_{k(p)} \varepsilon f(a_j)}} y(a_j) = c(u_{k(p)}) \quad ,$$

which implies

$$\sum_{\substack{j \geq p \\ a_j | u_{k(p)} \varepsilon f(a_j)}} y(a_j) + z(u_{k(p)}) = c(u_{k(p)} \quad ,$$

since $z(u_{k(p)}) = 0$ would hold. If p is not the subscript of any a, (2.19) results in

$$\sum_{\substack{j \geqq p+1 \\ a_j | u_{k(p)} \varepsilon f(a_j)}} y(a_j) + z(u_{k(p)}) = c(u_{k(p)}) \quad ,$$

which implies

$$\sum_{\substack{j \geqq p \\ a_j | u_{k(p)} \varepsilon f(a_j)}} y(a_j) + z(u_{k(p)}) = c(u_{k(p)}) \quad ,$$

since letting j=p does not add a term to the sum. This concludes the proof of (2.17).

The proof that the inequalities (1.27) hold is also by induction on j. For j=m, consider all u_k such that $n \geqq k > k(m)$. $u_k \varepsilon f(a_\ell)$ for some least ℓ, since m is the maximum value of j, which implies that $x(u_k)$ could not be made positive in step 1 of the algorithm. Since $\ell \leqq m$, it follows from (2.6) and (1.17) that $u_k \varepsilon f(a_j)$ must hold for any j that is the subscript of some a such that $\ell \leqq j \leqq m$. Let $m-r$ be the greatest j that is the subscript of some a in L. Then

$$\sum_{a | u_k \varepsilon f(a)} y(a) + z(u_k) \geqq y(a_{m-r}) \quad .$$

Now, using (2.17), we have

$$y(a_{m-r}) = c(u_{k(m-r)}) - c(u_{k(m-r+1)}) + z(u_{k(m-r+1)}) =$$

$$c(u_{k(m-r)}) - \sum_{\substack{j \geq m-r+1 \\ a_j | u_{k(m-r+1)} \varepsilon f(a_j)}} y(a_j) -$$

$$z(u_{k(m-r+1)}) + z(u_{k(m-r+1)}) = c(u_{k(m-r)}) -$$

$$\sum_{\substack{j \geq m-r+1 \\ a_j | u_{k(m-r+1)} \varepsilon f(a_j)}} y(a_j) = c(u_{k(m-r)}) \quad ,$$

since

$$\sum_{\substack{j \geq m-r+1 \\ a_j | u_{k(m-r+1)} \varepsilon f(a_j)}} y(a_j) = 0 \quad .$$

By step 0, $c(u_{k(m-r)}) \geqq c(u_k)$, so that

$$\sum_{a | u_k \varepsilon f(a)} y(a) + z(u_k) \geqq y(a_{m-r}) \geqq c(u_k) \quad .$$

Thus, (1.27) is true for all k such that $n \geqq k > k(m)$. For $k = k(m)$,

$$\sum_{a | u_{k(m)} \varepsilon f(a)} y(a) + z(u_{k(m)}) \geq \sum_{\substack{j \geq m \\ a_j | u_{k(m)} \varepsilon f(a_j)}} y(a_j) +$$

$$z(u_{k(m)}) = c(u_{k(m)}) \quad ,$$

by (2.17).

Now assume for all $k \geq k(p+1)$ that

$$\sum_{a | u_k \varepsilon f(a)} y(a) + z(u_k) \geq c(u_k) \quad .$$

Let $u_{k'}$ be an element of U such that $k' > k(p)$ but $k' < k(p+1)$. $u_{k'} \varepsilon f(a_\ell)$ for some least $\ell \leq p$, since otherwise $u_{k'}$ would be $u_{k(p+1)}$. Therefore, as a result of (2.6) and (1.17),

$$\text{(2.20)} \qquad \sum_{a | u_{k'} \varepsilon f(a)} y(a) + z(u_{k'}) \geq \sum_{\substack{j \geq \ell \\ a_j | u_{k'} \varepsilon f(a_j)}} y(a_j) +$$

$$z(u_{k'}) = \sum_{\substack{j \geq \ell \\ a_j | u_{k(\ell)} \varepsilon f(a_j)}} y(a_j) + z(u_{k'}) = c(u_{k(\ell)}) -$$

$$z(u_{k(\ell)}) + z(u_{k'}) \quad ,$$

by (2.17). But $z(u_{k(\ell)}) = 0$ and $z(u_{k'}) = 0$ by steps 3 and

4. Thus (2.20) implies

$$\sum_{a|u_{k'}\varepsilon f(a)} y(a) + z(u_{k'}) \geqq c(u_{k(\ell)}) \geqq c(u_{k(p)}) \geqq$$

$$c(u_{k'}) \quad ,$$

so that (1.27) is true for $k' > k(p)$. If $k' = k(p)$, by (2.17),

$$\sum_{a|u_{k(p)}\varepsilon f(a)} y(a) + z(u_{k(p)}) \geqq \sum_{\substack{j \geqq p \\ a_j|u_{k(p)}\varepsilon f(a_j)}} y(a_j) +$$

$$z(u_{k(p)}) = c(u_{k(p)}) \quad .$$

Thus (1.27) is true for all u_k such that $1 \leqq k \leqq n$. For u_k such that $k \geqq n+1$, $c(u_k) \leqq 0$, according to step 0, so that the inequalities (1.27) clearly hold. Therefore, (1.27) holds for all $u \varepsilon U$. This completes the proof of feasibility of the dual solution.

In order to prove optimality of the solutions, the complementary slackness optimality criterion [1] will be shown to be satisfied. The complementary slackness optimality criterion of linear programming states that if the solutions to the primal and dual solutions are feasible, then they are optimal if

$$(2.21)\qquad x(u_k) > 0 \rightarrow \sum_{a_j | u_k \varepsilon f(a_j)} y(a_j) + z(u_k) = c(u_k) \quad ,$$

$$(2.22)\qquad y(a) > 0 \rightarrow \sum_{u \varepsilon f(a)} x(u) = r(a) \quad ,$$

and

$$(2.23)\qquad z(u) > 0 \rightarrow x(u) = d(u) \quad .$$

$x(u_k) > 0$ only for those k that are equal to k(j) for some j. Now,

$$(2.24)\qquad \sum_{a_j | u_{k(\ell)} \varepsilon f(a_j)} y(a_j) = \sum_{\substack{j \geq \ell \\ a_j | u_{k(\ell)} \varepsilon f(a_j)}} y(a_j) \quad ,$$

since if $u_{k(\ell)} \varepsilon f(a_j)$, $j < \ell$, then $x(u_{k(\ell)})$ could not have been made positive in step 1. (2.17) and (2.24) imply that (2.21) holds.

According to step 1 of the algorithm, if j is the subscript of some a in L, $r^{k(j)-1}(a_j) = x(u_{k(j)})$, so that using (2.5) we have

$$x(u_{k(j)}) = r(a_j) - \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \leq k(j)-1}} x(u_\ell) \quad ,$$

which implies

$$(2.25) \qquad r(a_j) = \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \leq k(j)}} x(u_\ell) \quad .$$

For $\ell > k(j)$, if $u_\ell \varepsilon f(a_j)$, then $x(u_\ell) = 0$, so that (2.25) implies

$$r(a_j) = \sum_{u_\ell \varepsilon f(a_j)} x(u_\ell) \quad .$$

For any $a \neq a_j$ for some j, $y(a) = 0$. Therefore,

$$y(a_j) > 0 \rightarrow r(a_j) = \sum_{u_\ell \varepsilon f(a_j)} x(u_\ell) \quad ,$$

proving that (2.22) holds.

$z(u) > 0$ only if $u = u_{k(j)}$ for some j and j is not the subscript of any a in L. These conditions occur only when $x(u_{k(j)}) = d(u_{k(j)})$ in step 1 of the algortihm, so that (2.23) is true.

This completes the proof that the solutions are optimal. Since U is a finite set, the algorithm will terminate.

CHAPTER 3

AN ALGORITHM FOR THE SUPERMODULAR PROBLEM

In this chapter, we present the algorithm for the supermodular problem, in the case where f is either upper or lower. Since each of (1.17) and (1.18) clearly implies both (1.16) and (1.21), we need not assume (1.16) or (1.21).

