Jaroslav Hrouda

A contribution to Balas' algorithm

Aplikace matematiky, Vol. 16 (1971), No. 5, 336–353

Persistent URL: http://dml.cz/dmlcz/103366

A CONTRIBUTION TO BALAS' ALGORITHM

JAROSLAV HROUDA

(Received June 26, 1970)

INTRODUCTION

It is the Balas' additive algorithm of solving the zero-one linear programming problem which is meant in the title of the article. It is one of several possible approaches to the problem, perhaps the best-known even if not quite successful. This algorithm has become classical nowadays and many authors have added modifications and supplements to it (the most important are due to Glover and Geoffrion). The aim of our article is (1) to give an alternative description of the algorithm suitable as a propedeutics for the following article [16] in which some generalization of Balas' algorithm will be introduced, (2) to systematize and generalize some older tests. Let us mention the contents of the article more in detail.

In § 1 we briefly describe a backtracking-type enumeration process using our own terminology (essentially equivalent to that of [15]). At the time when terminology is being still formed such redundancy need not be useless. We believe that our terminology is quite natural and hence understandable.

In § 2 a modification of Balas', or better Geoffrion's, algorithm is presented under the name Algorithm BG (without tests so far). Treating the objective function as the 0-th constraint simplifies the formalism; e.g. two versions of the algorithm – for obtaining all optimal solutions (BG1), and at least one optimal solution (BG1′) differ from each other in only one point. Coefficients of the objective function are not assumed to be nonnegative since this assumption need not be always to the best advantage. Glover's method of bookkeeping of the enumeration process is somewhat adapted.

In § 3 three fairly strong tests are described: (1) Test BF generalizes some tests of Balas and Fleischmann. It works upon a pair of sets F and G consisting respectively of elements which must or must not be present so that feasibility may be achieved in a given branch. The sets F, G are constructed here to be the largest possible of the kind. (2) Test GZ generalizes a test of Glover and Zionts. It is no more related to the objective function exclusively. A heuristic procedure is suggested to determine an order in which constraints are to enter the test, and also a suggestion is given for surrogate constraints to be used. (3) Test P1 applies the same generalizing idea as GZ to a test of Petersen.

Now a few remarks about symbolics and terminology used in the article.

Inclusion signs $\subset$, $\supset$ have only sharp meaning (excluding equivalence).

Symbol $|S|$ where S stands for a set denotes the set of absolute values of the elements of S.

For typographic reasons we shall often write down a summation index into angle brackets after the sum, e.g.

$$\sum a_{ij}\langle j \in J\rangle \quad \text{instead of} \quad \sum_{j\in J} a_{ij}\,.$$

The braces $\{\ldots\}$, $\{\ldots \mid \ldots\}$ define sets by enumerating the elements or by indicating a condition, respectively.

The sayings "i-th row, j-th column, s-th iteration" refer to the value of the corresponding index, not to the order from the beginning; so i-th row is the row indexed by i, though it may lie at the $(i + 1)$-st position if there is a 0-th row.

§ 1. ENUMERATION PROCESS B

Given the sets $\mathscr{M} = \{0, 1, \ldots, m\}$, $\mathscr{N} = \{1, \ldots, n\}$. We start our consideration by

Problem U. *Determine all the vectors* $x = (x_1, \ldots, x_n)$ *satisfying the constraints*

$$\sum_{j=1}^{n} a_{ij}x_j \leqq b_i \quad (i \in \mathscr{M})\,, \tag{1.1}$$

$$x_j = 0 \quad or \quad 1\,. \tag{1.2}$$

A vector x will be said to be a *solution* or a *feasible solution* according to whether it satisfies only (1.2) or (1.2) and also (1.1). Each solution x can be assigned a *combination* J of elements from $\mathscr{N}$ in this simple way: $j \in \mathscr{N}$ belongs to J if and only if $x_j = 1$. Since this correspondence is unique, we may speak about combinations and feasible combinations concerning Problem U as well. We denote combinations that take l elements by J_l. The zero solution $x = 0$ is assigned empty "combination" $J_0 = \emptyset$.

