P. Hansen, ed., Studies on Graphs and Discrete Programming

MAXIMIZING SUBMODULAR SET FUNCTIONS: FORMULATIONS AND ANALYSIS OF ALGORITHMS*

G.L. NEMHAUSER

School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, USA

L.A. WOLSEY

CORE, Université Catholique de Louvain, Louvain-la-Neuve, Belgium

We consider integer programming formulations of problems that involve the maximization of submodular functions. A location problem and a 0–1 quadratic program are well-known special cases. We give a constraint generation algorithm and a branch-and-bound algorithm that uses linear programming relaxations. These algorithms are familiar ones except for their particular selections of starting constraints, subproblems and partitioning rules. The algorithms use greedy heuristics to produce feasible solutions, which, in turn, are used to generate upper bounds. The novel features of the algorithms are the performance guarantees they provide on the ratio of lower to upper bounds on the optimal value.

1. Introduction

The following four problems are fundamental to the practical solution of integer linear programming problems.

(a) *Formulation of the model.* There are, in many instances, a wide variety of different linear constraints that can, together with integrality conditions, be used to represent the same set of points of a (mixed) integer linear program. Generally these equivalent integer programming formulations give different linear programs when the integrality constraints are suppressed (see [18] for several examples). Solving these linear programs is an essential part of the fundamental algorithms (branch-and-bound and cutting plane) of integer programming. Both the size of the linear programs and the quality of the bounds they produce determine the success of an integer programming algorithm. Thus it would be highly desirable to have systematic ways of producing different formulations and to have criteria for comparing them.

(b) *Selection of an initial set of constraints.* There are integer programs whose formulations require a very large number of linear constraints. A particular example is the well-known formulation of the traveling salesman

* This work has been supported by a NATO Systems Science special research and U.S. National Science Foundation grant to Cornell University

problem, which requires an exponential number of constraints to eliminate subtours [6]; a general class of these problems is the formulation obtained by applying Benders' decomposition [2] to mixed integer linear programs. In these instances a general algorithmic approach is to begin with a relaxation that contains only a relatively small number of linear equalities and to generate others only if they are found to be violated in the course of the algorithm. One supposedly tries to choose the initial constraints on the basis of their importance or significance in defining the feasible region in a neighborhood of an optimal solution. It would be desirable to begin with a set of constraints having the property that the optimal value to the relaxed problem (at some stage) was guaranteed to be no larger than a fixed multiple of the optimal value of the original problem.

(c) *Decisions in a branch-and-bound algorithm.* It has been widely observed that the criterion for choosing the order in which subproblems will be solved can have a substantial effect on the running time of a branch-and-bound algorithm. In fact, most commercial codes use a combination of rules in order to balance the objectives of obtaining feasible solutions quickly and minimizing the amount of enumeration [12]. Another important decision in a branch-and-bound algorithm is the type of partition and the rule for choosing a variable (or set of variables) at each node of the tree to form the partition.

It would be desirable to have criteria for choosing subproblems and partitions that would, with a given amount of computation, guarantee a feasible solution whose value is a specified fraction of the upper bound.

(d) *Finding good feasible solutions.* Since an integer programming algorithm may have to be terminated before it completes the solution of a problem, an important feature of an algorithm is its capability of producing good feasible solutions. Although feasible solutions can be generated in a distinct heuristic phase, a highly desirable feature is an optimizing algorithm that yields feasible solutions (hopefully of increasing value) as intermediate results.

There is almost no mathematical theory that addresses these problems for general integer programs. It seems that some structure must be imposed even to obtain partial answers. In the few instances where there is enough structure to formulate the integer program directly with a set of linear constraints whose extreme points are the feasible integer solutions (e.g., network flows and matching) the formulation question is solved. In the case of network flows this formulation is compact and there are efficient algorithms so that questions (b), (c) and (d) are irrelevant. In the case of matching the number of linear constraints required is exponential in the number of variables. But questions (c) and (d) are still irrelevant since there are efficient algorithms that begin with a small number of constraints and then generate only a small number of the remaining ones.

For most integer programs all four of the problems are relevant and current practice is to cope with them largely by intuition and experience. Our intention is to provide some theoretical and possibly computationally useful answers to these problems for a very limited but nontrivial class of combinatorial optimization problems. This class contains some problems of practical interest. Furthermore our results suggest potentially useful algorithms for a much larger class of problems.

Our approach will be to design algorithms that with a specified amount of computation give both a lower bound (from a feasible solution) and an upper bound on the optimal value. These bounds have the property that the lower bound is at least a certain fraction of the upper bound. They will be obtained by giving appropriate initial sets of constraints and by specifying decision rules for a branch-and-bound algorithm. We will also show how the structure produces alternate integer linear programming formulations.

The structure that we require is submodularity. We will introduce this structure in terms of a practical and well-known integer programming problem. This problem involves the location of K facilities to maximize the profit from supplying a commodity to m clients. The set of available locations is $N = \{1, \ldots, n\}$ and $K < n$. There is no limit on the number of clients that can be supplied from a given facility; $c_{ij} \geqslant 0$ is the profit obtained by supplying client i from a facility at location j. Thus a particular problem is specified by an $m \times n$ nonnegative matrix $C = \{c_{ij}\}$ and a positive integer K.

A standard mixed integer linear programming formulation of this problem is

$$
\begin{aligned}
V = \max \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}, \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m, \\
& x_{ij} - y_j \leqslant 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n, \\
& \sum_{j=1}^{n} y_j = K, \\
& x_{ij} \geqslant 0, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \\
& y_j \in \{0, 1\}, \quad j = 1, \ldots, n
\end{aligned} \tag{1.1}
$$

where $y_j = 1$ means that a facility is placed at location j.

The y_j's are the strategic variables since given an $S \subseteq N$ and its characteristic vector y^S ($y_j^S = 1, j \in S$ and $y_j^S = 0$, otherwise) an optimal set of x_{ij}'s is given by

$$
x_{ij}^S = \begin{cases} 1 & \text{for some } j \text{ such that } c_{ij} = \max_{k \in S} c_{ik}, \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, m.
$$

The value of a solution (x^S, y^S) is

$$v(S)=\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}^S=\sum_{i=1}^{m}\max_{j\in S} c_{ij}.$$

We can therefore restate the problem as

$$V=\max(v(S):\ S\subseteq N, |S|=K). \tag{1.2}$$

With $v(\emptyset)=0$, it is well-known and easy to see that the set function $v(S)$ satisfies for all $S, T\subseteq N$

$$v(S)+v(T)\geq v(S\cup T)+v(S\cap T). \tag{1.3}$$

Set functions that satisfy (1.3) are called *submodular*. In addition, $v(S)$ satisfies for all $S\subset N$ and $j\notin S$

$$v(S\cup\{j\})-v(S)\geq 0. \tag{1.4}$$

Set functions that satisfy (1.4) are called *nondecreasing*.

Thus a natural generalization of this K-location problem is problem $\mathcal{P}_K$ given by

$$Z=\max(z(S):\ S\subseteq N, |S|=K,\\ z \text{ submodular and nondecreasing}, z(\emptyset)=0). \tag{1.5}$$

Problem family $\mathcal{P}_K$ is the class of problems for which we obtain theoretical results.