Theorem 2. The following algorithm solves the supermodular problem if f is either upper or lower:

Step 0. Order the elements of U so that $c(u_1) \geqq c(u_2) \geqq \ldots \geqq c(u_n) \geqq 0$. Let $r^0(a) = r(a)$ for all $a \in L$. Let $k=1$; let $i=0$; let $j=0$.

Step 1. If $u_k \in f(a_\ell)$ for some $\ell \leqq j$, let $x(u_k) = d(u_k)$, increase i by 1, let $k(i) = k$, where $k(i)$ represents the value of k corresponding to i, and go to step 2. Otherwise:

(a) Let $x(u_k) = \max \left\{ \max\limits_{a | u_k \in f(a)} [r^{k-1}(a) - \sum\limits_{\substack{u_\ell \in f(a) \\ \ell \geqq k+1}} d(u_\ell)], 0 \right\}$. If $x(u_k) > d(u_k)$, the problem is infeasible; stop.

(b) If $0 < x(u_k) < d(u_k)$, increase j by 1. Let the a in L giving the maximum value be a_j. If more than one a in L gives this maximum value, choose the maximum such a in L to be a_j if f is upper, and the minimum one in L if f is lower. Let $\bar{k}(j) = k$, where $\bar{k}(j)$ represents the value of k corresponding to j.

Step 2. Let $r^k(a) = r^{k-1}(a) - x(u_k)$ for all a such that $u_k \varepsilon f(a)$; let $r^k(a) = r^{k-1}(a)$, otherwise. If k=n, go to step 3. Otherwise, increase k by 1 and return to step 1.

Step 3. Let m=j, i.e., m will represent the maximum value of j. Let $y(a_m) = c(u_{\bar{k}(m)})$.

Step 4. Decrease j by 1. Let $y(a_j) = c(u_{\bar{k}(j)}) - c(u_{\bar{k}(j+1)})$. If j=1, go to step 5. Otherwise, repeat step 4.

Step 5. Let $z(u_{k(i)}) = \sum_{a_j | u_{k(i)} \varepsilon f(a_j)} y(a_j) - c(u_{k(i)})$.

If i=1, go to step 6. Otherwise, decrease i by 1 and repeat step 5.

Step 6. Let $y(a) = 0$ for all other $a \varepsilon L$. Let $z(u) = 0$ for all other $u \in U$. Stop.

Proof: The method of proof will be to show that the solutions to the primal and dual problems are feasible, and then to show that they satisfy the complementary slackness optimality criterion [1], to prove optimality. If we reverse the order of L, a lower function becomes an upper one. Thus, it suffices to prove the theorem for an upper f. We assume, then, that f is upper.

We must first show that if more than one $a \in L$ gives a maximum value in step 1(a) of the algorithm, then there is a maximum such a. Let a_s and a_t be two elements in L such that $u_k \in f(a_s)$, $u_k \in f(a_t)$, and

$$r^{k-1}(a_s) - \sum_{\substack{u_\ell \in f(a_s) \\ \ell \geq k+1}} d(u_\ell) = r^{k-1}(a_t) - \sum_{\substack{u_\ell \in f(a_t) \\ \ell \geq k+1}} d(u_\ell) =$$

$$\max_{a \mid u_k \in f(a)} \left[r^{k-1}(a) - \sum_{\substack{u_\ell \in f(a) \\ \ell \geq k+1}} d(u_\ell) \right] .$$

Since f is upper, $u_k \in f(a_s \vee a_t)$. It will be shown that

$$r^{k-1}(a_s \vee a_t) - \sum_{\substack{u_\ell \in f(a_s \vee a_t) \\ \ell \geq k+1}} d(u_\ell) = r^{k-1}(a_s) -$$

$$\sum_{\substack{u_\ell \in f(a_s) \\ \ell \geq k+1}} d(u_\ell) ,$$

assuming the problem is feasible. If

$$r^{k-1}(a_s \vee a_t) - \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \geq k+1}} d(u_\ell) > r^{k-1}(a_s) - \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geq k+1}} d(u_\ell) \quad ,$$

then clearly $a_s \vee a_t$ would have given the maximum value in step 1 of the algorithm. Suppose

$$r^{k-1}(a_s \vee a_t) - \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \geq k+1}} < r^{k-1}(a_s) - \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geq k+1}} d(u_\ell) \quad .$$

According to (3.10), which is proven below,

$$r(a) = r^{k-1}(a) + \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leq k-1}} x(u_\ell) \quad , \qquad \forall \;\; a \varepsilon L.$$

Thus, according to (1.24),

$$(3.1) \qquad r^{k-1}(a_s \vee a_t) + \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \leqq k-1}} x(u_\ell) + r^{k-1}(a_s \wedge a_t) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \leqq k-1}} x(u_\ell) \geqq r^{k-1}(a_s) + \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \leqq k-1}} x(u_\ell) +$$

$$r^{k-1}(a_t) + \sum_{\substack{u_\ell \varepsilon f(a_t) \\ \ell \leqq k-1}} x(u_\ell) \quad .$$

Now, by properties (1.22) and (1.17),

$$(3.2) \qquad \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \leqq k-1}} x(u_\ell) + \sum_{\substack{u_\ell \varepsilon f(a_t) \\ \ell \leqq k-1}} x(u_\ell) = \sum_{\substack{u_\ell \varepsilon f(a_s) \cup f(a_t) \\ \ell \leqq k-1}} x(u_\ell) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s) \cap f(a_t) \\ \ell \leqq k-1}} x(u_\ell) \geqq \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \cup f(a_s \wedge a_t) \\ \ell \leqq k-1}} x(u_\ell) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \cap f(a_s \wedge a_t) \\ \ell \leqq k-1}} x(u_\ell) = \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \leqq k-1}} x(u_\ell) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \leqq k-1}} x(u_\ell) \quad .$$

Thus, (3.1) leads to

$$(3.3) \qquad r^{k-1}(a_s \vee a_t) + r^{k-1}(a_s \wedge a_t) \geqq r^{k-1}(a_s) + r^{k-1}(a_t) \quad .$$

By properties (1.22) and (1.17),

$$(3.4) \qquad \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \geqq k+1}} d(u_\ell) + \sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \geqq k+1}} d(u_\ell) =$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \cup f(a_s \wedge a_t) \\ \ell \geqq k+1}} d(u_\ell) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \cap f(a_s \wedge a_t) \\ \ell \geqq k+1}} d(u_\ell) \leqq \sum_{\substack{u_\ell \varepsilon f(a_s) \cap f(a_t) \\ \ell \geqq k+1}} d(u_\ell) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s) \cup f(a_t) \\ \ell \geqq k+1}} d(u_\ell) = \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geqq k+1}} d(u_\ell) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_t) \\ \ell \geqq k+1}} d(u_\ell) \quad .$$

(3.3) and (3.4) imply

$$(3.5) \qquad r^{k-1}(a_s \vee a_t) - \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \geqq k+1}} d(u_\ell) + r^{k-1}(a_s \wedge a_t) -$$

$$\sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \geqq k+1}} d(u_\ell) \geqq r^{k-1}(a_s) - \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geqq k+1}} d(u_\ell) +$$

$$r^{k-1}(a_t) - \sum_{\substack{u_\ell \varepsilon f(a_t) \\ \ell \geqq k+1}} d(u_\ell) \quad .$$

If

$$r^{k-1}(a_s \vee a_t) - \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \geqq k+1}} d(u_\ell) < r^{k-1}(a_s) - \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geqq k+1}} d(u_\ell) \quad ,$$

then

$$r^{k-1}(a_s \wedge a_t) - \sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \geqq k+1}} d(u_\ell) > r^{k-1}(a_t) - \sum_{\substack{u_\ell \varepsilon f(a_t) \\ \ell \geqq k+1}} d(u_\ell) \tag{3.6}$$

must hold. Now

$$r^{k-1}(a_t) - \sum_{\substack{u_\ell \varepsilon f(a_t) \\ \ell \geqq k+1}} d(u_\ell) > 0 \quad , \tag{3.7}$$

since otherwise $x(u_k)$ would be given a value of zero in step 1 of the algorithm, and no $a \varepsilon L$ would be subscripted. (3.6) and (3.7) imply

$$r^{k-1}(a_s \wedge a_t) > \sum_{\substack{u_\ell \varepsilon f(a_s \wedge a_t) \\ \ell \geqq k+1}} d(u_\ell) \quad . \tag{3.8}$$

Then if $u_k \not\varepsilon f(a_s \wedge a_t)$, the problem is infeasible. Therefore, if the problem is feasible, $u_k \varepsilon f(a_s \wedge a_t)$. Then (3.6) would imply that a_t does not give the maximum value in step 1 of the algorithm. Therefore,

$$r^{k-1}(a_s \vee a_t) - \sum_{\substack{u_\ell \varepsilon f(a_s \vee a_t) \\ \ell \geqq k+1}} d(u_\ell) = r^{k-1}(a_s) - \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geqq k+1}} d(u_\ell) \quad ,$$

so that there is a maximum $a \varepsilon L$ giving the maximum value

$$r^{k-1}(a_s) - \sum_{\substack{u_\ell \varepsilon f(a_s) \\ \ell \geqq k+1}} d(u_\ell) \quad .$$

Feasibility of the primal solution will now be proven. $x(u_k) \geqq 0$ for all $u_k \varepsilon U$ is an immediate result of step 1 of the algorithm. $x(u_k) \leqq d(u_k)$ for all $u_k \varepsilon U$ is also an immediate result of step 1.