We define the *branch on* J_l *with the outset* $J_{l+1} \supset J_l$ as a set of all combinations $J_\mu \supseteq J_{l+1}$. The parameter l is called the *branching level* in this connexion. Naturally, sub-branches may be distinguished in the structure of a branch.

To obtain all the combinations J_l $(l = 0, \ldots, n)$, we shall use a procedure called here *enumeration process* B (briefly, Process B). It is known as 'backtracking procedure' in literature [1]) and can be characterized by a recursive rule: Form all branches on

[1]) Its general formulation was given by Walker [21] and Golomb and Baumert [13].

each newly obtained combination. We shall perform it by means of two elementary branching mechanisms:

(α) IBL (increasing of branching level): add an element which has not yet started branching on a given combination.

(β) DBL (decreasing of branching level): drop out the element which was added last to make a combination.

Obviously, starting from an arbitrary combination (e.g. J_0) and applying the branching mechanisms with priority (α) over (β), we shall realize Process B. In other words: decreasing of branching level can take place only when all possibilities of increasing of a given level have been exhausted. For our purpose, however, the branching process is necessary to be reduced in some ways. Two of them are known:

Reduction of the 1-st kind. When creating branches those combinations (subbranches) which were already created before are skipped. The part of a branch which has been made in this way is a *nonredundant* branch. To get a precise meaning, however, this concept must be understood in relation to the enumeration process — the size of a nonredundant branch depends on the stage of the process at which it was created, i.e. when its "turn" came. (The only branches created entirely are the first branches on $J_0 \subset J_1 \subset \ldots \subset J_{n-1}$.) In the whole, we get a nonredundant enumeration process — each combination being present only once.

Reduction of the 2-nd kind. Those combinations which can be found nonfeasible in advance are left out. The criteria that facilitate such conclusions are called *tests.* The most valuable will be those, of course, which eliminate from consideration whole nonredundant branches, not only individual combinations. The reduction of this kind is called implicit or partial enumeration in literature [8], [2]. It results in, as it might be called, shortened branches and, eventually, shortened enumeration process.

In the sequel the two reductions will usually be understood under the term Process B. In the same way, speaking about branches we shall usually mean shortened nonredundant ones.

For the realization of an enumeration process its *bookkeeping* is important; by that a way of recording the running process is meant so that we can read combinations being created and control the process. Naturally, the bookkeeping is required to be economical enough regarding the use of computers. We now describe briefly a bookkeeping due to Glover [10] that fulfils this requirement in a high degree (also described in [8]).[2])

A current state of Process B is determined by a sequence Γ consisting of elements j and $-j$ ($j \in \mathcal{N}$); all these are different in absolute value (thus Γ includes at most n elements). The positive elements represent the combination J created last, the negative ones keep "history" of the process — the combinations created before J or those eliminated by tests. The branching mechanisms then work as follows:

[2]) Other bookkeepings, also economical in some sense, were proposed in [19], [10] p. 913.

(α) IBL: add an element $j \in \mathcal{N} - |\Gamma|$ to the end of the sequence Γ;

(β) DBL: prefix the sign 'minus' to the last positive element of the sequence Γ and leave out all (negative) elements standing after it.

The process starts with Γ being empty or containing an arbitrary combination of different elements $j \in \mathcal{N}$, and ends when Γ contains no positive elements. Whenever we think of Process B in conjunction with this Glover's bookkeeping, we shall express it by the name *Process* BG.

Now, we shall return to our Problem U. In view of what was said above, to solve it means to select all the feasible combinations among those produced by Process B. Whether such way will be of practical value, it will depend especially on the efficiency of tests with respect to the given constraints (1.1). As a rule, Problem U is stated in a weaker form as

Problem U1. *Specify all feasible solutions of Problem* U *that minimize the objective function*

$$z(x) = \sum_{j=1}^{n} a_{0j} x_j \tag{1.3}$$

assuming $b_0 \geqq \sum_{j=1}^{n} \max \{a_{0j}, 0\}$.

