AD-A110 913 CARNEGIE INST OF TECH PITTSBURGH PA MANAGEMENT SCIEN--ETC F/G 12/1
INTEGER PROGRAMMING.(U)
NOV 81 E BALAS N00014-75-C-0621
UNCLASSIFIED NL

1 OF 1
AD A
110913

END
DATE
FILMED
03-82
DTIC

LEVEL II

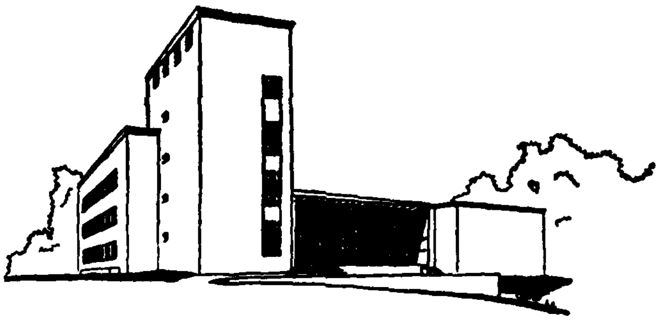
12

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



II

W.P.#16-81-82

Management Sciences Research Report No. 479

12

INTEGER PROGRAMMING

by

Egon Balas

November 1981

The research underlying this report was supported by Grant ECS-7902506 of the National Science Foundation and Contract N00014-75-C-0621 NR 047-048 with the U.S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

Management Science Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

This is an introductory survey of integer programming, its theory, methodology and applications, for the *Encyclopedia of Statistical Sciences*.

INTEGER PROGRAMMING

by

Egon Balas

A linear programming* or nonlinear programming* problem whose variables are constrained to be integer, is called a (linear or nonlinear) integer program. We will consider here only the linear case, although there exist extensions of the techniques to be discussed to nonlinear integer programming.

The integer programming problem can be stated as

$$\text{(P)} \qquad \min\{cx \mid Ax \geq b,\ x \geq 0,\ x_j \text{ integer},\ j \in N_1 \subseteq N\},$$

where A is a given mxn matrix, c and b are given vectors of conformable dimensions, $N = \{1,\ldots,n\}$, and x is a variable n-vector. (P) is called a pure integer program if $N_1 = N$, a mixed integer program if $\emptyset \neq N_1 \neq N$. Integer programming is sometimes called discrete optimization.

Scope and Applicability

Integer programming is the youngest branch of mathematical programming: its development started in the second half of the fifties. It is the most immediate and frequently needed extension of linear programming. Integrality constraints arise naturally whenever fractional values for the decision variables do not make sense. A case in point is the fixed charge problem, in which a function of the form $\sum_i c(x_i)$, with

$$c(x_i) = \begin{cases} f_i + c_i x_i & \text{if } x_i > 0 \\ 0 & \text{if } x_i = 0 \end{cases}$$

is to be minimized subject to linear constraints. Such a problem can be restated as an integer program whenever x is bounded, by setting

$$c(x_i) = c_i x_i + f_i y_i$$
$$x_i \le U_i y_i, \qquad y_i = 0 \text{ or } 1$$

where U_i is an upper bound on x_i.

By far the most important special case of integer programming is the <u>0-1 programming problem</u>, in which the integer-constrained variables are restricted to 0 or 1. This is so because a host of frequently occurring nonlinearities, like logical alternatives, implications, precedence relations, etc., or combinations thereof, can be formulated via 0-1 variables. For example, a condition like

$$x > 0 \Rightarrow \quad (f(x) \le a \;\vee\; f(x) \ge b),$$

where a and b are positive scalars, x is a variable with a known upper bound M, f(x) is a function whose value is bounded from above by U > 0 and from below by L < 0, while the symbol "∨" means disjunction (logical "or"), can be stated as

$$x \le M(1 - \delta_1)$$
$$f(x) \le a + (U - a)\delta_1 + (U - a)\delta_2$$
$$f(x) \ge b + (L - b)\delta_1 + (L - b)(1 - \delta_2)$$
$$\delta_1, \delta_2 = 0 \text{ or } 1.$$

A linear program with "logical" conditions (conjunctions, disjunctions and implications involving inequalities) is called a <u>disjunctive program</u>, since it is the presence of disjunctions that makes these problems nonconvex. Disjunctive programming is coextensive with 0-1 programming.

Nonconvex optimization problems like bimatrix games, separable programs involving piecewise linear nonconvex/nonconcave functions, the general (nonconvex) quadratic programming problem, the linear complementarity problem and many others can be stated as disjunctive or 0-1 programming problems.