The proof that all the inequalities (1.30) hold is by induction. Consider all a in L with the property that 1 is the largest value of k such that $u_k \varepsilon f(a)$. Then

$$\sum_{u | u \varepsilon f(a)} x(u) = x(u_1) \geqq r^0(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \geqq 2}} d(u_\ell) =$$

$$r^0(a) = r(a) \quad ,$$

for all a with the above property, by definition of $x(u_1)$. Assume now for all $a \in L$ with the property that p is the largest value of k such that $u_k \in f(a)$ that $\sum_{u|u\in f(a)} x(u) \geqq r(a)$. Consider any $a' \in L$ with the property that p+1 is the largest value of k such that $u_k \in f(a')$. Then

$$\sum_{u|u\in f(a')} x(u) = x(u_{p+1}) + \sum_{\substack{u_k \in f(a') \\ k \leqq p}} x(u_k) \geqq$$

$$\sum_{\substack{u_k \in f(a') \\ k \leqq p}} x(u_k) + \max_{a|u_{p+1}\in f(a)} \left[r^p(a) - \sum_{\substack{u_\ell \in f(a) \\ \ell \geqq p+2}} d(u_\ell) \right] \geqq$$

$$\sum_{\substack{u_k \in f(a') \\ k \leqq p}} x(u_k) + r^p(a) - \sum_{\substack{u_\ell \in f(a) \\ \ell \geqq p+2}} d(u_\ell) \quad ,$$

for all a such that $u_{p+1} \in f(a)$. Therefore, in particular,

$$(3.9) \qquad \sum_{u\in f(a')} x(u) \geqq \sum_{\substack{u_k \in f(a') \\ k \leqq p}} x(u_k) + r^p(a') - \sum_{\substack{u_\ell \in f(a') \\ \ell \geqq p+2}} d(u_\ell) \quad .$$

To complete the proof, we will show that

$$(3.10) \qquad r^k(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq k}} x(u_\ell), \qquad \forall\, a \varepsilon L \quad .$$

$r^0(a) = r(a)$, for all $a \varepsilon L$, by definition of $r^0(a)$.

$$r^1(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq 1}} x(u_\ell)$$

is true for all a, since if $u_1 \not\varepsilon f(a)$, then $r^1(a) = r^0(a) = r(a)$, and if $u_1 \varepsilon f(a)$, then $r^1(a) = r^0(a) - x(u_1) = r(a) - x(u_1)$. Assume that

$$r^p(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq p}} x(u_\ell), \qquad \forall\, a \quad .$$

$r^{p+1}(a) = r^p(a) - x(u_{p+1})$, if $u_{p+1} \varepsilon f(a)$, and $r^{p+1}(a) = r^p(a)$, otherwise. Therefore,

$$r^{p+1}(a) = r(a) - \sum_{\substack{u_\ell \varepsilon f(a) \\ \ell \leqq p+1}} x(u_\ell) \quad ,$$

which completes the proof of (3.10). It follows from (3.9) and (3.10) that

$$\sum_{u \varepsilon f(a')} x(u) \geqq \sum_{\substack{u_k \varepsilon f(a') \\ k \leqq p}} x(u_k) + r(a') -$$

$$\sum_{\substack{u_k \varepsilon f(a') \\ k \leq p}} x(u_k) - \sum_{\substack{u_\ell \varepsilon f(a') \\ \ell \geq p+2}} d(u_\ell) = r(a') -$$

$$\sum_{\substack{u_\ell \varepsilon f(a') \\ \ell \geq p+2}} d(u_\ell) = r(a') \quad ,$$

since $\{u_\ell | u_\ell \varepsilon f(a'),\ \ell \geq p+2\} = \phi$. This concludes the proof of feasibility of the solution to the primal problem.

To prove feasibility of the solution to the dual, we will first prove the following lemma:

(3.11) <u>Lemma</u>. Assuming feasibility of the problem,

$$a_{j+1} > a_j \text{ for all } j=1,2, \ldots, m-1.$$

<u>Proof</u>: Suppose $a_{j+1} \leq a_j$. Then since f is upper, if $u_k \varepsilon f(a_{j+1})$, $u_k \varepsilon f(a_j)$. Then there is no $u_k \varepsilon f(a_{j+1})$ such that $u_k \not\varepsilon f(a_j)$. Then a_{j+1} could not have been subscripted in step 1 of the algorithm. Therefore, $a_{j+1} \not\leq a_j$. We must show that a_j and a_{j+1} are not incomparable. Suppose they were. There are two cases to consider:

<u>Case I</u>. $u_{\bar{k}(j)} \varepsilon f(a_{j+1})$.

We will be using the fact that

$$(3.12) \qquad r^{\bar{k}(j)-1}(a_j \wedge a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \wedge a_{j+1}) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) \leq$$

$$r^{\bar{k}(j)-1}(a_j) - \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) \quad ,$$

which is true for the following reasons. If $u_{\bar{k}(j)} \varepsilon f(a_j \wedge a_{j+1})$, then (3.12) is true, since otherwise $a_j \wedge a_{j+1}$ would have been subscripted instead of a_j. If $u_{\bar{k}(j)} \not\varepsilon f(a_j \wedge a_{j+1})$, (3.12) must be true, since

$$r^{\bar{k}(j)-1}(a_j) - \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) \geq 0 \quad ,$$

and if (3.12) were not true, it would follow that

$$r^{\bar{k}(j)-1}(a_j \wedge a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \wedge a_{j+1}) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) > 0 \quad ,$$

which implies the problem is infeasible.

By step 2 of the algorithm,

$$(3.13) \qquad r^{\bar{k}(j+1)-1}(a_{j+1}) = r^{\bar{k}(j)}(a_{j+1}) - \sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_{j+1})}} d(u_{k(i)}) \quad ,$$

since for those k(i) such that $\bar{k}(j) < k(i) < \bar{k}(j+1)$, $x(u_{k(i)}) =$

$d(u_{k(i)})$, and for all ℓ such that $\ell \neq k(i)$ for some i and $\bar{k}(j) < \ell < \bar{k}(j+1)$, $x(u_\ell) = 0$. Since $u_{\bar{k}(j)} \varepsilon f(a_{j+1})$,

$$(3.14) \qquad r^{\bar{k}(j)}(a_{j+1}) = r^{\bar{k}(j)-1}(a_{j+1}) - x(u_{\bar{k}(j)}) =$$

$$r^{\bar{k}(j)-1}(a_{j+1}) - r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) \ .$$

(3.13) and (3.14) imply that

$$(3.15) \qquad r^{\bar{k}(j+1)-1}(a_{j+1}) = r^{\bar{k}(j)-1}(a_{j+1}) - r^{\bar{k}(j)-1}(a_j) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) - \sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \ \varepsilon \ f(a_{j+1})}} d(u_{k(i)}) \ .$$

We can similarly see that

$$(3.16) \qquad r^{\bar{k}(j+1)-1}(a_j \vee a_{j+1}) = r^{\bar{k}(j)-1}(a_j \vee a_{j+1}) -$$

$$r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geq \bar{k}(j+1)}} d(u_\ell) -$$

$$\sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_j \vee a_{j+1})}} d(u_{k(i)}) \ .$$