This problem is known as zero-one linear programming problem. The feasible solutions which solve it are *optimal solutions*. The minimization requirement may be well utilized to increase reduction effect of tests. If it is sufficient to find only one from all optimal solutions, we shall refer to the problem as U1′.

§ 2. ALGORITHM BG

Under this title we are going to describe an algorithm which solves Problem U according to the outline given in § 1. Superscripts without parentheses stand for iteration numbers.

Preparation part. Set $s = 0$, $l^0 = 0$, $\Gamma^0 = \emptyset$, $y_i^0 = b_i \; (i \in \mathcal{M})$, $v_0^0 = 1$.

Iteration part (s-th iteration). Given a sequence Γ^s, numbers $l^s \equiv l$, $y_i^s \; (i \in \mathcal{M})$, and a set $\{v_0^s, \ldots, v_l^s\}$.

If $y_i^s \geqq 0$ for all $i \in \mathcal{M}$ and if either $s = 0$ or IBL took place in the preceding iteration, then score a *feasible solution* $J_l^s = \{j \in \Gamma^s \mid j > 0\}$.

In any case, go on by selecting a set of "free" elements $N^s = \mathcal{N} - |\Gamma^s|$. If $N^s = \emptyset$, then go to DBL (see below); else apply tests to reduce the enumeration process (see § 3). If none the tests is satisfied, go to

IBL: Let $F^s = \{j_1, \ldots, j_\varrho\}$, $G^s = \{k_1, \ldots, k_\sigma\}$ be disjoint subsets of N^s with the following meaning: F^s contains the elements which *must* be present in combinations of the branches on J_l^s so that a feasible solution may be reached; G^s, on the other hand, contains the elements which *must not* be present for this reason.[3]) Reduction possibilities which might arise from the size of these sets (e.g. $N^s - G^s = \emptyset$) are supposed to have been accepted in the tests.

(a) If $F^s \neq \emptyset$, then put

$$\Gamma^{s+1} = \Gamma^s \cup \{-k_1, \ldots, -k_\sigma\} \cup \{j_1, \ldots, j_\varrho\}, \tag{2.1}$$

$$y_i^{s+1} = y_i^s - \sum_{j \in F^s} a_{ij} \quad (i \in \mathcal{M}), \tag{2.2}$$

$$v_{l+\varrho}^{s+1} = v_l^s + \varrho, \tag{2.3}$$

$$l^{s+1} = l^s + \varrho. \tag{2.4}$$

The conjunction sign in equation (2.1) means adding new elements to the end of Γ^s in the indicated order.

(b) If $F^s = \emptyset$, then determine an element j_1 from the relation

$$v_{j_1}^s = \max_{j \in N^s - G^s} v_j^s \text{ [4])} \tag{2.5}$$

where

$$v_j^s = \sum_{i \in \mathcal{M}} \min \{y_i^s - a_{ij}, 0\}. \tag{2.6}$$

Apply formulas (2.1), (2.2), and put

$$v_{l+1}^{s+1} = 1, \tag{2.3'}$$

$$l^{s+1} = l^s + 1. \tag{2.4'}$$

Assign superscript $s + 1$ also to all symbols which pass to the next iteration unchanged (namely $v_0, \ldots, v_l$), and proceed to the $(s + 1)$-st iteration.

DBL: If $l^s < v_l^s$, then stop – the problem has been solved. Otherwise, leave out the elements of Γ^s from the end through v_l^s positive ones, prefix the sign 'minus' to the last of them to get the sequence Γ^{s+1}, and put

$$l^{s+1} = l^s - v_l^s, \tag{2.7}$$

$$y_i^{s+1} = y_i^s + \sum_{j_\varkappa > 0} a_{ij_\varkappa} \quad (i \in \mathcal{M}) \tag{2.8}$$

where the summation is done over the positive elements $j_\varkappa$ left out from Γ^s. Assign superscript $s + 1$ to all symbols which pass to the next iteration unchanged, and proceed to the $(s + 1)$-st iteration.