A host of interesting combinatorial problems can be formulated as 0-1 programming problems defined on a graph. The joint study of these problems by mathematical programmers and graph theorists has led to the recent development of a burgeoning area of research known as <u>combinatorial optimization</u>. Some typical problems studied in this area are: edge matching and covering, vertex packing and covering, clique covering, vertex coloring; set packing, partitioning and covering; Euler tours; Hamiltonian cycles (traveling salesman problem).

Applications of integer programming abound in all spheres of decision making. Some typical real world problem areas where integer programming is particularly useful as a modeling tool, include: facility (plant, warehouse, hospital, fire station) location; scheduling (of personnel, production, other activities); routing (of trucks, tankers, airplanes); design of communication (road, pipeline, telephone) networks; capital budgeting; project selection; analysis of capital development alternatives. In statistics, integer programming is useful, for instance, in experimental design,* stratified sampling,* cluster analysis.*

Solution Methods: Overview

We denote by v(P), and call the value of (P), the optimal objective function value for (P). We denote by (L), and call the linear programming relaxation of (P), the linear program obtained from (P) by removing the integrality requirements.

Integer programs are notoriously hard: in the language of computational complexity theory, the general 0-1 programming problem, as well as most of its special cases, is NP-complete. Polynomial time integer programming algorithms do not exist. However, sometimes an integer program can be solved as a linear program; i.e., solving the linear programming relaxation (L) of the integer program (P), one obtains an integer solution. In particular, this is the case when all basic solutions of (L) are integer. For an arbitrary integer vector b, the constraint set $Ax \leq b$, $x \geq 0$ is known (Hoffman and Kruskal, 1958) to have only integer basic solutions if and only if the matrix A is totally unimodular (i.e., all nonsingular submatrices of A have a determinant of 1 or -1).

The best known instances of total unimodularity are the vertex-edge incidence matrices of directed graphs and undirected bipartite graphs. As a consequence, shortest path and network flow problems on arbitrary directed graphs, edge matching (or covering) and vertex packing (or covering) problems on bipartite graphs, as well as other integer programs whose constraint set is defined by the incidence matrix of a directed graph or an undirected bipartite graph, with arbitrary integer right hand side, are in fact linear programs.

Apart from this important but very special class of problems, the difficulty in solving integer programs lies in the nonconvexity of the feasible set, which makes it impossible to establish global optimality from local conditions. The two main approaches to solving integer programs try to circumvent this difficulty in two different ways.

The first approach, which in the current state of the art is the standard way of solving integer programs, is enumerative (branch and bound, implicit enumeration). It partitions the feasible set into successively smaller subsets, calculates bounds on the objective function value over each subset, and uses these bounds to discard certain subsets from further consideration. The procedure ends when each subset has either produced a feasible solution, or was shown to contain no better solution than the one already in hand. The best solution found during the procedure is a global optimum. Two early prototypes of this approach are due to Land and Doig (1960) and Balas (1963, English version 1965).

The second approach, known as the cutting plane method, is a convexification procedure: it approximates the convex hull of the set F of feasible integer points, by a sequence of inequalities that cut off (hence the term "cutting planes") parts of the linear programming polyhedron, without removing any point of F. When sufficient inequalities have been generated to cut off every fractional point better than the integer optimum, then the latter is found as an optimal solution to the linear program (L) amended with the cutting planes. The first finitely convergent procedure of this type is due to Gomory (1958).

Depending on the type of techniques used to describe the convex hull of F and generate cutting planes, one can distinguish three main directions in this area. The first one uses algebraic methods, like modular arithmetic and group theory. Its key concept is that of subadditive functions. It is sometimes called the algebraic or group theoretic approach. The second one uses convexity, polarity, propositional calculus. Its main thrust comes from looking at the 0-1 programming problem as a disjunctive program. It is known as the convex analysis/disjunctive programming approach. Finally, the third direction applies to combinatorial programming problems, and it combines graph theory and matroid theory with mathematical programming. It is sometimes called polyhedral combinatorics.

Besides these two basic approaches to integer programming (enumerative and convexifying), two further procedures need to be mentioned, that do not belong to either category, but can rather be viewed as complementary to one or the other. Both procedures essentially <u>decompose</u> (P), one of them by partitioning the variables, the other one by partitioning the constraints. The first one, due to Benders (1962), gets rid of the continuous variables of a mixed integer program (P) by projecting the feasible set F into the subspace of the integer-constrained variables. The second one, known as Lagrangean relaxation, gets rid of some of the constraints of (P) by assigning multipliers to them and taking them into the objective function.

Each of the approaches outlined here aims at solving (P) exactly. However, since finding an optimal solution tends to be expensive beyond a certain problem size, approximation methods or <u>heuristics</u>* play an increasingly important role in this area.