By (3.5),

$$\begin{aligned} & r^{\bar{k}(j)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) - \\ & r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \geqq r^{\bar{k}(j)-1}(a_{j+1}) - \\ & \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) - r^{\bar{k}(j)-1}(a_j \wedge a_{j+1}) + \\ & \sum_{\substack{u_\ell \varepsilon f(a_j \wedge a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \quad . \end{aligned} \tag{3.17}$$

(3.17) and (3.12) imply that

$$\begin{aligned} & r^{\bar{k}(j)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) - \\ & r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \geqq r^{\bar{k}(j)-1}(a_{j+1}) - \\ & \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) - r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \quad . \end{aligned} \tag{3.18}$$

Now

$$(3.19) \qquad \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) = \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geq \bar{k}(j+1)+1}} d(u_\ell) + \sum_{\substack{\bar{k}(j)<k(i)<\bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_j \vee a_{j+1})}} d(u_{k(i)}) + d(u_{\bar{k}(j+1)})$$

for the following reasons. It is quite clear that every term on the right hand side is also on the left hand side. If $u_\ell \varepsilon f(a_j \vee a_{j+1})$ and $\ell \geq \bar{k}(j)+1$, then $u_\ell \varepsilon f(a_j) \cup f(a_{j+1})$ by property (1.22). If $u_\ell \varepsilon f(a_{j+1}) - \bigcup_{q=1}^{j} f(a_q)$ and $\bar{k}(j) < \ell < \bar{k}(j+1)$, then u_ℓ would have been $u_{\bar{k}(j+1)}$. Therefore, $u_\ell \varepsilon f(a_q)$ for some $q \leq j$, so that $u_\ell = u_{k(i)}$ for some i. Thus, every term on the left hand side is also on the right hand side. It can also be shown that

$$(3.20) \qquad \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) = \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geq \bar{k}(j+1)+1}} d(u_\ell) + \sum_{\substack{\bar{k}(j)<k(i)<\bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_j+1)}} d(u_{k(i)}) + d(u_{\bar{k}(j+1)}) .$$

Again, it is clear that every term on the right hand side is on the left hand side. If $\ell \geq \bar{k}(j)+1$ and $u_\ell \varepsilon f(a_{j+1})$, then either $u_{\ell} \varepsilon f(a_q)$ for some $q \leqq j$ so that $\ell = k(i) < \bar{k}(j+1)$ for some i, or $u_{\ell} = u_{\bar{k}(j+1)}$, or $\ell \geqq \bar{k}(j+1) + 1$. Therefore, all terms on the left hand side are also on the right hand side. From (3.18), (3.19), and (3.20) we have

$$\begin{aligned}
(3.21)\qquad & r^{\bar{k}(j)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_{\ell} \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geqq \bar{k}(j+1)+1}} d(u_{\ell}) - \\
& \sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_j \vee a_{j+1})}} d(u_{k(i)}) - d(u_{\bar{k}(j+1)}) - \\
& r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_{\ell} \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_{\ell}) \geqq r^{k(j)-1}(a_{j+1}) - \\
& \sum_{\substack{u_{\ell} \varepsilon f(a_{j+1}) \\ \ell \geqq \bar{k}(j+1)+1}} d(u_{\ell}) - \sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_{j+1})}} d(u_{k(i)}) - \\
& r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_{\ell} \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_{\ell}) \quad .
\end{aligned}$$

(3.21), (3.15), and (3.16) imply that

$$r^{\bar{k}(j+1)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_{\ell} \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geqq \bar{k}(j+1)+1}} d(u_{\ell}) \geqq$$

$$r^{\bar{k}(j+1)-1}(a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geq \bar{k}(j+1)+1}} d(u_\ell) ,$$

which implies that $a_j \vee a_{j+1}$ would have been chosen instead of a_{j+1} in step 1 of the algorithm.

Case II. $u_{\bar{k}(j)} \not\varepsilon f(a_{j+1})$.

Since f is upper, $u_{\bar{k}(j)} \not\varepsilon f(a_{j+1})$ implies that $u_{\bar{k}(j)} \not\varepsilon f(a_j \wedge a_{j+1})$. Therefore, if the problem is feasible,

$$(3.22) \qquad r^{\bar{k}(j)-1}(a_j \wedge a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \wedge a_{j+1}) \\ \ell \geq \bar{k}(j)+1}} d(u_\ell) \leqq 0 .$$

Since $u_{\bar{k}(j)} \not\varepsilon f(a_{j+1})$,

$$(3.23) \qquad r^{\bar{k}(j)}(a_{j+1}) = r^{\bar{k}(j)-1}(a_{j+1}) .$$

Therefore, from (3.13) and (3.23),

$$(3.24) \qquad r^{\bar{k}(j+1)-1}(a_{j+1}) = r^{\bar{k}(j)-1}(a_{j+1}) -$$

$$\sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_{j+1})}} d(u_{k(i)}) .$$

Since $u_{\bar{k}(j)} \varepsilon f(a_j \vee a_{j+1})$, (3.16) still holds. (3.5) and

(3.22) imply that

$$(3.25)\qquad r^{\bar{k}(j)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) - r^{\bar{k}(j)-1}(a_j) +$$

$$\sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \geqq r^{\bar{k}(j)-1}(a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) -$$

$$r^{\bar{k}(j)-1}(a_j \wedge a_{j+1}) + \sum_{\substack{u_\ell \varepsilon f(a_j \wedge a_{j+1}) \\ \ell \geqq \bar{k}(j+1)}} d(u_\ell) \geqq$$

$$r^{\bar{k}(j)-1}(a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \quad .$$

Since (3.19) and (3.20) hold in this case also, (3.25) implies

$$(3.26)\qquad r^{\bar{k}(j)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geqq \bar{k}(j+1)+1}} d(u_\ell) -$$

$$\sum_{\substack{\bar{k}(j) < k(i) < \bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_j \vee a_{j+1})}} d(u_{k(i)}) - d(u_{\bar{k}(j+1)}) -$$

$$r^{\bar{k}(j)-1}(a_j) + \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \geqq \bar{k}(j)+1}} d(u_\ell) \geqq r^{\bar{k}(j)-1}(a_{j+1}) -$$

$$\sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geq \bar{k}(j+1)+1}} d(u_\ell) - \sum_{\substack{\bar{k}(j)<k(i)<\bar{k}(j+1) \\ u_{k(i)} \varepsilon f(a_{j+1})}} d(u_{k(i)}) -$$

$$d(u_{\bar{k}(j+1)}) \quad .$$

(3.26), (3.16), and (3.24) imply

$$(3.27) \qquad r^{\bar{k}(j+1)-1}(a_j \vee a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_j \vee a_{j+1}) \\ \ell \geq \bar{k}(j+1)+1}} d(u_\ell) \geq$$

$$r^{\bar{k}(j+1)-1}(a_{j+1}) - \sum_{\substack{u_\ell \varepsilon f(a_{j+1}) \\ \ell \geq \bar{k}(j+1)+1}} d(u_\ell) \quad .$$

Then $a_j \vee a_{j+1}$ would have been subscripted instead of a_{j+1} in step 1. Therefore, $a_{j+1} > a_j$ must hold, which completes the proof of the lemma.

We will now prove the feasibility of the solution to the dual. $y(a) \geq 0$, for all a, is an immediate result of steps 3, 4, and 6 of the algorithm, using the fact that $c(u_{\bar{k}(j)}) \geq c(u_{\bar{k}(j+1)})$ from step 0.

To prove $z(u_{k(i)}) \geq 0$ for all i, we must prove that

$$\sum_{a_j | u_{k(i)} \varepsilon f(a_j)} y(a_j) - c(u_{k(i)}) \geq 0 \ , \qquad \forall\, i \quad .$$

For each value of i, there is some value of j, say p, such that $\bar{k}(p) < k(i) < \bar{k}(p+1)$. Then $u_{k(i)} \varepsilon f(a_\ell)$ for some least $\ell \leqq p$. Then by (3.11) and (1.17), $u_{k(i)} \varepsilon f(a_p)$. Thus,

$$\sum_{a_j | u_{k(i)} \varepsilon f(a_j)} y(a_j) \geqq y(a_p) = c(u_{\bar{k}(p)}) -$$

$$c(u_{\bar{k}(p+1)}) \geqq c(u_{\bar{k}(p)}) \geqq c(u_{k(i)}) \quad ,$$

since $\bar{k}(p) < k(i)$. For each $k \neq k(i)$ for any i, $z(u_k) = 0$, so that $z(u) \geqq 0$ for all $u \varepsilon U$.