In [3] it was shown that a simple greedy heuristic for problem (1.1) produces a value V^G that satisfies for all nonnegative $m\times n$ matrices C and all m and n

$$\frac{V^G}{V^{LP}}\geq 1-\left(\frac{K-1}{K}\right)^K\geq\frac{e-1}{e}\cong 0.63 \tag{1.6}$$

where V^{LP} is the value of the linear programming relaxation of (1.1) obtained by replacing $y_j\in\{0,1\}$ by $y_j\geq 0$ and e is the base of the natural logarithm. Since $V^G\leq V\leq V^{LP}$, (1.6) implies

$$\frac{V^G}{V}\geq 1-\left(\frac{K-1}{K}\right)^K. \tag{1.7}$$

It was shown in [17] that (1.7) generalized to the problem $\mathcal{P}_K$, in other words to all nondecreasing submodular functions. A slightly more general result of [17] is that partial enumeration of all solutions of cardinality q together with a greedy choice of the remaining $K-q$ elements yields for problem (1.5)

$$\frac{Z^{G(q)}}{Z}\geq 1-\left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}, \tag{1.8}$$

where $Z^{G(q)}$ is the value of the q-enumeration plus greedy solution. Note that putting $q=0$ (the greedy algorithm) in (1.8) yields (1.7).

Our interest in the q-enumeration plus greedy family of algorithms is that in a certain sense they are an optimal family of heuristics for $\mathcal{P}_K$. For fixed K, the q-enumeration plus greedy algorithm requires $O(n^{q+1})$ values[1] of the function z. In [16] we showed that any other algorithm for $\mathcal{P}_K$ that could give a performance guarantee (bound) larger than the right-hand side of (1.8) would require at least $O(n^{q+2})$ values of the function z.

In contrast with this prior work on heuristics, here we will be concerned with exact algorithms for solving $\mathcal{P}_K$ and particular cases of it. Nevertheless, the results on heuristics will be relevant since we will adopt a point of view motivated by the questions raised at the beginning of this section.

In Section 2 we give a linearization of nondecreasing, submodular functions the leads immediately to an integer linear programming formulation of $\mathcal{P}_K$ containing a large number of constraints. In Section 3 we propose a constraint generation algorithm for this formulation, which is similar to Benders' algorithm [2]. Section 4 gives a branch-and-bound algorithm that uses linear programming relaxations. These algorithms are familiar ones except for their particular selections of starting constraints, subproblems, and partitioning rules. The algorithms use heuristics to produce feasible solutions, which, in turn, are used to generate upper bounds. The novel features of the algorithms are the performance guarantee they provide on the ratio of lower to upper bounds on the optimal value.

In Section 5 we consider the maximization of a general submodular function subject to linear constraints. These are the problems for which the earlier results may have computational significance as the algorithms extend rather naturally to this larger class. Section 6 addresses the question of simplified linearizations for cases in which the submodular function has additional structure. We apply these ideas to the K-location problem and the minimization of some quadratic 0–1 programs.

2. Linearization of nondecreasing submodular functions and an integer programming formulation

Let $\rho_j(S)=z(S\cup\{j\})-z(S)$, $\forall S\subset N$ and $j\in N$.

Proposition 1 [17]. *A real-valued function z on the subsets of N is submodular*

[1] We use $O(n^r)$ for the family of functions that is bounded below by c_1n^r and above by c_2n^r for some $0<c_1<c_2$.

and nondecreasing if and only if

(a) $\rho_j(S) \geqslant \rho_j(T) \geqslant 0$, $\forall S \subset T \subset N$ *and* $j \in N-T$, *or*

(b) $z(T) \leqslant z(S) + \sum_{j \in T-S} \rho_j(S)$, $\forall S, T \subseteq N$.

Recall that y^U is the characteristic vector of $U \subseteq N$. Consider the set

$$X = \left\{(\eta, y)\colon \eta \leqslant z(S) + \sum_{j \in N-S} \rho_j(S) y_j, \forall S \subseteq N, y_j \in \{0, 1\}, j \in N\right\}.$$

Lemma 1. *If z is submodular and nondecreasing, then $(\xi, y^U) \in X$ if and only if $\xi \leqslant z(U)$.*

Proof. Suppose $\xi \leqslant z(U)$. Then for all $S \subseteq N$

$$z(S) + \sum_{j \in N-S} \rho_j(S) y_j^U = z(S) + \sum_{j \in U-S} \rho_j(S) \geqslant z(U) \geqslant \xi,$$

where the first inequality follows from Proposition 1(b). Conversely $(\xi, y^U) \in X$ implies in particular that

$$\xi \leqslant z(U) + \sum_{j \in N-U} \rho_j(U) y_j^U = z(U).$$

Consider problem (1.5) and the linear integer program

$$\begin{aligned} \max \quad & \eta \\ & \eta \leqslant z(S) + \sum_{j \in N-S} \rho_j(S) y_j, \quad \forall S \subseteq N, \\ & \sum_{j \in N} y_j = K, \\ & y_j \in \{0, 1\}, \quad j \in N. \end{aligned} \tag{2.1}$$

Problem (2.1) is a reformulation of (1.5) since Lemma 1 implies:

Theorem 1. $(\eta, y) = (z(U), y^U)$ *is an optimal solution to* (2.1) *if and only if U is an optimal solution to* (1.5).

Note that although the inequalities of (2.1) are valid for all $S \subseteq N$, they are needed only for $|S| = K$. However, as we shall see later, there may be computational advantages in using some of the inequalities corresponding to sets of cardinality smaller than K.

When the particular class of submodular functions v obtained from the K-location problem are used in (2.1) we obtain an integer linear programming formulation of the location problem. Although we will not give the details here, a surprising observation is that this integer linear program is precisely the integer program obtained by applying Benders' decomposition (with natural choice of dual extreme points) to the mixed integer linear programming formulation (1.1), see [10, 15]. The results of the next section will suggest how we might choose an initial set of constraints if we are going to solve the K-location problem by Benders' algorithm.

3. A constraint generation algorithm

First we describe an algorithm for problem (2.1), and then we analyze its behavior for a particular initialization. An example is given in an appendix.

Algorithm 1

Initialization. Let $Q^t=\{R^0,\ldots,R^{t-1}\}$ be a nonempty set of distinct subsets of N. Set $p=t$.

Iteration p

Step 1. Solve the problem

$$\begin{aligned}\eta^p=\max\quad & \eta,\\ & \eta\leqslant z(S)+\sum_{j\in N-S}\rho_j(S)y_j,\quad S\in Q^p,\\ & \sum_{j\in N}y_j=K,\\ & y_j\in\{0,1\},\quad j\in N.\end{aligned}$$

Let (η^p, y^{R^p}) be an optimal solution.

Step 2. (a) If $\eta^p=z(R^p)$, terminate. R^p is optimal.

(b) If $\eta^p>z(R^p)$, set $Q^{p+1}=Q^p\cup\{R^p\}$, $p\leftarrow p+1$, and return to Step 1.

The optimality of R^p when Step 2(a) occurs is implied by Lemma 1. To prove finiteness we note that if Step 2(b) occurs, $\eta^p>z(R^p)$ implies $R^p\notin Q^p$. Also, since R^p is feasible the algorithm must terminate after at most $\binom{n}{K}$ iterations of Step 2.