Next we briefly review the approaches sketched above, and give some references for each of them. As general references on integer programming, see the book by Garfinkel and Nemhauser (1971), and the recent volumes edited by Christofides, Mingozzi, Toth and Sandi (1979), Hammer, Johnson and Korte (1979a, b), Padberg (1980).

Branch and Bound/Implicit Enumeration

The following are the basic steps of a typical enumerative algorithm. Start by putting (P) on the list of subproblems, and by setting $\bar{v}(P) = \infty$, where $\bar{v}(P)$ is an upper bound on v(P).

1. Choose, and remove from the list, a subproblem (P_i), according to some criterion specified by the search strategy. If the list is empty, stop: if no solution was found, (P) is infeasible; otherwise the current best solution is optimal.

2. *If* (P_i) *has constraints involving only* 0-1 variables, explore their implications via logical tests to impose as many new constraints of the type $x_i = 0$, or $x_i = 1$ (*or of* a more complex type), as possible. If as a result (P_i) is shown infeasible, discard (P_i) and go to 1.

3. Generate a lower bound $\underline{v}(P_i)$ on $v(P_i)$, by solving some relaxation of (P_i) (like the linear programming relaxation, or a Lagrangean relaxation, or either of these two amended with cutting planes). If $\underline{v}(P_i) \geq \bar{v}(P)$, discard (P_i) and go to 1.

4. Attempt to generate an improved upper bound on v(P) by using some heuristic to find an improved feasible

solution. If successful, update $\bar{v}(P)$ and remove from the list all (P_j) such that $\underline{v}(P_j) \geq \bar{v}(P)$.

5. Split (P_i) into two or more subproblems by partitioning its feasible set according to some specified rule. Add the new subproblems to the list and go to 1.

The search strategies that can be used in step 1 range between the two extremes known as "breadth first" (always choose the most promising subproblem, i.e., the one with smallest $\underline{v}(P_i)$, and "depth first" (always choose one of the new subproblems just created). The first approach carries a high cost in terms of storage requirements, therefore the second one is preferred in most codes. Flexible intermediate rules seem to give the best results.

The branching, or partitioning, rule of step 5, is usually a dichotomy of the form

$$x_k \leq \lfloor \bar{x}_k \rfloor \quad \vee \quad x_k \geq \lceil \bar{x}_k \rceil,$$

where x_k is some integer-constrained variable whose value $\bar{x}_k$ in the current solution to (P_i) is noninteger, while $\lfloor a \rfloor$ and $\lceil a \rceil$ denote the largest integer $\leq a$ and the smallest integer $\geq a$, respectively. The choice of the variable is important, but no reliable criterion is known for it. "Penalties" and "pseudo-costs" try to assess the change in $v(P_i)$ that will be produced by branching on x_k, with a view of providing a choice that will force the value of at least one of the new subproblems as high as possible.

In problems with some structure, more efficient branching rules are possible. In the presence of a "multiple choice" constraint

$$\sum_{j\in Q} x_j = 1, \qquad x_j = 0 \text{ or } 1,\ j\in Q,$$

for instance, one can branch on the dichotomy

$$x_j = 0,\ j\in Q_1 \ \vee \ \sum_{j\in Q_1} x_j = 1,\ x_j = 0,\ j\in Q\setminus Q_1$$

for some $Q_1 \subset Q$, thus fixing several variables at a time. Other, more sophisticated branching rules have been used for set covering, set partitioning and traveling salesman problems.

The logical tests of step 2, and/or associated inequalities, whenever applicable, were shown to substantially speed up the procedure. However, by far the most important ingredients of any enumerative procedure are the bounding devices used in steps 3 and 4. Dramatic improvements were registered in the case of such special structures like the traveling salesman problem, where the knowledge of deep cutting planes (usually facets of the convex hull of F) has made it possible to replace the common linear programming relaxation (L) by a much "stronger" one, either by amending (L) with cutting planes of the latter type, or by taking those same cutting planes into the objective function in the Lagrangean manner. In either case, the resulting vastly enhanced lower bounding capability has drastically reduced computing times. Similarly, improvements in the upper bounding procedure, like the use of an efficient heuristic to find feasible solutions, were found to affect decisively the performance of branch and bound methods. For surveys of this area see Balas (1975), Beale (1979), Spielberg (1979).