Before proving that the inequalities (1.32) hold, we will prove

$$(3.28) \qquad \sum_{\substack{j \geqq \ell \\ a_j | u_{\bar{k}(\ell)} \varepsilon f(a_j)}} y(a_j) = c(u_{\bar{k}(\ell)}) \text{ for } \ell = 1, \ldots, m.$$

(3.28) will be used in the proof that (1.32) holds and in the proof of optimality. When $\ell = m$, $y(a_m) = c(u_{\bar{k}(m)})$, so that (3.28) is true. Assume that (3.28) is true for $\ell \geqq p+1$. For $\ell = p$, by step 4 of the algorithm and the induction hypothesis,

$$(3.29) \qquad y(a_p) = c(u_{\bar{k}(p)}) - c(u_{\bar{k}(p+1)}) = c(u_{\bar{k}(p)}) -$$

$$\sum_{\substack{j \geqq p+1 \\ a_j | u_{\bar{k}(p+1)} \varepsilon f(a_j)}} y(a_j) \quad .$$

$$(3.30) \qquad \sum_{\substack{j \geqq p+1 \\ a_j | u_{\bar{k}(p+1)} \varepsilon f(a_j)}} y(a_j) = \sum_{\substack{j \geqq p+1 \\ a_j | u_{\bar{k}(p)} \varepsilon f(a_j)}} y(a_j) \quad ,$$

by (3.11) and (1.17). Therefore, since $u_{\bar{k}(p)} \varepsilon f(a_p)$, (3.29) and (3.30) result in

$$\sum_{\substack{j \geqq p \\ a_j | u_{\bar{k}(p)} \varepsilon f(a_j)}} y(a_j) = c(u_{\bar{k}(p)}) \quad ,$$

so that (3.28) is true.

The proof that the inequalities (1.32) hold is also by induction. If k = k(i) for some i, equality holds in (1.32) by step 5 of the algorithm. If k = $\bar{k}$(j) for some j, say j=ℓ, $z(u_{\bar{k}(\ell)}) = 0$, and

$$\sum_{\substack{j \geqq \ell \\ a_j | u_{\bar{k}(\ell)} \varepsilon f(a_j)}} y(a_j) = \sum_{a_j | u_{\bar{k}(\ell)} \varepsilon f(a_j)} y(a_j) =$$

$$\sum_{a | u_{\bar{k}(\ell)} \varepsilon f(a)} y(a) \quad ,$$

so that equality holds in (1.32) by (3.28). For k not equal to k(i) for any i, nor equal to $\bar{k}(j)$ for any j, $z(u_k) = 0$. Therefore, we must show that

$$\sum_{a|u_k \varepsilon f(a)} y(a) \leqq c(u_k) \quad .$$

Consider all u_k such that $k > \bar{k}(m)$. If $u_k \varepsilon f(a_\ell)$ for some $\ell \leqq m$, $x(u_k) = d(u_k)$ would hold and $k = k(i)$ would hold for some i. Therefore, $u_k \not\varepsilon f(a_\ell)$ for any $\ell \leqq m$. But then for all a such that $u_k \varepsilon f(a)$, $y(a) = 0$, so that

$$\sum_{a|u_k \varepsilon f(a)} y(a) = 0 \leqq c(u_k) \quad .$$

Assume that

$$\sum_{a|u_k \varepsilon f(a)} y(a) \leqq c(u_k)$$

for all k such that $\bar{k}(p+1) > k > \bar{k}(p)$. Since $k \neq k(i)$ for any i, $u_k \not\varepsilon f(a_\ell)$ for any $\ell \leqq p$. Therefore, using (3.28) and step 0,

$$\sum_{a|u_k \varepsilon f(a)} y(a) \leqq \sum_{j \geqq p+1} y(a_j) = \sum_{\substack{j \geqq p+1 \\ a_j | u_{\bar{k}(p+1)} \varepsilon f(a_j)}} y(a_j) =$$

$$c(u_{\bar{k}(p+1)}) \leqq c(u_k) \quad .$$

Therefore, (1.32) holds for all $u \varepsilon U$, which concludes the proof of feasibility of the dual solution.

In order to prove optimality of the solutions, we will show that the complementary slackness optimality criterion [1] is satisfied. The complementary slackness optimality criterion of linear programming states that if the solutions to the primal and dual problems are feasible, then they are optimal if

$$(3.31) \qquad x(u_k) > 0 \rightarrow \sum_{a_j | u_k \varepsilon f(a_j)} y(a_j) - z(u_k) = c(u_k) \quad ,$$

$$(3.32) \qquad y(a) > 0 \rightarrow \sum_{u \varepsilon f(a)} x(u) = r(a) \quad ,$$

and

$$(3.33) \qquad z(u) > 0 \rightarrow x(u) = d(u) \quad .$$

$x(u_k) > 0$ only if $k = k(i)$ for some i or $k = \bar{k}(j)$ for some j. If $k = k(i)$, (3.31) is true as a result of step 5 of the algorithm. If $k = \bar{k}(j)$ for some j, say ℓ, by (3.28)

$$c(u_{\bar{k}(\ell)}) = \sum_{\substack{j \geq \ell \\ a_j | u_{\bar{k}(\ell)} \varepsilon f(a_j)}} y(a_j) = \sum_{a_j | u_{\bar{k}(\ell)} \varepsilon f(a_j)} y(a_j) \ .$$

Since $z(u_{\bar{k}(\ell)}) = 0$, (3.31) is true.

To see that (3.32) is true, we note that $y(a) > 0$ only for $a = a_j$, $j = 1, \ldots, m$. By (3.10)

$$r^{\bar{k}(j)-1}(a_j) = r(a_j) - \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \leqq \bar{k}(j)-1}} x(u_\ell) \quad .$$

But

$$r^{\bar{k}(j)-1}(a_j) = x(u_{\bar{k}(j)}) + \sum_{\substack{u_k \varepsilon f(a_j) \\ k \geqq \bar{k}(j)+1}} d(u_k)$$

and $u_{\bar{k}(j)} \varepsilon f(a_j)$. Thus,

$$r(a_j) = \sum_{\substack{u_\ell \varepsilon f(a_j) \\ \ell \leqq \bar{k}(j)}} x(u_\ell) + \sum_{\substack{u_k \varepsilon f(a_j) \\ k \geqq \bar{k}(j)+1}} d(u_k) \quad ,$$

$$\forall \quad j = 1, 2, \ldots, m \quad .$$

Now $x(u_\ell) = d(u_\ell)$ for $u_\ell \varepsilon f(a_j)$ such that $\ell \geqq \bar{k}(j) + 1$. Therefore,

$$r(a_j) = \sum_{u_\ell \varepsilon f(a_j)} x(u_\ell) \quad ,$$

so that (3.32) is true.

$z(u) > 0$ only if $x(u) = d(u)$. Therefore, (3.33) is true as a result of step 5 of the algorithm. This completes the proof that the solutions are optimal. Since U is a finite set, the algorithm will terminate.

CHAPTER 4

AN ALGORITHM FOR THE SUPERMODULAR PROBLEM WITHOUT UPPER BOUNDS

An algorithm for the supermodular problem without upper bounds is presented in this chapter. We now assume that $r(a)$ is monotonic, i.e, $r(a)$ satisfies (1.35) or (1.36), and that L is a finite lattice, but we do not assume that f is either upper or lower. Since (1.16) clearly implies (1.21), we need not assume (1.21).

<u>Theorem 3</u>. Suppose $r(a)$ is monotonic and L is a finite lattice. Then the following algorithm solves the supermodular problem without upper bounds:

Step 0. Let a_0 be the maximum element in L if $r(a)$ satisfies (1.35); let a_0 be the minimum element in L if $r(a)$ satisfies (1.36). Let $c^0(u) = c(u)$ for all $u \in U$. Let $k=0$.

Step 1. Let $y(a_k) = \min_{u|u \in f(a_k)} c^k(u)$. Let u_k be the u giving this minimum. Consider $A_k = \{a|u_\ell \notin f(a);\ \ell=0, 1, \ldots, k\}$. If $A_k = \phi$, go to step 2. Otherwise, choose the maximum element in A_k if $r(a)$

satisfies (1.35); choose the minimum element in A_k if $r(a)$ satisfies (1.36). Call this element a_{k+1}. Let $c^{k+1}(u) = c^k(u) - y(a_k)$ for all $u \varepsilon f(a_k)$. Let $c^{k+1}(u) = c^k(u)$, otherwise. Replace k with k+1 and repeat step 1.