We note that $\eta^t\geqslant\cdots\geqslant\eta^p\geqslant\eta^{p+1}\geqslant\cdots\geqslant Z$ and $Z\geqslant z(S)\ \forall S\in Q^p$, so that the nonincreasing upper bound η^p and the nondecreasing lower bound $\zeta^p=\max_{S\in Q^p}z(S)$ give some measure of the progress of the algorithm.

With an appropriate choice of $Q=Q^t$, a performance guarantee can be obtained relating η^t and ζ^t. The value of the guarantee depends on t. For fixed K, we will give $O(n^q)$ initial constraints for $q=0,\ldots,K-1$. The set of constraints comes from the q-enumeration plus greedy algorithm.

The q-enumeration plus greedy algorithm $G(q)$

The algorithm has two parts.

(i) *q-enumeration*. Produce a list of all subsets S_h^q of N of cardinality q, $h=1,\dots,\binom{n}{q}$.

(ii) *greedy*. Do for all sets of the list

Initialization. Let S_h^q be the set chosen from the list, $N_h^q=N-S_h^q$ and $t=q+1$.

Iteration t. Let $\rho_j(S)=z(S\cup\{j\})-z(S)$. Select $i(t)\in N_h^{t-1}$ for which

$$\rho_{i(t)}(S_h^{t-1})=\max_{i\in N_h^{t-1}}\rho_i(S_h^{t-1})$$

with ties settled arbitrarily. Set $\rho_{t-1}=\rho_{i(t)}(S_h^{t-1})$. Set $S_h^t=S_h^{t-1}\cup\{i(t)\}$ and $N_h^t=N_h^{t-1}-\{i(t)\}$. If $t=K$ stop with the set S_h^K; otherwise set $t\leftarrow t+1$ and continue.

Solution. Output the best solution found in the $\binom{n}{q}$ passes i.e.,

$$Z^{G(q)}=\max_{h=1,\dots,\binom{n}{q}} z(S_h^K).$$

Let $F(q)=\{S_h^t\colon t=q,\dots,K-1;\ h=1,\dots,\binom{n}{q}\}$ and consider the relaxation of (2.1) given by

$$\begin{aligned}\eta^{G(q)}=\max\quad &\eta,\\ &\eta\leqslant z(S)+\sum_{j\in N-S}\rho_j(S)y_j,\quad S\in F(q),\\ &\sum_{j\in N}y_j=K,\\ &y_j\in\{0,1\},\quad j\in N.\end{aligned}\tag{3.1}$$

Note that for fixed K problem (3.1) has $O(n^q)$ constraints, each of which is constructed from $O(n)$ values of the function z. Thus the formulation of (3.1) requires $O(n^{q+1})$ values of z.

Theorem 2. *For any solution of the q-enumeration plus greedy algorithm to $\mathcal{P}_K$,*

$$\frac{Z^{G(q)}}{\eta^{G(q)}}\geqslant 1-\left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}.$$

Proof. Let $(\eta^{G(q)}, y^T)$ be an optimal solution to (3.1). Applying q steps of the greedy algorithm to the set T, we obtain a set $S_{h*}^q=\{i_1,\dots,i_q\}$ for which the submodularity of z implies

$$\rho_j(S_{h*}^q)\leqslant z(S_{h*}^q)/q,\quad \forall j\in T-S_{h*}^q.\tag{3.2}$$

Now as $S^q_{h^*} \in F(q)$ and $(\eta^{G(q)}, y^T)$ is feasible to (3.1), it follows that

$$\begin{aligned}\eta^{G(q)} &\leq z(S^q_{h^*}) + \sum_{j\in N-S^q_{h^*}} \rho_j(S^q_{h^*})y^T_j \\ &\leq z(S^q_{h^*}) + (K-q)\max_{j\in T-S^q_{h^*}} \rho_j(S^q_{h^*}) \\ &\leq z(S^q_{h^*}) + \frac{K-q}{q} z(S^q_{h^*}) \\ &= \frac{K}{q} z(S^q_{h^*})\end{aligned} \tag{3.3}$$

where the middle inequality follows from $|T-S^q_{h^*}| = K-q$ and the last one from (3.2).

Consider the $K-q$ constraints of (3.1)

$$\eta \leq z(S^t_{h^*}) + \sum_{j\in N-S^t_{h^*}} \rho_j(S^t_{h^*})y_j, \quad t=q,\ldots,K-1.$$

Let $\rho_{q+i} = z(S^{q+i+1}_{h^*}) - z(S^{q+i}_{h^*})$, $i=0,\ldots,K-q-1$. Since $(\eta^{G(q)}, y^T)$ is a feasible solution to (3.1) we obtain

$$\begin{aligned}\eta^{G(q)} &\leq z(S^t_{h^*}) + \sum_{j\in N-S^t_{h^*}} \rho_j(S^t_{h^*})y^T_j \\ &= z(S^q_{h^*}) + \sum_{i=0}^{t-q-1} \rho_{q+i} + \sum_{j\in T-S^t_{h^*}} \rho_j(S^t_{h^*}) \\ &\leq z(S^q_{h^*}) + \sum_{i=0}^{t-q-1} \rho_{q+i} + (K-q)\rho_t, \quad t=q,\ldots,K-1\end{aligned}$$

where the last inequality follows from $\rho_t \geq \rho_j(S^t_{h^*})$ by the greedy algorithm and $|T-S^t_{h^*}| \leq K-q$.

With $\eta^{G(q)} - z(S^q_{h^*})$ normalized to one, the problem of minimizing

$$\frac{z(S^K_{h^*}) - z(S^q_{h^*})}{\eta^{G(q)} - z(S^q_{h^*})} = \sum_{i=0}^{K-1-q} \rho_{q+i}$$

subject to

$$\eta^{G(q)} \leq z(S^q_{h^*}) + \sum_{i=0}^{s-1} \rho_{q+i} + (K-q)\rho_{q+s}, \quad s=0,\ldots,K-q-1$$

is a linear program with variables $(\rho_q,\ldots,\rho_{K-1})$. Using the weak duality theorem of linear programming we obtain a lower bound on its value given by [17, Theorem 4.1]

$$\frac{z(S^K_{h^*}) - z(S^q_{h^*})}{\eta^{G(q)} - z(S^q_{h^*})} \geq 1 - \left(\frac{K-q-1}{K-q}\right)^{K-q}. \tag{3.4}$$

Finally using $Z^{G(q)} \geqslant z(S_{h^*}^K)$ and $\eta^{G(q)} \leqslant (K/q) z(S_{h^*}^q)$ from (3.3) in (3.4), we obtain

$$\eta^{G(q)} - Z^{G(q)} \leqslant \left(\frac{K-q-1}{K-q}\right)^{K-q} \left(\frac{K-q}{K}\right) \eta^{G(q)}.$$

An important observation about the set $F(q)$ is that it contains subsets of cardinality less than K. Since $\rho_j(S) \geqslant 0$, the algorithm will always produce subsets of size K, even if we replace $\sum_{j \in N} y_j = K$ by $\sum_{j \in N} y_j \leqslant K$. However the small subsets are needed to obtain the bound of Theorem 2.