[3]) For a way of preparation of such sets see Test BF (§ 3).

[4]) In case of tie choose j_1 according to the least a_{0j}.

For solving *Problems* U1 *and* U1′ the above algorithm is to be only slightly modified: After a feasible solution has been scored, i.e. $y_i^s \geqq 0$ $(i \in \mathcal{M})$ with IBL preceded, replace y_0^s by $\tilde{y}_0^s = 0$ in case of U1 and by $\tilde{y}_0^s = -\varepsilon$ in case of U1′. Here $\varepsilon > 0$ is a given number not exceeding the possibly least variation of the value of the objective function (so if all a_{0j} are integers, we may simply take $\varepsilon = 1$). These two adaptations will be referred to as Algorithms BG1 and BG1′ respectively.

We provide only a few comments on Algorithm BG which we believe to be sufficient for its understanding:

The mechanism IBL is treated here in a more general sense than described in § 1 – now more elements can be added all at once (those from F^s). A corresponding generalization for DBL is ensured by quantities v_l which register the way of adding elements to Γ making use of the characteristic property of sets F: After the branching level has been decreased to the level of J_l^s on which IBL with $F^s \neq \emptyset$ preceded, it is possible to decrease the branching level further without any testing (branches created on J_l^s would no longer contain elements of F^s, which is the necessary condition for feasibility in this case). This property of sets F^s is a cumulative one – therefore apply (2.3) if $F^s \neq \emptyset$; otherwise (2.3′). Thus v_l^s always specifies how much the level l^s is to be decreased when DBL has occured.[5])

The quantity y_i^s, as it is apparent from Preparatoin part of the algorithm, (2.2), and (2.8), represents a current 'slack' value of the i-th constraint, i.e.

$$y_i^s = b_i - \sum a_{ij} \langle j \in J_l^s \rangle \,. \tag{2.9}$$

It is sufficient for feasible solutions to be scored only after IBL (or when $s = 0$) because at that time they appear first. Scoring after DBL, when the algorithm "passes through a solution on its back way", would be a repetition.

The way of determining j_1 according to (2.5) was originally suggested by Balas [1]. The aim involved is to reach a feasible solution as fast as possible. Of course, another criterion of preparing j_1 may be installed here, e.g. see [8] sect. 3.2.

Algorithm BG is in its mathematical background identical with the algorithms described and proved by Balas [1], Glover [10], and Geoffrion [8]. Hence it is not necessary to carry out a formal proof to make sure that Algorithm BG actually realizes Process BG.

Concerning Algorithm BG1: This differs from BG only in the role of the 0-th constraint, the right hand side of which now represents the current lowest value of the objective function so far achieved upon feasible solutions. All the solutions which would produce a greater value of the objective are then eliminated as "infeasible" ones. Indeed, let us define a sequence of quantities $b_0^s = y_0^s + z^s$ where $z^s = \sum a_{0j}$ $\langle j \in J_l^s \rangle$. It follows from (2.9) that the values of b_0^s may change merely at those

---

[5]) v_l's along with the 'minus' sign have a function equivalent to that of the Glover's underline [10], [8].

iterations at which a feasible solution has been scored, and they decrease since there is $y_0^s \geqq 0$ there. Using tilde to distinguish the changed value of b_0^s, we have

$$b_0^s = y_0^s + z^s \geqq z^s = \tilde{y}_0^s + z^s = \tilde{b}_0^s\,. \tag{2.10}$$