Step 2. Let $x(u_k) = r(a_k) - \sum_{\substack{\ell \geq k+1 \\ u_\ell \varepsilon f(a_k)}} x(u_\ell)$.

If k=0, go to step 3. Otherwise, decrease k by 1 and repeat step 2.

Step 3. Let $y(a) = 0$ for all other $a \varepsilon L$. Let $x(u) = 0$ for all other $u \varepsilon U$.

Proof: The method of proof will be to show that the solutions to the primal and dual problems are both feasible, and then to show that the complementary slackness optimality criterion [1] is satisfied. If $r(a)$ satisfies (1.36) and the order of L is reversed, $r(a)$ will then satisfy (1.35). Therefore, it suffices to prove the theorem in the case that $r(a)$ satisfies (1.35). We therefore assume $r(a)$ satisfies (1.35).

First we must show that, for each k, A_k has a maximum element. Suppose a and b are incomparable elements in A_k. For each $\ell=0, 1, \ldots, k$, $u_\ell \not\varepsilon f(a)$ and $u_\ell \not\varepsilon f(b)$. Therefore, $u_\ell \not\varepsilon f(a) \cup f(b)$. By property (1.22), $u_\ell \not\varepsilon f(a \vee b) \cup f(a \wedge b)$.

Therefore, $a \vee b \varepsilon A_k$, providing the maximum element needed.

We will now prove the following lemma, which is necessary in the proof of the theorem.

(4.1) Lemma. For all k, $a_k > a_{k+1}$.

Proof: It is quite clear that $a_0 > a_1$, since a_0 is the maximum element in L. Now assume $a_0 > a_k > \ldots > a_p$. a_p is the maximum element in $A_{p-1} = \{a \mid u_\ell \not\varepsilon f(a), \ \ell = 0, 1, \ldots, p-1\}$, which contains $\{a \mid u_\ell \not\varepsilon f(a), \ \ell = 0, 1, \ldots, p\} = A_p$. Therefore $a_p > a$ for all $a \varepsilon A_p$, so that, in particular, $a_p > a_{p+1}$. Thus, the subscripted a's form a chain in L.

We will now prove feasibility of the dual solution. In order to prove feasibility, we will show that for all k, $c^k(u) \geqq 0$ for all $u \varepsilon U$. $c^0(u) \geqq 0$ since $c(u) \geqq 0$. Assume $c^p(u) \geqq 0$ for all $u \varepsilon U$. Now $y(a_p) \leqq c^p(u)$ for all $u \varepsilon f(a_p)$, and $c^{p+1}(u) = c^p(u) - y(a_p)$ for all $u \varepsilon f(a_p)$. Therefore, $c^{p+1}(u) \geqq 0$ for all $u \varepsilon f(a_p)$. Also, $c^{p+1}(u) = c^p(u) \geqq 0$, otherwise. It is now quite simple to see that $y(a_k) \geqq 0$ for all k. For all k, $y(a_k) = \min\limits_{u \varepsilon f(a_k)} c^k(u) \geqq 0$.

We will now prove that

$$(4.2) \qquad \sum_{a \mid u \,\varepsilon\, f(a)} y(a) \leqq c(u) \qquad \forall \ u \varepsilon U \ .$$

Consider all $u \varepsilon U$ with the property that a_0 is the minimum $a \varepsilon \hat{L} = \{a_k \mid k = 0, 1, \ldots, m\}$ such that $u \varepsilon f(a)$, where m represents the highest value k attains in the algorithm. Then, using (4.1),

$$\sum_{a|u\varepsilon f(a)} y(a) = \sum_{a_\ell|u\varepsilon f(a_\ell)} y(a_\ell) = y(a_0) \leqq c(u)$$

for all u such that $u \varepsilon f(a_0)$, by definition of $y(a_0)$, since $c(u) = c^0(u)$. Assume now that (4.2) holds for all u with the property that a_p is the minimum element a in $\hat{L}$ such that $u \varepsilon f(a)$. Consider any u' with the property that a_{p+1} is the minimum element a in $\hat{L}$ such that $u' \varepsilon f(a)$.

$$\sum_{a|u'\varepsilon f(a)} y(a) = \sum_{a_\ell|u' \varepsilon f(a_\ell)} y(a_\ell) = \sum_{\substack{a_\ell|u'\varepsilon f(a_\ell) \\ \ell<p+1}} y(a_\ell) +$$

$$y(a_{p+1}) = \sum_{\substack{a_\ell|u'\varepsilon f(a_\ell) \\ \ell<p+1}} y(a_\ell) + \min_{u\varepsilon f(a_{p+1})} c^{p+1}(u) \leqq$$

$$\sum_{\substack{a_\ell|u'\varepsilon f(a_\ell) \\ \ell<p+1}} y(a_\ell) + c^{p+1(u)}$$

for all $u \varepsilon f(a_{p+1})$. Therefore, in particular,

$$\text{(4.3)} \qquad \sum_{a|u'\varepsilon f(a)} y(a) \leqq \sum_{\substack{a_\ell|u'\varepsilon f(a_\ell) \\ \ell<p+1}} y(a_\ell) + c^{p+1}(u') \quad .$$

To complete the proof, we will show that for all $u \varepsilon U$

$$(4.4) \qquad c^k(u) = c(u) - \sum_{\substack{a_\ell | u \varepsilon f(a_\ell) \\ \ell < k}} y(a_\ell) \quad .$$

By definition of $c^0(u)$, $c^0(u) = c(u)$.

$$c^1(u) = c(u) - \sum_{\substack{a_\ell | u \varepsilon f(a_\ell) \\ \ell < 1}} y(a_\ell)$$

is true, since if $u \notin f(a_0)$, then $c^1(u) = c^0(u) = c(u)$, and if $u \varepsilon f(a_0)$, $c^1(u) = c^0(u) - y(a_0) = c(u) - y(a_0)$. Assume that

$$c^{p-1}(u) = c(u) - \sum_{\substack{a_\ell | u \varepsilon f(a_\ell) \\ \ell < p-1}} y(a_\ell) \quad .$$

$c^p(u) = c^{p-1}(u) - y(a_{p-1})$ if $u \varepsilon f(a_{p-1})$ and $c^p(u) = c^{p-1}(u)$, otherwise. Therefore, by the induction hypothesis,

$$c^p(u) = c(u) - \sum_{\substack{a_\ell | u \varepsilon f(a_\ell) \\ \ell < p}} y(a_\ell) \quad ,$$

so that (4.4) is true. From (4.3) and (4.4),

$$\sum_{a | u' \varepsilon f(a)} y(a) \leqq \sum_{\substack{a_\ell | u' \varepsilon f(a_\ell) \\ \ell < p+1}} y(a_\ell) + c(u') -$$

$$\sum_{\substack{a_\ell | u' \varepsilon f(a_\ell) \\ \ell < p+1}} y(a_\ell) = c(u') \quad ,$$

which completes the proof of (4.2), so that we have proven feasibility of the solution to the dual.