Let Z be the value of an optimal solution to $\mathcal{P}_K$. Since $Z^{G(q)} \leqslant Z \leqslant \eta^{G(q)}$ we obtain

Corollary 1. *For any solution of the q-enumeration plus greedy algorithm to* $\mathcal{P}_K$,

$$\frac{Z^{G(q)}}{Z} \geqslant 1 - \left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}.$$

Corollary 2

$$\frac{Z}{\eta^{G(q)}} \geqslant 1 - \left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}.$$

Note that Corollary 1 is the result (1.8) given in the introduction. Our result on best algorithms [16], which states that the bound of (1.8) cannot be greater for any heuristic requiring $O(n^{q+1})$ values of z, also applies to the result of Theorem 2. In particular, suppose we initialize Algorithm 1 by a collection of constraints generated from $O(n^r)$ values of z. Then if we obtain the bound

$$\frac{\max_{S \in Q^t} z(S) \colon |S| = K}{\eta^t} > 1 - \left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}$$

for all nondecreasing submodular functions, it follows that $r \geqslant q+2$.

For the K-location problem Algorithm 1 is precisely Benders' algorithm and thus Theorem 2 suggests how we might initialize Benders' algorithm. Although our result on best algorithms for ($\mathcal{P}_k$) does not apply to the subclass of K-location problems, we do not know of any better initialization in terms of a guarantee on the ratio of the lower to the upper bound generated by Benders' algorithm. Furthermore, the results in [4] imply that, in the worst case, Benders' algorithm will require $O(n^K)$ values of the function v to verify optimality.

To solve the integer program (3.1), we might first solve its linear programming relaxation. For $q=0$ we obtain the linear program

$$\begin{aligned} Z^{\mathrm{LP}(G)} = \max \quad & \eta, \\ & \eta \leq z(S^t) + \sum_{j \in N - S^t} \rho_j(S^t) y_j, \quad t = 0, \ldots, K-1, \\ & \sum_{j \in N} y_j = K, \\ & 0 \leq y_j \leq 1, \quad j \in N \end{aligned}$$

where $\{S^t\}_{t=0}^{K-1}$ are the sets generated by the greedy algorithm. Since $Z^{\mathrm{LP}(G)} \geq \eta^{G(0)}$ a stronger version of Theorem 2 for $q=0$ is:

Theorem 3. *For any solution of the greedy algorithm to $\mathcal{P}_K$,*

$$\frac{Z^{G(0)}}{Z^{\mathrm{LP}(G)}} \geq 1 - \left(\frac{K-1}{K}\right)^K.$$

Proof. Since $\rho_t = z(S^t) - z(S^{t-1}) \geq \rho_j(S^t) \geq 0$ for all $j \in N - S^t$, we have that $\sum_{j \in N - S^t} \rho_j(S^t) y_j \leq K\rho_t$ for any (real) y satisfying $\sum_{j \in N} y_j = K$. Now from

$$Z^{\mathrm{LP}(G)} \leq z(S^t) + \sum_{j \in N - S^t} \rho_j(S^t) y_j,$$

we obtain $Z^{\mathrm{LP}(G)} \leq \sum_{i=0}^{t-1} \rho_i + K\rho_t$, $t = 0, \ldots, K-1$. The result then follows from the linear programming arguments sketched in the proof of Theorem 2.

For some very special cases, including the K-location problem, a result similar to Theorem 3 has been given in Section 6 of [17].

4. A branch-and-bound algorithm for $\mathcal{P}_K$

Algorithm 1 requires the solution of a sequence of integer programs. Here we propose a branch-and-bound algorithm that uses linear programming relaxations. The basic idea is to enumerate implicitly all subsets of cardinality K. For a given subset of cardinality $k < K$, we use the greedy algorithm to augment it by $K-k$ elements and thus determine a lower bound on the value of all solutions containing the given k-element subset. We also use the $K-k$ elements generated by the greedy algorithm to construct an integer programming relaxation of the formulation of $\mathcal{P}_K$ given in (2.1). We get an upper bound on the value of all solutions containing the k-element subset by solving the linear program obtained by dropping the integrality conditions in the integer program.

A node at level k of the enumeration tree corresponds to a partial solution in which a subset S of cardinality k has been selected, $N^S \subseteq N - S$ remains and $N - S - N^S$ has been discarded. We denote such a partial solution (or node) by the pair (S, N^S).

Algorithm 2

Initialization. The list conists only of the node (ϕ, N). The upper bound for this node is $\bar{Z}^\phi = \infty$. The problem lower bound is $\underline{Z} = 0$. The incumbent is $I = \phi$.

General Iteration

Step 1. If the list is empty, stop; the incumbent is optimal. Otherwise remove a node (S, N^S) from the top of the list.

Step 2. If $\bar{Z}^S \leq \underline{Z}$ return to Step 1. Otherwise let $k = |S|$. Compute a greedy solution consisting of S plus $K - k$ elements from N^S. Let $\underline{Z}^S$ denote its value. Suppose the greedy algorithm generates the sets $(S = S^k, \ldots, S^K)$. Compute the linear programming upper bound

$$\begin{aligned} \max \quad & \eta, \\ & \eta \leq z(S^t) + \sum_{j \in N^S - S^t} \rho_j(S^t) y_j, \quad t = k, \ldots, K-1, \\ & \sum_{j \in N^S} y_j = K - k, \\ & 0 \leq y_j \leq 1, \quad j \in N^S. \end{aligned} \tag{4.1}$$

Set $\bar{Z}^S = \max \eta$.

Step 3. If $\underline{Z}^S > \underline{Z}$, then $\underline{Z} \leftarrow \underline{Z}^S$ and $I \leftarrow S^K$. If $\underline{Z}^S = \bar{Z}^S$ or if $\bar{Z}^S \leq \underline{Z}$ go to Step 1, otherwise go to Step 4.

Step 4. Suppose $N^S = \{j_1, \ldots, j_r\}$ and $\rho_{j_1}(S) \geq \cdots \geq \rho_{j_r}(S)$. Add the nodes $(S \cup \{j_t\}, N^S - \{j_1, \ldots, j_t\})$ to the bottom of the list unless

$$|(S \cup \{j_t\}) \cup (N^S - \{j_1, \ldots, j_t\})| < K, \quad t = 1, \ldots, r.$$

Assign each of these nodes an upper bound of $\bar{Z}^S$. Go to Step 1.

An example of Algorithm 2 is given in the Appendix.

Note that after any iteration of the algorithm all nodes on the list are such that $|S| = k$ or $k+1$ for some $k < K$ and that all nodes with $|S| = k$ are removed before any nodes with $|S| = k+1$.

We now consider the situation in which we have completed Step 3 for the last node corresponding to $|S| = q$. At this point we have, for fixed K, evaluated the function z $O(n^{q+1})$ times. Also $\underline{Z} = \max_{|S|=q} \underline{Z}^S = Z^{G(q)}$, from the implicit application of the q-enumeration plus greedy algorithm. Define $\bar{Z}^{LP(q)} = \max_{|S|=q} \bar{Z}^S$, where $\bar{Z}^S$ is the linear programming bound computed from (4.1).