Partitioning the Variables or Constraints

Benders' partitioning procedure is based on the following result. Consider the problem

(P_1) $\quad \min\{cx + dy \mid Bx + Dy = b,\ x \geq 0,\ y \in Q\}$

where B and D are mXp and mXq matrices, respectively, c, d and b are vectors of conformable dimensions, while Q is an arbitrary set (for instance, the set of integer q-vectors) such that for every $y \in Q$, there exists an $x \geq 0$ satisfying $Bx + Dy = b$. Let $U = \{u \mid uB \leq c\}$, and let vert U be the (finite) set of vertices of the polyhedron U. Then (P_1) is equivalent to

(P_2) $\quad \min\{w_o \mid w_o \geq (d - uD)y + ub,\ u \in \text{vert } U,\ y \in Q\}$,

in the sense that if $(\bar{x}, \bar{y})$ solves (P_1), then $\bar{y}$ solves (P_2); and if $\hat{y}$ solves (P_2), there exists an $\hat{x}$ such that $(\hat{x}, \hat{y})$ solves (P_1). Although the inequalities of (P_2) usually outnumber those of (P_1) by far, they can be generated as needed by solving a linear program in the continuous variables x, or its dual (the latter having U as its constraint set). This approach can be useful in particular when B has a structure making it easy to solve the linear programs that provide the constraints of (P_2).

The second type of decomposition procedure, Lagrangean relaxation, partitions the set of constraints $Ax \geq b$ of (P) into $A_1x \geq b_1$ and $A_2x \geq b_2$, and formulates the Lagrangean problem

$$L(u) = \min\{(c - uA_2)x + ub_2 | A_1x \geq b_1,\ x \geq 0,\ x_j \text{ integer},\ j \in N_1 \subseteq N\}.$$

For any u, L(u) is a lower bound on the objective function value of (P). The problem in the variables u of maximizing L(u) subject to $u \geq 0$ is sometimes called the Lagrangean dual of (P). There are several methods for maximizing L(u) as a function of $u \geq 0$, one of them being subgradient optimization. If $\bar{u} \geq 0$ maximizes L(u) and $\bar{x}$ is a minimizing vector in $L(\bar{u})$, then $\bar{x}$ is an optimal solution to (P) if $A_2\bar{x} \geq b_2$ and $\bar{u}(A_2\bar{x} - b_2) = 0$. However, this is usually not the case, since $L(\bar{u})$ and the optimal objective function value of (P) tend to be separated by a so-called duality gap. Nevertheless, since calculating the value of L(u) for fixed u may be a lot easier than solving (P), this is often a convenient way of generating good lower bounds.

In particular, since $A_2x \geq b_2$ may consist partly (or wholly) of cutting planes, this is one way of using the latter without vastly increasing the number of inequalities explicitly added to the constraint set. For surveys of these techniques see Geoffrion (1974), Shapiro (1979), Fisher (1981).

Cutting Plane Theory

A central problem of integer programming theory is to characterize the convex hull of F, the set of integer points satisfying the inequalities of (P). F is called the feasible set, its convex hull (defined as the smallest convex set containing F) is denoted conv F. From a classical result of Weyl (1935), it is known that conv F is the intersection of a finite number of linear inequalities. In other words, (P) is equivalent to a linear program. Unfortunately, however, the constraint set of this linear program is in general hard to identify. Only for a small number of highly structured combinatorial optimization problems do we have at this time a linear characterization of conv F, i.e., an explicit representation of conv F by a system of linear inequalities. In the general case, all that we have are some procedures to generate sequences of inequalities that can be shown to converge to such a representation.

One way to solve the general integer program (P) is thus to start by solving (L), the linear programming relaxation of (P), and then to successively amend the constraint set of (L) by additional inequalities (cutting planes), until the whole region between the optimum of (L) and that of (P) is cut off. How much work is involved in this, depends on the strength (depth) of the cuts, as well as on the size of the region that is to be cut off, i.e., the size of the gap between v(L) and v(P), the value of L and P. This gap can be very large indeed, as evidenced by a recent result for the class of 0-1 programs called (unweighted) set covering problems (where all entries of A are 0 or 1, and all entries of b and c are 1). For a set covering problem in n variables and an arbitrary number of constraints, the ratio v(P)/v(L) is bounded by $\frac{n}{4} + \frac{1}{2}$ for n even, and by $\frac{n}{4} + \frac{1}{2} + \frac{1}{4n}$ for n odd. Furthermore, this is a best possible bound.

As to the strength of various cutting planes, it is useful to address the question from the following angle. Let $F \subset \mathbb{R}^n$, $d \in \mathbb{R}^n$ and $d_o \in \mathbb{R}$. The set $\{x \in \text{conv } F \mid dx = d_o\}$ is called a facet of conv F, if $dx \geq d_o$ for all $x \in F$ and $dx = d_o$ for n affinely independent points $x \in F$. In the integer programming literature the inequality $dx \geq d_o$ defining the facet is also called a facet. Facets are important because among many possible representations of conv F in terms of inequalities, the facets of conv F provide a minimal one. Obviously, they are the strongest possible cutting planes.