Feasibility of the primal solution will now be proven. To show that $x(u) \geqq 0$ for all $u \varepsilon U$, we will prove that

$$(4.5) \qquad r(a_k) \geqq \sum_{\substack{u_\ell \varepsilon f(a_{k-1}) \\ \ell \geqq k}} x(u_\ell) \quad , \qquad \text{for all } k \quad ,$$

by induction on k. When k=m,

$$\sum_{\substack{u_\ell \varepsilon f(a_{m-1}) \\ \ell \geqq m-1}} x(u_\ell) = x(u_m) \qquad \text{or} \qquad \sum_{\substack{u_\ell \varepsilon f(a_{m-1}) \\ \ell \geqq m-1}} x(u_\ell) = 0 \quad .$$

Since $0 \leqq r(a_m)$ and $x(u_m) \leqq r(a_m)$, in either case (4.5) is true. Assuming (4.5) to be true for k=p+1, we must prove (4.5) is true for k=p. Since $u_p \varepsilon f(a_p)$,

$$(4.6) \qquad r(a_p) = \sum_{\substack{u_\ell \varepsilon f(a_p) \\ \ell \geqq p}} x(u_\ell) \quad ,$$

by step 2 of the algorithm. By property (1.16) and because for all ℓ we have $u_\ell \varepsilon f(a_\ell)$, it follows that if $\ell \geqq p$,

$u_\ell \varepsilon f(a_{p-1})$ implies $u_\ell \varepsilon f(a_p)$. Therefore,

$$(4.7) \qquad \sum_{\substack{u_\ell \varepsilon f(a_p) \\ \ell \geqq p}} x(u_\ell) \geqq \sum_{\substack{u_\ell \varepsilon f(a_{p-1}) \\ \ell \geqq p}} x(u_\ell) \quad .$$

(4.6) and (4.7) imply

$$r(a_p) \geqq \sum_{\substack{u_\ell \varepsilon f(a_{p-1}) \\ \ell \geqq p}} x(u_\ell) \ ,$$

which completes the proof of (4.5). Now we can show that $x(u) \geqq 0$. Obviously, this only needs to be shown for the subscripted u's. When k=m, $x(u_m) = r(a_m) \geqq 0$. Assume $x(u_{p+1}) \geqq 0$.

$$x(u_p) = r(a_p) - \sum_{\substack{u_\ell \varepsilon f(a_p) \\ \ell \geqq p+1}} x(u_\ell) \geqq r(a_p) - r(a_{p+1}) \quad ,$$

by (4.5), and $r(a_k) - r(a_{k+1}) \geqq 0$ by monotonicity of r(a) and lemma (4.1). Therefore, $x(u_k) \geqq 0$ for all k=1, 2, ..., m.

The proof that

$$(4.8) \qquad \sum_{u | u \varepsilon f(a)} x(u) \geqq r(a) \qquad \forall \ a \varepsilon L$$

is satisfied is also by induction. First, (4.8) is true for all $a \leqq a_m$. This is so because if $a < a_m$, then the only $u_k \varepsilon f(a)$ is u_m, since if there were any $u_k \varepsilon f(a)$ such that $k < m$, then $u_k \varepsilon f(a_m)$ would be true because of property (1.16) and the fact that $u_k \varepsilon f(a_k)$. But $a_m \varepsilon A_{m-1}$, and therefore $u_\ell \not\varepsilon f(a_m)$, $\ell = 0, 1, \ldots, m-1$. $u_m \varepsilon f(a)$ must be true since A_m was empty, i.e., there were no a's such that $u_k \not\varepsilon f(a)$, $k = 0, 1, \ldots, m$. Thus, for all $a \leqq a_m$, $u_m \varepsilon f(a)$. Now $x(u_m) = r(a_m) \geqq r(a)$ for all $a \leqq a_m$, by (1.35). Therefore (4.8) is true for all $a \leqq a_m$. Assume (4.8) is true for all $a \leqq a_{p+1}$. Consider a', $a' \not\leqq a_{p+1}$, but $a' < a_p$, and suppose

$$\sum_{u\varepsilon f(a')} x(u) < r(a') \quad . \tag{4.9}$$

Since $a' \wedge a_{p+1} < a_{p+1}$, by the induction hypothesis,

$$\sum_{u\varepsilon f(a' \wedge a_{p+1})} x(u) \geqq r(a' \wedge a_{p+1}) \quad . \tag{4.10}$$

Hence, by (4.9) and (4.10),

$$r(a' \wedge a_{p+1}) - r(a') < \sum_{u\varepsilon f(a' \wedge a_{p+1})} x(u) - \sum_{u\varepsilon f(a')} x(u) \quad . \tag{4.11}$$

By supermodularity,

$$(4.12)\qquad r(a'\wedge a_{p+1}) + r(a'\vee a_{p+1}) \geqq r(a') + r(a_{p+1}) \quad ,$$

which implies

$$(4.13)\qquad r(a'\wedge a_{p+1}) - r(a') \geqq r(a_{p+1}) - r(a'\vee a_{p+1}) \quad .$$

By (4.11) and (4.13),

$$(4.14)\qquad \sum_{u\varepsilon f(a'\wedge a_{p+1})} x(u) - \sum_{u\varepsilon f(a')} x(u) > r(a_{p+1}) - r(a'\vee a_{p+1}) \quad .$$

Since it is assumed that L is a lattice, and since $a_p > a'$ and $a_p > a_{p+1}$, it is true that $a_p \geqq a' \vee a_{p+1}$, which implies, by (1.35), $r(a_p) \geqq r(a'\vee a_{p+1})$. Thus

$$(4.15)\qquad r(a_{p+1}) - r(a'\vee a_{p+1}) \geqq r(a_{p+1}) - r(a_p) \quad .$$

Therefore, by (4.14) and (4.15),

$$\sum_{u\varepsilon f(a'\wedge a_{p+1})} x(u) - \sum_{u\varepsilon f(a')} x(u) > r(a_{p+1}) - r(a_p) = \sum_{u\varepsilon f(a_{p+1})} x(u) - \sum_{u\varepsilon f(a_p)} x(u) \quad .$$

Therefore, to prove that a contradiction exists, it suffices to show that

$$(4.16)\qquad \sum_{u\varepsilon f(a_{p+1})} x(u) - \sum_{u\varepsilon f(a_p)} x(u) \geqq \sum_{u\varepsilon f(a'\wedge a_{p+1})} x(u) - \sum_{u\varepsilon f(a')} x(u) \ .$$

Since

$$\sum_{u\varepsilon f(a_{p+1})} x(u) - \sum_{u\varepsilon f(a_p)} x(u) = \sum_{u\varepsilon f(a_{p+1})-f(a_p)} x(u) - \sum_{u\varepsilon f(a_p)-f(a_{p+1})} x(u)$$

and

$$\sum_{u\varepsilon f(a'\wedge a_{p+1})} x(u) - \sum_{u\varepsilon f(a')} x(u) = \sum_{u\varepsilon f(a'\wedge a_{p+1})-f(a')} x(u) - \sum_{u\varepsilon f(a')-f(a'\wedge a_{p+1})} x(u) \ ,$$

it is equivalent to (4.16) to show that

$$(4.17)\qquad \sum_{u_k\varepsilon f(a_{p+1})-f(a_p)} x(u_k) - \sum_{u_k\varepsilon f(a_p)-f(a_{p+1})} x(u_k) \geqq$$

$$\sum_{u_k \varepsilon f(a' \wedge a_{p+1}) - f(a')} x(u_k) - \sum_{u_k \varepsilon f(a') - f(a' \wedge a_{p+1})} x(u_k) \ .$$

(4.17) will be shown by proving that

(4.18) $f(a' \wedge a_{p+1}) - f(a') \subsetneq f(a_{p+1}) - f(a_p)$

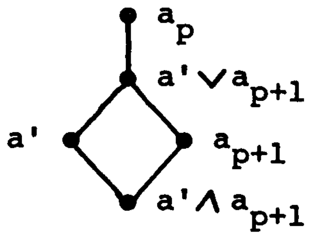
and

(4.19) $\{u_k | u_k \varepsilon f(a_p) - f(a_{p+1})\} \subseteq \{u_k | u_k \varepsilon f(a') - f(a' \wedge a_{p+1})\}$.