Theorem 4

$$\frac{Z^{G(q)}}{\bar{Z}^{LP(q)}} \geq 1-\left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}.$$

Proof. Theorem 3 implies that

$$\frac{\underline{Z}^S - z(S)}{\bar{Z}^S - z(S)} \geq 1-\left(\frac{K-q-1}{K-q}\right)^{K-q}$$

for each node (S, N^S) with $|S| = q$. Alternatively,

$$\bar{Z}^S - \underline{Z}^S \leq \left(\frac{K-q-1}{k-q}\right)^{K-q}(\bar{Z}^S - z(S)). \tag{4.2}$$

From the first inequality of (4.1),

$$\eta \leq z(S) + \sum_{j\in N^S} \rho_j(S)y_j \quad \text{and} \quad \sum_{j\in N^S} y_j = K-q,$$

we have that

$$\bar{Z}^S \leq z(S) + (K-q)\max_{j\in N^S} \rho_j(S).$$

From the construction of nodes in Step 4 of the algorithm we have that $\max_{j\in N^S} \rho_j(S) \leq z(S)/q$. Hence

$$\bar{Z}^S \leq z(S) + \frac{K-q}{q}z(S) = \frac{K}{q}z(S). \tag{4.3}$$

Using (4.3) and $Z^{G(q)} \geq \underline{Z}^S$ in (4.2) yields

$$\frac{Z^{G(q)}}{\bar{Z}^S} \geq 1-\left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}. \tag{4.4}$$

The theorem follows since (4.4) holds for every node (S, N^S) with $|S| = q$.

Theorem 4 shows that Algorithm 2 has properties similar to those of Algorithm 1. For fixed K and with $O(n^{q+1})$ function evaluations, Algorithm 2 can be run down to level q, and we can guarantee

$$\frac{\text{value of best feasible solution}}{\text{linear programming upper bound}} \geq 1-\left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}.$$

As before, our result of [16] implies that no better performance guarantee can be obtained with $O(n^{q+1})$ values of z.

Theorem 4 is based on implicit enumeration of the sets S with $|S| \leq q$, which is the reason for the particular subproblem selection rule of the algorithm. Its proof requires $\max_{j\in N^S} \rho_j(S) \le z(S)/q$, which is the reason for the branching rule of Step 4. Without these subproblem selection and branching rules, the bound cannot be guaranteed.

We note that the linear programming problem (4.1) at node (S, N^S) does not necessarily contain any of the inequalities from previously solved subproblems. Using these additional inequalities can improve the bounds for particular problems but not the worst case bound of Theorem 4.

It is possible to do a similar analysis for the more conventional binary tree in which the two sons of a node (S, N^S) correspond to selecting and discarding a $j^* \in N^S$ such that $\rho_{j^*}(S) = \max_{j\in N^S} \rho_j(S)$. It is still, however, necessary to choose nodes with S of minimum cardinality first to guarantee the bound of Theorem 4. In the binary tree, this approach requires $O(n^{q+2})$ function values in the worst case.

When Algorithm 2 is applied to the K-location problem, at node (S, N^S) we could solve the linear program

$$\begin{aligned}
\bar{V}^S = \max \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}, \\
& \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, m, \\
& \sum_{j \in N^S} y_j = K - |S|, \\
& x_{ij} - y_j \le 0, \quad i = 1, \ldots, m, \quad j \in N^S, \\
& x_{ij} = 0, \quad i = 1, \ldots, m, \quad j \in N - N^S - S, \\
& x_{ij} \ge 0, \quad i = 1, \ldots, m, \quad j \in S \cup N^S, \\
& y_j \ge 0, \quad j \in N^S.
\end{aligned} \tag{4.5}$$

For every node (S, N^S), (4.1) is a relaxation of (4.5). This follows from the fact that (4.1) is obtained from a Benders' decomposition of (4.5) using only those constraints generated from a subset of the dual extreme points of (4.5). Therefore

$$\bar{V}^S \le \bar{Z}^S \quad \text{and} \quad \bar{V}^{\mathrm{LP}(q)} = \max_{|S|=q} \bar{V}^S \le \bar{Z}^{\mathrm{LP}(q)}$$

so that Theorem 4 yields:

Theorem 5. *For the K-location problem* (1.2),

$$\frac{V^{G(q)}}{\bar{V}^{\mathrm{LP}(q)}} \ge 1 - \left(\frac{K-q}{K}\right)\left(\frac{K-q-1}{K-q}\right)^{K-q}.$$

5. Maximizing submodular functions subject to linear constraints

Many submodular functions that arise in practice are not nondecreasing and many problems have more complicated feasibility conditions than the cardinality constraint that we have supposed so far. Here we indicate how the formulations and algorithms presented previously can be adapted to the maximization of general submodular functions subject to linear constraints.

Proposition 2 [17]. *A real-valued function z on the subsets of N is submodular if and only if*

$$\text{(a)}\qquad z(T) \leq z(S) + \sum_{j \in T-S} \rho_j(S) - \sum_{j \in S-T} \rho_j(S \cup T - \{j\}) \qquad \forall S, T \subseteq N$$

or

$$\text{(b)}\qquad z(T) \leq z(S) + \sum_{j \in T-S} \rho_j(S \cap T) - \sum_{j \in S-T} \rho_j(S - \{j\}) \qquad \forall S, T \subseteq N.$$

Consider the problem

$$\operatorname*{Max}_{S \in \mathcal{S}}(z(S)\text{: } z \text{ submodular, } \mathcal{S} = \{S\text{: } Ay^S = b,\ y_j \in \{0, 1\},\ j \in N\}). \tag{5.1}$$

Let U be an optimal solution to (5.1) and let $g_S + h_S y$ represent one of the linear forms

$$\text{(a)}\qquad z(S) + \sum_{j \in N-S} \rho_j(S) y_j - \sum_{j \in S} \alpha_j (1 - y_j)$$

where for $j \in S$, $\alpha_j \leq \rho_j(S \cup U - \{j\})$, e.g., $\alpha_j = \rho_j(N - \{j\})$, or

$$\text{(b)}\qquad z(S) + \sum_{j \in N-S} \gamma_j y_j - \sum_{j \in S} \rho_j(S - \{j\})(1 - y_j)$$

where for $j \in N - S$, $\gamma_j \geq \rho_j(S \cap U - \{j\})$, e.g., $\gamma_j = \rho_j(\emptyset)$.

Now consider the integer linear program

$$\begin{aligned} \max\quad & \eta, \\ & \eta \leq g_S + h_S y, \quad \forall S \subseteq N, \\ & Ay = b, \\ & y_j \in \{0, 1\},\ j \in N, \end{aligned} \tag{5.2}$$

where $g_S + h_S y$ is one of the linear forms (a) or (b).

Precisely as in Section 2, we see that problem (5.2) is a reformulation of problem (5.1) since:

Theorem 6. $(z(U), y^U)$ *is an optimal solution to* (5.2) *if and only if* U *is an optimal solution to* (5.1).

Algorithms 1 and 2 can be adapted to problem (5.2). This is straightforward for Algorithm 1 and Theorem 2 suggests the use of the greedy algorithm in selecting the intial constraints. When $Ay = b$ is the simple cardinality constraint $\sum_{j\in N} y_i = K$, the greedy algorithm can be applied directly and, in fact, a performance guarantee that is related to the one in Theorem 2 (but weaker) can be obtained [17]. For a general set of linear constraints, the greedy algorithm would have to be modified so that maximum improving elements were selected sequentially subject to $Ay = b$. In this case no performance guarantee is known.