Subadditive cuts

Consider the integer program (P), with $N_1 = N$.

Solving the linear programming relaxation (L) of (P) produces a simplex tableau of the form

(1) $\qquad x_i = a_{io} + \sum_{j \in J} a_{ij}(-x_j) \qquad i \in I$

where I and J are the index sets of basic and nonbasic variables respectively. If a_{io} is noninteger and we denote $f_{ij} = a_{ij} - \lfloor a_{ij} \rfloor$, $\forall$ i, j, one can show that (1) together with the integrality of the variables, implies for every $i \in I$,

(2) $\qquad \sum_{j \in J} f_{ij} x_j \geq f_{io}$.

The inequality (2) is a cutting plane, since it is satisfied by every integer x that satisfies (1), but is violated for instance by the optimal solution to (L) associated with (1), in which all nonbasic variables are equal to 0. This cut was the basis of Gomory's method of integer forms, the first finitely convergent cutting plane algorithm for pure integer programs. An analogous cut provides a finitely convergent algorithm for mixed integer programs (with integer-constrained objective function value).

The derivation of the cut (2) is based on simple modular arithmetic. However, the integer program over the polyhedral cone defined by (1), together with the conditions

(3) $\qquad x_j$ integer, $j \in I \cup J$; $\qquad x_j \geq 0$, $j \in J$

(note that the conditions $x_j \geq 0$, $j \in I$ are omitted), is equivalent to an optimization problem over a commutative abelian group, that can be solved as a shortest path problem (Gomory, 1969). Whenever the vector $\bar{x}$ corresponding to the optimal solution found for the group problem satisfies the conditions $x_j \geq 0$, $j \in I$, it is an optimal solution to (P). When this is not the case, $\bar{x}$ provides a lower bound on v(P).

The key concept in Gomory's characterization of the "corner polyhedron", i.e., the convex hull of integer points in the above mentioned cone, is subadditivity. This has subsequently led to a subadditive characterization of the convex hull of F itself.

A function f defined on a monoid (semigroup) M is subadditive if $f(a + b) \leq f(a) + f(b)$ for all $a, b \in M$. Let A be an mXn matrix with rational entries, let a_j be the j^{th} column of A, and let $X = \{x | Ax = b, x \geq 0 \text{ integer}\} \neq \emptyset$. Then for any subadditive function f on the monoid $M = \{y | y = Ax \text{ for some integer } x \geq 0\}$, such that $f(0) = 0$, the inequality

$$\sum_{j=1}^{n} f(a_j)x_j \geq f(Ax) \tag{4}$$

is satisfied by every $x \in X$. Conversely, all valid inequalities for X are dominated by an inequality (4) for some subadditive function f on M such that $f(0) = 0$. For literature see Johnson (1974, 1980), Jeroslow (1979).

Disjunctive cuts

A different, geometrically motivated approach derives cutting planes from convexity considerations (intersection or convexity cuts, disjunctive cuts). This approach is directed primarily to the 0-1 programming problem. As mentioned earlier, 0-1 programming is coextensive with disjunctive programming, and the best way of describing the approach is by applying it to the disjunctive program

$$\text{(D)} \qquad \min\{cx \mid \bigvee_{i \in Q} (A^i x \geq b^i,\ x \geq 0)\}.$$

Here Q is an index set, A^i and b^i are $m_i \times n$ and $m_i \times 1$ matrices, and "V" means that at least one of the systems $A^i x \geq b^i$, $x \geq 0$, must hold. This is the disjunctive normal form of a constraint set involving logical conditions on inequalities, and any such constraint set can be brought to this form.

The convex hull of a disjunctive set is characterized by the following two results. Let the set be

$$F = \{x \mid \bigvee_{i \in Q} (A^i x \geq b^i,\ x \geq 0)\},$$

where A^i, b^i, $i \in Q$ are as above, and let Q^* be the set of those $i \in Q$ such that the system $A^i x \geq b^i$, $x \geq 0$ is consistent. Let $\alpha \in \mathbb{R}^n$ and $\alpha_o \in \mathbb{R}$. Then the inequality $\alpha x \geq \alpha_o$ is satisfied by every $x \in F$ if and only if there exists a set of vectors $\theta^i \in \mathbb{R}^{m_i}$, $\theta^i \geq 0$, $i \in Q^*$, such that

$$(5) \qquad \alpha \geq \theta^i A^i \text{ and } \alpha_o \leq \theta^i b^i \qquad , \qquad i \in Q^* .$$

Furthermore, if F is full dimensional, Q is finite, and $\alpha_o \neq 0$, then $\alpha x \geq \alpha_o$ is a facet of conv F if and only if $\alpha \neq 0$ is a vertex of the polyhedron

$$F\# = \{\alpha | \alpha \text{ satisfies (5) for some } \theta^i \geq 0,\ i \in Q^*\}.$$

The first of these results can be used to generate computationally inexpensive cutting planes for a variety of special cases of F, corresponding to logical conditions inherent to the problem at hand; whereas the second result can be used to strengthen any such cut, at an increasing computational cost, up to the point where it becomes a facet of conv F.