To prove (4.18), let $u \varepsilon f(a' \wedge a_{p+1}) - f(a')$. By property (1.22), $u \varepsilon f(a_{p+1})$. To show $u \not\varepsilon f(a_p)$, in (1.16) let $a = a' \wedge a_{p+1}$, $b = a'$, and $c = a_p$. Since $a_p > a' > a' \wedge a_{p+1}$ and $u \varepsilon f(a' \wedge a_{p+1}) - f(a')$, $u \not\varepsilon f(a_p)$. To prove (4.19) we must prove that

(4.20) $\{u_k | u_k \varepsilon f(a_p)\} \subsetneq \{u_k | u_k \varepsilon f(a' \vee a_{p+1})\}$

Since L is a lattice, the following Hasse diagram represents the situation:



This is so because $a_p > a'$ and $a_p > a_{p+1}$, but $a' \vee a_{p+1}$ is the least element which is greater than both a' and a_{p+1}. It is possible, of course, that $a_p = a' \vee a_{p+1}$, in which case (4.20) is trivial. Now $u_k \notin f(a_p)$ for $k = 0, 1, \ldots, p-1$. Therefore, since each u_k is in $f(a_k)$, $u_k \notin f(a' \vee a_{p+1})$ for $k = 0, 1, \ldots, p-1$, by property (1.16) and lemma (4.1). $u_p \varepsilon f(a_p)$, and $u_p \varepsilon f(a' \vee a_{p+1})$, since if it were not, $u_k \notin f(a' \vee a_{p+1})$ for $k = 0, 1, \ldots, p$ would be true, which would imply that $a' \vee a_{p+1}$ is in A_p. But a_{p+1} is the maximum element in A_p, and $a' \vee a_{p+1} \neq a_{p+1}$ because $a_{p+1} \not> a'$. Thus, $u_p \varepsilon f(a' \vee a_{p+1})$. For any $k \geqq p+1$, if $u_k \varepsilon f(a_p)$, then $u_k \varepsilon f(a' \vee a_{p+1})$ must hold, since $u_k \varepsilon f(a_k)$. Therefore, (4.20) is true. Now, to prove (4.19), let $u_k \varepsilon f(a_p) - f(a_{p+1})$. By (4.20), $u_k \varepsilon f(a' \vee a_{p+1}) - f(a_{p+1})$. By property (1.22), $u_k \varepsilon f(a')$. To show $u_k \notin f(a' \wedge a_{p+1})$, in property (1.16) let $c = a_p$, $b = a_{p+1}$, and $a = a' \wedge a_{p+1}$. $a_p > a_{p+1} > a' \wedge a_{p+1}$ and $u_k \varepsilon f(a_p) - f(a_{p+1})$ implies $u_k \notin f(a' \wedge a_{p+1})$. Therefore, a contradiction exists, proving that

$$\sum_{u \varepsilon f(a')} x(u) \geqq r(a')$$

must hold for any $a' < a_p$. Thus, (4.8) is true for all $a \varepsilon L$, which completes the proof of feasibility of the solution to the primal.

In order to prove optimality of the solutions, we will show that the complementary slackness optimality criterion [1] is satisfied. This criterion states that if the solutions to the primal and dual problems are feasible, then they are optimal if

$$(4.21)\qquad x(u) > 0 \rightarrow \sum_{a_\ell | u \varepsilon f(a_\ell)} y(a_\ell) = c(u)$$

and

$$(4.22)\qquad y(a) > 0 \rightarrow \sum_{u \varepsilon f(a)} x(u) = r(a) \quad .$$

$x(u) > 0$ only for $u = u_k$, $k = 0, 1, \ldots, m$. Therefore, we must show that for each $k = 0, 1, \ldots, m$,

$$\sum_{a_\ell | u_k \varepsilon f(a_\ell)} y(a_\ell) = c(u_k) \quad .$$

From (4.4) we have

$$c^k(u_k) = c(u_k) - \sum_{\substack{a_\ell | u_k \varepsilon f(a_\ell) \\ \ell < k}} y(a_\ell) \quad .$$

But by step 1 of the algorithm, $c^k(u_k) = y(a_k)$, so that

$$c(u_k) = \sum_{\substack{a_\ell | u_k \varepsilon f(a_\ell) \\ \ell \leq k}} y(a_\ell) = \sum_{a_\ell | u_k \varepsilon f(a_\ell)} y(a_\ell) \quad ,$$

since for any a_ℓ, $\ell > k$, $u_k \notin f(a_\ell)$. Thus, (4.21) is satisfied.

$y(a) > 0$ only for $a = a_k$, $k = 0, 1, \ldots, m$. Thus, we must show for $k = 0, 1, \ldots, m$ that

$$\sum_{u_\ell \varepsilon f(a_k)} x(u_\ell) = r(a_k) \quad . \tag{4.23}$$

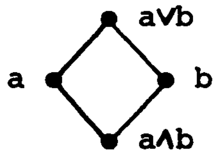
According to step 2 of the algorithm,

$$\sum_{\substack{u_\ell \varepsilon f(a_k) \\ \ell \geq k}} x(u_\ell) = r(a_k).$$

But if $\ell < k$, $u_\ell \notin f(a_k)$, so that (4.23), and thus (4.22), is true, which completes the proof of optimality.

It is clear that the algorithm terminates, since L is finite. Therefore, A_k will be empty for some value of k, which was called m.

We might wish to know if the "greedy" approach will work on the dual of the submodular problem without upper bounds, under the assumption that r(a) is monotonic. However, the following simple example will convince us that the "greedy" approach will not work. Consider the lattice $L = \{a \vee b, a, b, a \wedge b\}$ with the following Hasse diagram:



Let $r(a \vee b) \geqq r(a), r(b)$ and $r(a \wedge b) \leqq r(a), r(b)$, i.e., we will assume $r(a)$ satisfies (1.35). Let $U = \{u_1, u_2, u_3\}$, $f(a \vee b) = \{u_1, u_2\}$, $f(a) = \{u_1\}$, $f(b) = \{u_2\}$, and $f(a \wedge b) = \{u_3\}$. There are no upper bounds on the dual variables $y(a)$, so that we can assume $y(a) = +\infty$ for all $a \in L$, before we make each variable as small as possible. $r(a \vee b) \geqq r(a), r(b), r(a \wedge b)$, so that we would first assign the smallest possible value to $y(a \vee b)$, while insuring that the constraints can be satisfied. We would therefore let $y(a \vee b) = \max[c(u_1) - y(a), c(u_2) - y(b), 0] = 0$. Whether $r(a) \geqq r(b)$ or $r(a) \leqq r(b)$, we would next let $y(a) = c(u_1)$ and $y(b) = c(u_2)$. Finally, we would let $y(a \wedge b) = c(u_3)$. Then $\sum_{\bar{a} \in L} r(\bar{a}) y(\bar{a}) = c(u_1) r(a) + c(u_2) r(b) + c(u_3) r(a \wedge b)$. However, if we did not take the "greedy" approach, we could let $y(a \vee b) = \max[c(u_1), c(u_2)]$, $y(a) = 0$, $y(b) = 0$, and $y(a \wedge b) = c(u_3)$. Suppose that $c(u_1) = c(u_2) \neq 0$. Then $c(u_1) r(a) + c(u_2) r(b) = c(u_1)[r(a) + r(b)] \geqq c(u_1) r(a \vee b)$, by submodularity, (1.23). Then the solution obtained by the "greedy" approach would be greater than the latter solution. Thus, we see that the "greedy" approach will not work in this situation.

REFERENCES

[1] G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.

[2] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, 1 (1959), 269-271.

[3] J. Edmonds, "Matroids and the Greedy Algorithm," Mathematical Programming, 1 (1971), 127-136.

[4] J. Edmonds, "Submodular Functions, Matroids, and Certain Polyhedra," in Combinatorial Structures and their Applications, Gordon and Breach, New York, 1970, pp. 69-87.

[5] J. Edmonds and R. Giles, "A Min-Max Relation for Submodular Functions on Graphs," C.O.R.E. Discussion Paper 7615, Université Catholique de Louvain, 1976.

[6] J.W. Gaddum, A.J. Hoffman, and D. Sokolowsky, "On the Solution of the Caterer Problem," Naval Res. Logist. Quart., 1 (1954), 154-165.

[7] F. Harary, Graph Theory, Addison-Wesley Publishing Company, Reading, Massachusetts, 1969.

[8] A.J. Hoffman, "A Generalization of Max Flow - Min Cut," Mathematical Programming, 6 (1974), 352-359.

[9] A.J. Hoffman, "On Simple Linear Programming Problems," Convexity, Proc. of the Seventh Sympos. in Pure Math., edited by V. Klee, Amer. Math. Soc., Providence, R.I., 1963, 317-327.

[10] A.J. Hoffman, "On Lattice Polyhedra II: Construction and Examples," IBM Research Report.

[11] A.J. Hoffman, "On Lattice Polyhedra III: Blockers and Anti-Blockers of Lattice Clutters," IBM Research Report.

[12] A.J. Hoffman and D.E. Schwartz, "On Lattice Polyhedra," Proc. Keszthely Combinatorics Conference, 1975.

[13] E. Johnson, "On Cut-Set Integer Polyhedra," Cahiers du Centre de Recherches Operationelle, 17 (1975), 235-251.

[14] J.B. Kruskal, Jr., "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," Proc. Amer. Math. Soc., 7 (1956), 48-50.

[15] E. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.

[16] H. Whitney, "On the Abstract Properties of Linear Dependence," Amer. J. Of Math., 57 (1935), 509-533.