We would propose a similar adaptation for Algorithm 2. The constraints for the linear program of Step 2 would be obtained from the sets generated by the modified greedy algorithm just mentioned. The branching rule of Step 4 would be based on decreasing ρ_j's subject to feasibility. This type of algorithm resembles the implementation given in [14] of Benders' decomposition of mixed integer programs since it synthesizes constraint generation, linear programming relaxation and implicit enumeration.

6. Structure, Simplifications and applications

Here we show how simplified linearizations can be obtained when the submodular functions have structure. We apply these ideas to the location problem and the minimization of some quadratic 0–1 programs. (Unless stated otherwise we use constraints of type (a) in (5.2).)

A (nondecreasing) submodular function z on the set N is said to be *separable* if $z \equiv \sum_{i\in I} v^i$, where I is a finite set with $|I| > 1$, and each function v^i is (nondecreasing) submodular on N. Let $\rho_j^i(S) = v^i(S \cup \{j\}) - v^i(S)$.

Consider the formulation

$$\begin{aligned} \max \quad & \sum_{i\in I} \eta^i, \\ & \eta^i \leq v^i(S) + \sum_{j\in N-S} \rho_j^i(S) y_j - \sum \rho_j^i(N-\{j\})(1-y_j), \quad \forall i \in I, \quad \forall S \subseteq N, \qquad (6.1) \\ & Ay = b, \\ & y_j \in \{0,1\}, \quad j \in N. \end{aligned}$$

This is a new formulation of (5.1) as:

Theorem 7. $(z(U), y^U)$ *is an optimal solution to* (6.1) *if and only if* U *is an optimal solution of* (5.1).

Proof. As in Lemma 1, (ξ^i, y^U) is feasible in (6.1) if and only if y^U is feasible and $\xi^i \leq v^i(U)$, $\forall i \in I$.

The value of separability evidently depends to some extent on whether the resulting functions v^i have a simpler structure than z. We now examine two cases, where the majority of the inequalities in formulation (5.2) are redundant.

Let $\{c_j\}_{j\in N}$, $N=\{1,\ldots,n\}$ be a set of nonnegative numbers and consider the submodular function $z^*(S)=\max_{j\in S} c_j$, $\forall S\subseteq N$. Note that $v(S)$, the function associated with the K-location problem, is a sum of such functions. Define $c_0\equiv 0$ and suppose, without loss of generality, that $c_n \geq c_{n-1} \geq \cdots \geq c_1 \geq c_0$. Let $\chi^+ = \max(0, \chi)$.

Theorem 8. *For the function* z^*, *problem* (5.2) *can be formulated as*

$$\begin{aligned}
\max\quad & \eta, \\
& \eta \leq c_r + \sum_{j\in N} (c_j - c_r)^+ y_j, \quad r=0,\ldots,n-1, \\
& Ay = b, \\
& y_j \in \{0,1\}, \quad j \in N.
\end{aligned} \tag{6.2}$$

Proof. Note first that the above formulation is a relaxation of (5.2), as the inequalities in (6.2) can be obtained from the sets $S_0=\{\emptyset\}$, and $S_i=\{i\}$, $i=1,\ldots,n-1$ with $\alpha_j=0$. Hence it suffices to show that (ξ, y^U) is infeasible if $\xi > z^*(U)$. Let $c_r = \max_{j\in U}\{c_j\}$. From the rth inequality we obtain

$$\xi \leq c_{r-1} + \sum_{j\in N} (c_j - c_{r-1})^+ y_j^U = c_{r-1} + (c_r - c_{r-1}) = c_r = z^*(U).$$

As an immediate consequence of Theorem 8, letting $c_{i_n} \geq c_{i_{n-1}} \geq \cdots \geq c_{i_1} \geq c_{i_0} \equiv 0$ be an ordering of $\{c_{ij}\}$, $j \in N$, we obtain the formulation of the K-location problem (1.2) given by

$$\begin{aligned}
\max\quad & \sum_{i=1}^{m} \eta^i, \\
& \eta^i \leq c_{i_r} + \sum_{j\in N} (c_{ij} - c_{i_r})^+ y_j, \quad i=1,\ldots,m, \quad r=0,\ldots,n-1, \\
& \sum_{j\in N} y_j = K, \\
& y_j \in \{0,1\}, \quad j \in N.
\end{aligned}$$

This formulation requires only mn linear inequalities to represent $v(S)$. Again we note that an appropriate application of Benders' procedure also leads to this formulation. See [5] for alternative reformulations.

A different simplification may occur by expressing the set function as a polynomial in 0–1 variables. Define $g_\emptyset = z(\emptyset)$ and recursively $g_S = z(S) - \sum_{T \subset S} g_T$. Let $f(y) = \sum_{T \subseteq N} g_T \prod_{i \in T} y_i$ so that $z(S) = f(y^S)$.

A representation of f in the form:

$$f(y) = \sum_{T \subseteq T^*} \prod_{i \in T} y_i \Big(d_0^T + \sum_{j \in N} d_j^T y_j\Big)$$

will be called a *polynomial linearization* of z. Clearly every set function has a trivial polynomial linearization with $T^* = N$, $d_0^T = g_T$, $d_j^T = 0$, $\forall j \in N$. We are interested in cases where $|T^*| < n$ since:

Theorem 9. *If z is submodular and has a polynomial linearization, the following integer program is a reformulation of* (5.1):

$$\begin{aligned} \max \quad & \eta, \\ & \eta \leq z(S) + \sum_{j \in N-S} \rho_j(S) y_j - \sum_{j \in S} \alpha_j (1 - y_j), \quad \forall S \subseteq T^*, \\ & Ay = b, \\ & y_j \in \{0, 1\}, \quad j \in N. \end{aligned} \tag{6.3}$$

Proof. From Lemma 1 it suffices to show that (ξ, y^U) is infeasible in (6.3) if $\xi > z(U)$. Let $S = U \cap T^*$, and consider the corresponding inequality. We obtain

$$\begin{aligned} \xi &\leq z(S) + \sum_{j \in U-S} \rho_j(S) \\ &= \sum_{T \subseteq S} \Big(d_0^T + \sum_{j \in S} d_j^T\Big) + \sum_{j \in U-S} \sum_{T \subseteq S} d_j^T \qquad (\text{as } (U-S) \cap T^* = \emptyset) \\ &= \sum_{T \subseteq S} \Big(d_0^T + \sum_{j \in S} d_j^T + \sum_{j \in U-S} d_j^T\Big) \\ &= \sum_{T \subseteq U \cap T^*} \Big(d_0^T + \sum_{j \in U} d_j^T\Big) = z(U). \end{aligned}$$

As an example we consider the quadratic 0–1 programming problem

$$\min(cy + y'Qy \colon Ay = b,\ y_j \in \{0, 1\},\ j \in N) \tag{6.4}$$

where $q_{ii} = 0$, $q_{ij} = q_{ji} \geq 0$, $i \neq j$. This example combines the use of separability, polynomial linearization, and the choice of alternative inequalities of type (b). Let

$$m_j = \min\Big\{\sum_{i:\, i \neq j} q_{ij} y_i \colon Ay = b,\ y_i \in \{0, 1\},\ i \in N\Big\},$$

and

$$M_j = \max\left\{\sum_{i:\, i\neq j} q_{ij}y_i:\ Ay=b,\ y_i\in\{0,1\},\ i\in N\right\}.$$