Often there is advantage in casting an integer program into the form of a disjunctive program with integrality constraints on some of the variables. For such problems, a procedure called <u>monoidal cut strengthening</u> that combines the disjunctive and subadditive approaches can be used to derive a family of cutting planes whose strength versus computational cost ratio compares favorably with cutting planes based on either approach taken separately.

A fundamental question of integer programming theory is whether the convex hull of feasible points can be generated seqtentially, by imposing the integrality conditions step by step. That is, by first producing all the facets of the convex hull of points satisfying the linear inequalities, plus the integrality condition on, say, x_1; then adding all these facet inequalities to the constraint set and generating the convex hull of points satisfying this amended set of inequalities, plus the integrality condition on x_2; etc. The question also has practical importance, since convex hull calculations for a mixed integer program with a single integer variable are much easier than for one with many integer variables.

To be more specific, suppose we wish to generate the convex hull of the set

$$X = \{x | Ax \geq b,\ x \geq 0,\ x_j \text{ integer},\ j = 1,\ldots,n\}.$$

Let

$$X_o = \{x | Ax \geq b,\ x \geq 0\}$$

and for $j = 1,\ldots,n$, define recursively

$$X_j = \text{conv}\{x \in X_{j-1} | x_j \text{ integer}\}.$$

Obviously, $X_n \subseteq \text{conv } X$; the question is, whether $X_n = \text{conv } X$?

The answer, obtained from disjunctive programming considerations, is that for a general integer program the statement $X_n = \text{conv } X$ is false; but that for a 0-1 program it is true. This is one of the main distinguishing properties of 0-1 programs among integer programs.

For literature see Balas (1979), Glover (1974), Jeroslow (1977).

Combinatorial cuts

Given a graph $G = (V, E)$ with vertex set V and edge set E, a matching in G is a set of pairwise nonadjacent edges of G. If A is the incidence matrix of vertices versus edges of G and a weight w_j is assigned to every edge j, the problem of finding a maximum-weight matching in G is the integer program

$$\max\{wx \mid Ax \leq e,\ x_j = 0 \text{ or } 1,\ j \in E\}$$

where $e = (1,\ldots,1)$ has $|V|$ components, and $x_j = 1$ if edge j is in the matching, $x_j = 0$ otherwise. Edmonds (1965) has shown that this problem can be restated as a linear program in the same variables, by adding an inequality of the form

$$\sum_{j \in E(S)} x_j \leq \frac{1}{2}(|S| - 1)$$

for every $S \subseteq V$ such that $|S|$ is odd. Here E(S) is the set of edges with both ends in S.

Unfortunately, the matching polytope is the exception rather than the rule, and for most combinatorial problems such a simple linear characterization of the convex hull of feasible points does not exist. However, certain classes of facets of the convex hull have been identified for several problems.

The vertex packing problem in a graph $G = (V, E)$ with vertex-weights c_i, $i \in V$, consists in finding a maximum weight independent (i.e., pairwise nonadjacent) set of vertices. If A is the same incidence matrix as before and T denotes transposition, the vertex packing problem is the integer program

$$\max\{cx \mid A^T x \leq e,\ x_j = 0 \text{ or } 1,\ j \in V\}$$

where e has $|E|$ components and $x_j = 1$ if vertex j is in the packing, $x_j = 0$ otherwise. Let I(G) denote the packing polytope of G, i.e., the convex hull of incidence vectors of packings in G.

Several classes of facets of I(G) are known. For instance, an inequality of the form

$$(6) \qquad \sum_{j \in K} x_j \leq 1$$

is a facet of I(G) if and only if $K \subseteq V$ is a clique, i.e., a maximal set of pairwise adjacent vertices of G. The class of graphs whose packing polytope I(G) is completely described by this family of inequalities (i.e., I(G) has no other nontrivial facets) is called perfect. A graph is known to be perfect if and only if its complement is perfect. The properties of perfect graphs and their packing polyhedra have been intensely studied during the sixties and seventies and have, among other things, served as a starting point for a theory of blocking and antiblocking polyhedra developed by Fulkerson (1971).

More generally, many classes of facets of I(G) are associated with certain induced subgraphs G' of G. When G' is induced by a clique, the corresponding inequality (6) is, as mentioned above, a facet of I(G). Other induced subgraphs G' yield inequalities that are facets of $I(G')$ rather than I(G), but can be used to obtain corresponding facets for $I(G)$ through a procedure called _lifting_. For instance, if $G' = (V', E')$ is (i) an odd hole (i.e., a chordless cycle of odd length), or (ii) the complement of an odd hole, then

$$\sum_{j \in V'} x_j \leq k$$

is a facet of $I(G')$, with $k = \frac{1}{2}(|V'| - 1)$ in case (i), and $k = 2$ in case (ii).

The above mentioned lifting procedure is based on the following result. Let G' be any subgraph of G induced by $V' \subset V$, and let

$$\sum_{j \in V'} \alpha_j x_j \leq \alpha_o$$

be a facet of $I(G')$. Then there exist integers β_j, $0 \leq \beta_j \leq \alpha_o$, such that

$$\sum_{j \in V'} \alpha_j x_j + \sum_{j \in V \setminus V'} \beta_j x_j \leq \alpha_o$$

is a facet of $I(G)$. The coefficients β_j can be calculated sequentially, and their values depend on the particular sequence. These calculations involve the solution of an integer program for each coefficient, but for certain special structures they become manageable.

Other combinatorial problems for which several classes of facets of the feasible set have been characterized, include the knapsack problem, the traveling salesman problem, etc.

For literature see the books by Ford and Fulkerson (1962) and Lawler (1976) and the surveys by Balas and

Padberg (1976), Hoffman (1979), Lovász (1979), Klee (1980), Padberg (1979).

Computer Implementation

At present all commercially available integer programming codes are of the branch and bound type. While they can sometimes solve problems with hundreds of integer and thousands of continuous variables, they cannot be guaranteed to find optimal solutions in a reasonable amount of time to problems with more than 30-40 variables. On the other hand, they usually find feasible solutions of acceptable quality to much larger problems. These commercial codes, while quite sophisticated in their linear programming subroutines, do not incorporate any of the results obtained in integer programming during the last decade.

A considerable number of specialized branch and bound/implicit enumeration algorithms have been implemented by operations research groups in universities or industrial companies. They usually contain other features besides enumeration, like cutting planes and/or Lagrangean relaxation. Some of these codes can solve general (unstructured) 0-1 programs with up to 80-100 integer variables, and structured problems with up to several hundred (assembly line balancing, multiple choice, facility location), a few thousand (sparse set covering or set partitioning, generalized assignment), or several thousand (knapsack traveling salesman) 0-1 variables.

Cutting plane procedures for general pure and mixed integer programs are at present too erratic and slow to compete with enumerative methods. However, for a number of special structures (set covering, traveling salesman problem) where information available about the convex hull of feasible points has made it possible to generate strong inequalities at acceptable computational cost, cutting planes, either by themselves, or in combination with enumerative and/or Lagrangean techniques, have been highly successful.

At the current state of the art, while many real world problems amenable to an integer programming formulation fit within the stated limits and are solvable in useful time, others substantially exceed those limits. Furthermore, some important and frequently occurring real world problems, like job shop scheduling and others, lead to integer programming models that are almost always beyond the limits of what is currently solvable. Hence the great importance of approximation methods for such problems.

For literature on computer codes see Land and Powell (1979), Spielberg (1979).

References

Balas, E. (1965), "An Additive Algorithm for Solving Linear Programs with Zero-One Variables." Operations

Research, 13, p. 517-546.

Balas, E. (1975), "Bivalent Programming by Implicit Enumeration." J. Belzer, A. G. Holzman and A. Kent (editors), Encyclopedia of Computer Science and Technology, vol. 2, M. Dekker Inc., New York, p. 479-494.

Balas, E. (1979), "Disjunctive Programming." Annals of Discrete Mathematics 5, p. 3-51.

Balas, E. and M. Padberg (1976), "Set Partitioning: A Survey," SIAM Review, 18, p. 710-760.

Beale, E. M. L. (1979), "Branch and Bound Methods for Mathematical Programming." Annals of Discrete Mathematics 5, p. 201-220.

Benders, J. F. (1962), "Partitioning Procedures for Solving Mixed-Variables Programming Problems." Numerische Mathematik, 4, p. 238-252.

Christofides, N., A. Mingozzi, P. Toth and C. Sandi, editors (1979), Combinatorial Optimization, Wiley.

Edmonds, J. (1965), "Maximum Matching and a Polyhedron with 0-1 Vertices." J. Res. Nat. Bur. Stds., 69B, p. 125-130.

Fisher, M. L. (1981), "The Lagrangean Relaxation Method for Solving Integer Programming Problems." Management Science, 27, p. 1-18.