Now consider the problem

$$\begin{aligned} \min\quad & \sum_{j\in N}[(c_j+m_j)y_j+\lambda_j],\\ & \lambda_j \geqslant -M_j+\sum_{i:\, i\neq j} q_{ij}y_i+(M_j-m_j)y_j,\quad j\in N,\\ & Ay=b,\\ & y_j\in\{0,1\},\quad \lambda_j\geqslant 0,\quad j\in N. \end{aligned} \tag{6.5}$$

Theorem 10. *y^U is an optimal solution to problem* (6.4) *if and only if* (y^U, λ^U) *is an optimal solution to* (6.5), *where*

$$\lambda_j^U = \left(-M_j+\sum_{i:\, i\neq j} q_{ij}y_i^U+(M_j-m_j)y_j^U\right)^+.$$

Proof. Consider the objective $\max(-cy-y'Qy)$. Let $f(y)=-y'Qy$. With the given conditions on the $\{q_{ij}\}$, f is submodular, [17]. Taking $f(y)=\sum_{j=1}^n f^j(y)$ with $f^j(y)=y_j(-\sum_{i\neq j} q_{ij}y_i)$, we see that f is separable. Note that f^j has a polynomial linearization with $T^*=\{j\}$. As $|T^*|=1$ Theorem 9 implies that precisely two inequalities are required to linearize f^j. Taking $S=\emptyset$ we obtain the inequality $\eta^j\leqslant 0$. Taking $S=\{j\}$ we obtain $\eta^j\leqslant -\sum_{i\neq j} q_{ij}y_i-\alpha_j(1-y_j)$ where $\alpha_j\leqslant \rho_j(U-\{j\})$ and U is an optimal solution. Noting that $\rho_j(U-\{j\})=-\sum_{i\neq j} q_{ij}y_i^U$, we can take $\alpha_j=-M_j$, and the second inequality becomes

$$\eta^j\leqslant -\sum_{i\neq j} q_{ij}y_i+M_j(1-y_j).$$

Now note that inequality 5.2(b) applied to $f^j(y)$ with $S=N-\{j\}$ yields $\eta^j\leqslant \gamma_j y_j$, where $\gamma_j\geqslant \rho_j(U-\{j\})$. Since $-m_j\geqslant \rho_j(U-\{j\})$ and $m_j\geqslant 0$, the inequality $\eta^j\leqslant -m_jy_j$ dominates $\eta^j\leqslant 0$ and can be used in its place.

Hence we obtain the reformulation

$$\begin{aligned} \max\quad & -cy+\sum_{j\in N}\eta^j,\\ & \eta^j\leqslant -m_jy_j,\quad j\in N,\\ & \eta^j\leqslant -\sum_{i:\, i\neq j} q_{ij}y_i+M_j(1-y_j),\quad j\in N,\\ & Ay=b,\\ & y_j\in\{0,1\},\quad j\in N. \end{aligned}$$

Now setting $\lambda_j=-m_jy_j-\eta^j$, $j\in N$, the result follows.

This formulation can also be obtained by the methods in [9].

7. Conclusion

We have indicated how the structure of submodularity arises in combinatorial optimization problems, how it may be used in the design of algorithms (both approximate and exact) and in the generation of problem reformulations. Our point of view has been to use results on the behavior of heuristics to design exact algorithms. From intermediate stages of these algorithms we obtain performance guarantees on the optimal solution.

Other well-known problems to which these models are applicable include a capacitated location problem [7], multiproduct, and even multi-level distribution systems, see for example [7, 8] where Benders' algorithm was successfully used, and the quadratic assignment problem [1, 11, 13], a special case of the quadratic 0–1 program. Furthermore the formulations that we have given here do not exhaust the possibilities since other linearizations of submodular functions can be constructed as well.

Appendix

We consider the K-location problem (1.1) with $K=3$ and

$$C=\begin{bmatrix} 2 & 2 & 0 & 0 & 0 & 1 \\ 2 & 0 & 2 & 0 & 0 & 1 \\ 2 & 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 & 0 & 1 \\ 0 & 0 & 2 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 4 & 2 \end{bmatrix}.$$

Applying the greedy algorithm to this problem, with ties broken by choosing the lowest index element, yields the subsets $S^0=\emptyset$, $S^1=\{6\}$, $S^2=\{6,1\}$, $S^3=\{6,1,2\}$ and $Z^{G(0)}=13$.

Algorithm 1

Initialization. Choose the sets S^0, S^1, and S^2 given by the greedy algorithm to obtain the relaxed problem (see 3.1):

$$\begin{aligned} \eta^{G(0)} = \max \quad & \eta \\ & \eta \leq 0+6y_1+6y_2+6y_3+6y_4+4y_5+8y_6, \\ & \eta \leq 8+3y_1+3y_2+3y_3+3y_4+2y_5, \\ & \eta \leq 11 \qquad +2y_2+2y_3+2y_4+2y_5, \\ & \sum_{j=1}^{6} y_j = 3, \\ & y_j \in \{0,1\}, \quad j=1,\dots,6. \end{aligned} \tag{A.1}$$

The solution is $y_2 = y_3 = y_4 = 1$, $y_j = 0$, otherwise and $\eta^{G(0)} = 17$. Note that

$$\frac{Z^{G(0)}}{\eta^{G(0)}} = \frac{13}{17} \geq 1 - \left(\frac{2}{3}\right)^3 = \frac{19}{27},$$

which is the bound given by Theorem 2.

Iterative Phase. $R^3 = \{2, 3, 4\}$, $z(R^3) = 12$. Add the constraint

$$\eta \leq 12 + 4y_5 + 2y_6$$

to (A.1) and re-solve. There are three optimal solutions, $y_2 = y_3 = y_5 = 1$, $y_j = 0$ otherwise, $y_2 = y_4 = y_5 = 1$, $y_j = 0$ otherwise, $y_3 = y_4 = y_5 = 1$, $y_j = 0$ otherwise, and $\eta^4 = 16$. Also $z(R^4) = 14$. The constraints generated by these solutions are respectively

$$\eta \leq 14 + 2y_1 + 2y_4 + y_6,$$
$$\eta \leq 14 + 2y_1 + 2y_3 + y_6;$$
$$\eta \leq 14 + 2y_1 + 2y_2 + y_6.$$

These constraints are added successively in the next 3 iterations. We then obtain the optimal solutions $y_1 = y_2 = y_5 = 1$, $y_j = 0$ otherwise, $y_1 = y_3 = y_5 = 1$, $y_j = 0$ otherwise, $y_1 = y_4 = y_5 = 1$, $y_j = 0$ otherwise and $\eta^7 = 15$. Also $z(R^7) = 14$. The constraints generated by these solutions are respectively

$$\eta \leq 14 + 2y_3 + 2y_4 + y_6,$$
$$\eta \leq 14 + 2y_2 + 2y_4 + y_6,$$
$$\eta \leq 14 + 2y_2 + 2y_3 + y_6.$$

These constraints are added successively in the next three iterations and we finally obtain $\eta^{10} = 14$, which verifies the optimality of selecting any two of the first four columns and the fifth; i.e., the solutions produced in the last six iterations of the algorithm.

As we noted in Section 3, Benders' algorithm gives results identical to the ones we have just given.

Algorithm 2

Initialization. The list consists only of $(\emptyset, N)$. $\underline{Z} = 0$, $I = \emptyset$.

Iterative Phase. We select $(\emptyset, N)$ from the list, apply the greedy algorithm and obtain $\underline{Z}^{\emptyset} = 13$. Next the linear programming relaxation of (A.1) is solved and we obtain $\bar{Z}^{\emptyset} = 17$. Thus $\underline{Z} = 13$ and $I = \{1, 2, 6\}$.

We have $\rho_6(\emptyset) > \rho_1(\emptyset) = \rho_2(\emptyset) = \rho_3(\emptyset) = \rho_4(\emptyset) > \rho_5(\emptyset)$. The list consists of $(\{6\}, N - \{6\})$, $(\{1\}, N - \{1, 6\})$, $(\{2\}, \{3, 4, 5\})$, $(\{3\}, \{4, 5\})$. Each of these nodes has $\bar{Z}^S = 17$.

Select $(\{6\}, N - \{6\})$ from the list. Set $k = 1$ and compute the greedy solution

$\{6, 1, 2\}$ of value $\underline{Z}^{\{6\}} = 13$. Solve the linear program (see 4.1):

$$\begin{aligned} \max \quad & \eta, \\ & \eta \leq 8 + 3y_1 + 3y_2 + 3y_3 + 3y_4 + 2y_5, \\ & \eta \leq 11 + \phantom{3y_1 +{}} 2y_2 + 2y_3 + 2y_4 + 2y_5, \\ & \sum_{j=1}^{5} y_j = 2, \\ & 0 \leq y_j \leq 1, \quad j = 1, \ldots, 5. \end{aligned}$$

An optimal solution is $\eta = 14$, $y_2 = y_3 = 1$, $y_j = 0$ otherwise so that $\bar{Z}^{\{6\}} = 14$. $\rho_1(\{6\}) = \rho_2(\{6\}) = \rho_3(\{6\}) = \rho_4(\{6\}) > \rho_5(\{6\})$. We add $(\{1, 6\}, N - \{1, 6\})$, $(\{2, 6\}, \{3, 4, 5\})$, $(\{3, 6\}, \{4, 5\})$ $(\{4, 6\}, \{5\})$ to the list. Each of these nodes has $\bar{Z}^S = 14$.

Select $(\{1\}, N - \{1, 6\})$ from the list. Set $k = 1$ and compute the greedy solution $\{1, 2, 5\}$ of value $\underline{Z}^{\{1\}} = 14$. Solve the linear program

$$\begin{aligned} \max \quad & \eta, \\ & \eta \leq 6 + 4y_2 + 4y_3 + 4y_4 + 4y_5, \\ & \eta \leq 10 + + 2y_3 + 2y_4 + 4y_5, \\ & \sum_{j=2}^{5} y_j = 2, \\ & 0 \leq y_j \leq 1, \quad j = 2, \ldots, 5. \end{aligned}$$

An optimal solution is $\eta = 14$, $y_3 = y_4 = 1$, $y_j = 0$ otherwise so that $\underline{Z}^{\{1\}} = 14$. We update the incumbent to $I = \{1, 2, 5\}$ and $\underline{Z} = 14$.

The problems generated from the nodes $(\{2\}, \{3, 4, 5\})$ and $(\{3\}, \{4, 5\})$ also give lower and upper bounds of 14. The remaining problems on the list have upper bounds of 14 so the problem is solved.

Since our problem is a K-location problem, we could have solved (4.5) instead of (4.1) for each node of the list. For the node $(\emptyset, N)$ we obtain $y_1 = y_2 = \frac{1}{2}$, $y_3 = y_5 = 1$, $y_4 = y_6 = 0$ and $\bar{V}^{\emptyset} = 16$, which is better than the bound obtained from (4.1). However, for this example the same amount of enumeration is required.

References

[1] E.M.L. Beale, Integer Programming, in: T.B. Boffey, ed., Proceedings of CP 77 (University of Liverpool, 1977) 269–279.

[2] J.F. Benders, Partitioning procedures for solving mixed-variables programming problems, Numer. Math. 4 (1962) 238–252.

[3] G. Cornuéjols, M.L. Fisher and G.L. Nemhauser, Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms, Management Sci. 23 (1977) 789–810.

[4] G. Cornuéjols, G.L. Nemhauser and L.A. Wolsey, Worst-case and probabilistic analysis of algorithms for a location problem, Oper. Res. 28 (1980) 847–858.

[5] G. Cornuéjols, G.L. Nemhauser and L.A. Wolsey, A canonical representation of simple plant locations problems and its applications, SIAM J. Algebraic and Discrete Methods 1 (1980) 261–272.

[6] G. Dantzig, D.R. Fulkerson and S. Johnson, Solution of a large-scale travelling salesman problem, Oper. Res. 2 (1954) 393–410.

[7] A.M. Geoffrion, How can specialized discrete and convex optimization methods be married, in: P.L. Hammer, E.L. Johnson, B.H. Korte and G.L. Nemhauser, eds., Studies in Integer Programming (North-Holland, Amsterdam, 1977) 205–220.

[8] A.M. Geoffrion and G.W. Graves, Multicommodity distribution system design by Benders' decomposition, Management Sci. 20 (1974) 822–844.

[9] F. Glover, Improved linear integer programming formulations of nonlinear integer problems, Management Sci. 22 (1975) 455–460.

[10] M. Guignard and K. Spielberg, Algorithms for exploiting the structure of the simple plant location problem, in: P.L. Hammer, E.L. Johnson, B.H. Korte and G.L. Nemhauser, eds., Studies in Integer programming (North-Holland, Amsterdam, 1977) 247–272.

[11] L. Kaufman and F. Broeckx, An algorithm for the quadratic assignment problem using Benders' decomposition, European J. Oper. Res. 2 (1978) 207–211.

[12] A Land and S. Powell, Computer codes for problems of integer programming, Ann. Discrete Math. 5 (1979) 221–269.

[13] E.L. Lawler, The quadratic assignment problem, Management Sci. 9 (1963) 586–599.

[14] C.E. Lemke and K. Spielberg, Direct search zero–one and mixed integer programming, Oper. Res. 15 (1967) 892–914.

[15] T.L. Magnanti and R.T. Wong, Accelerating Benders' decomposition: Algorithmic enhancements and model selection criteria, CORE Discussion Paper 8003, Louvain-la-Neuve, Belgium (January 1980).

[16] G.L. Nemhauser and L.A. Wolsey, Best algorithms for approximating the maximum of a submodular set function, Math. Oper. Res. 3 (1978) 177–188.

[17] G.L. Nemhauser, L.A. Wolsey and M.L. Fisher, An analysis of approximations for maximizing a submodular set function—I, Math. Programming 14 (1978) 265–294.

[18] H.P. Williams, Experiments in the formulation of integer programming problems, Math. Programming Studies 2 (1974) 180–197.