Ford, L. R., Jr. and D. R. Fulkerson (1962), Flows in Networks, Princeton University Press.

Fulkerson, D. R. (1971), "Blocking and Antiblocking Pairs of Polyhedra." <u>Mathematical Programming</u>, <u>1</u>, p. 168-194.

Garfinkel, R. S. and G. L. Nemhauser (1972), <u>Integer Programming</u>, Wiley.

Geoffrion, A. (1974), "Lagrangean Relaxation and Its Uses in Integer Programming." <u>Mathematical Programming Study 2</u>, p. 82-114.

Glover, F. (1975), "Polyhedral Annexation in Mixed Integer Programming." <u>Mathematical Programming</u>, <u>9</u>, p. 161-188.

Gomory, R. (1958), "Outline of an Algorithm for Integer Solutions to Linear Programs." <u>Bull. Amer. Math. Soc.</u>, <u>64</u>, p. 275-278.

Gomory, R. (1969), "Some Polyhedra Related to Combinatorial Problems." <u>Linear Algebra and Its Applications</u>, <u>2</u>, p. 451-558.

Hammer, P. L., E. L. Johnson and B. H. Korte, editors (1979a), <u>Discrete Optimization I (Annals of Discrete Mathematics 4</u>), North-Holland.

Hammer, P. L., E. L. Johnson and B. H. Korte, editors (1979b), <u>Discrete Optimization II (Annals of Discrete Mathematics 5</u>), North-Holland.

Hoffman, A. J. (1979), "The Role of Unimodularity in Applying Linear Inequalities to Combinatorial

Theorems." Annals of Discrete Mathematics 4, p. 73-84.

Hoffman, A. J. and J. B. Kruskal (1958), "Integral Boundary Points of Convex Polyhedra," in Kuhn, H. W. and A. W. Tucker (editors), Linear Inequalities and Related Systems, Princeton University Press, p. 223-246.

Jeroslow, R. G. (1977), "Cutting Plane Theory: Disjunctive Methods." Annals of Discrete Mathematics, 1, p. 293-330.

Jeroslow, R. G. (1979), "An Introduction to the Theory of Cutting Planes." Annals of Discrete Mathematics, 5, p. 71-95.

Johnson, E. L. (1979), "On the Group Problem and a Subadditive Approach to Integer Programming." Annals of Discrete Mathematics, 5, p. 97-112.

Johnson, E. L. (1980), Facets, Subadditivity and Duality for Group and Semi-group Problems, SIAM.

Klee, V. (1980), "Combinatorial Optimization: What Is the State of the Art?" Mathematics of Operations Research, 5, p. 1-26.

Land, A. H. and A. G. Doig (1960), "An Automatic Method for Solving Discrete Programming Problems." Econometrica, 28, p. 497-520.

Land, A. and S. Powell (1970), "Computer Codes for Problems of Integer Programming." Annals of Discrete

Mathematics 5, p. 221-269.

Lawler, E. L. (1976), Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston.

Lovász, L. (1979), "Graph Theory and Integer Programming." Annals of Discrete Mathematics 4, p. 141-159.

Padberg, M. W. (1979), "Covering, Packing and Knapsack Problems." Annals of Discrete Mathematics 4, p. 265-287.

Padberg, M. W., editor (1980), Combinatorial Optimization. Mathematical Programming Study 12, North-Holland.

Shapiro, J. F. (1979), "A Survey of Lagrangean Techniques for Discrete Optimization." Annals of Discrete Mathematics 5, p. 113-138.

Spielberg, K. (1979), "Enumerative Methods in Integer Programming." Annals of Discrete Mathematics 5, p. 139-184.

Weyl, H. (1935), "Elementare Theorie der Konvexen Polyeder." Commentarii Mathematici Helvetici, 7, p. 290-306. English translation in Kuhn, H. W. and A. G. Tucker, editors (1950), Contributions to the Theory of Games, Princeton University Press.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: MSRR #479 — 2. GOVT ACCESSION NO.: AD-A110 7	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Integer Programming	5. TYPE OF REPORT & PERIOD COVERED: Technical Report November 1981
	6. PERFORMING ORG. REPORT NUMBER: MSRR 479
7. AUTHOR(s): Egon Balas	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0621 NR 047-048
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Personnel and Training Research Programs, Office of Naval Research (Code 434), Arlington, VA 22217	12. REPORT DATE: November 1981
	13. NUMBER OF PAGES: 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

integer programming, disjunctive programming, discrete optimization, combinatorial optimization, branch and bound, implicit enumeration, cutting planes, convexifications

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is an introductory survey of integer programming, its theory, methodology and applications, for the Encyclopedia of Statistical Sciences.

ATE
LMED

-8