For two arbitrary feasible combinations J^s and J^u $(u > s)$ it is

$$z^u = b_0^u - y_0^u \leqq \tilde{b}_0^s = z^s\,. \tag{2.11}$$

Concerning Algorithm BG1′: Analogously to the above arguments, replacing relations (2.10), (2.11) by

$$b_0^s = y_0^s + z^s \geqq z^s > -\varepsilon + z^s = \tilde{y}_0^s + z^s = \tilde{b}_0^s\,, \tag{2.10′}$$

$$z^u = b_0^u - y_0^u \leqq \tilde{b}_0^s = -\varepsilon + z^s\,. \tag{2.11′}$$

Remark 1. Enumeration can be started from an arbitrary combination J_l^0 $(l \geqq 0)$; only Preparation part of the algorithm is to be altered: $\Gamma^0 = J_l^0$, $l^0 = l$, $y_i^0 = b_i - \sum a_{ij} \langle j \in J_l^0 \rangle$ $(i \in \mathcal{M})$, $v_0^0 = v_1^0 = \ldots = v_l^0 = 1$.

Remark 2. The function of Algorithm BG may be simulated by Algorithm BG1: we solve an auxiliary "minimization" problem, which is Problem U with the zero objective function added.

The usefulness of a nontrivial upper bound b_0 of the optimal value of the objective function has been emphasized [7]. We may attempt to get such a bound by various ways, e.g. to determine some feasible solution $\tilde{x}$ by a heuristic procedure ([20], [5]) and put $b_0 = z(\tilde{x})$. In [17] another way is suggested: Starting from a trial value of b_0, increase it gradually by steps until a feasible solution is found; hence the problem splits into a number of stages which may amount, on the whole, to less computation than the original problem itself (this idea is more deeply elaborated in [16]).

Balasian algorithms usually require $a_{0j} \geqq 0$.[6]) Some tests are actually based on this assumption. Obviously, this is because of the important reduction role the objective function may play in Problem U1 (of course, from the moment a "good" feasible solution is found). Otherwise, there is no reason for this neither as regards the construction of tests, as we are going to show in the next paragraph, nor regarding a possibly better course of the enumeration process, as it turned out in some numerical experiments [17]. Therefore Algorithm BG (BG1) assumes nothing about the coefficients of the problem.

[6]) If this requirement is not fulfilled, it can be easily achieved by the transformation $x'_j = 1 - x_j$.

§ 3. TESTS

First we mention some *principles* that are used to construct tests. These are:

(1) Some y_i $(i \in \mathscr{M})$ is such that in the branches on J^s

(1α) it cannot be nonnegative, or

(1β) if it were nonnegative, then another y_k would turn out negative.

In both cases DBL may follow.

(2) Some x_j $(j \in N^s)$ are determined to assume specific values to reach feasibility in a branch on J^s, particularly,

(2α) $x_j = 1$ for $j \in F^s \subseteq N^s$,

(2β) $x_j = 0$ for $j \in G^s \subseteq N^s$ $(F^s \cap G^s = \emptyset)$.

Then those branches which do not contain all the elements of F^s or do contain some elements of G^s can be omitted.

A great number of tests based more or less on the principles mentioned above has been described in literature [1], [2], [3], [4], [6], [10], [12], [15], [19], [20]. The situation has become rather confused. Moreover, some of these tests are redundant in the sense that they just anticipate the effect of other, more general tests. In this paragraph we are going to formulate some realizations of the principles (1) and (2) as general as we are able to find.

1. Test BF. Let us denote sets $\mathscr{S}_i^- = \{j \in \mathscr{N} \mid a_{ij} < 0\}$ and $\mathscr{S}_i^+ = \{j \in \mathscr{N} \mid a_{ij} > 0\}$, put $F^{(0)} = G^{(0)} = \emptyset$, and start a recursive process for $k = 0, 1, \ldots$:

$$(3.0)\qquad r_i^{(k)} = y_i^s - \sum a_{ij} \langle j \in ((N^s - G^{(k)}) \cap \mathscr{S}_i^-) \cup (F^{(k)} \cap \mathscr{S}_i^+) \rangle \quad (i \in \mathscr{M}),$$

$$F^{(k+1)} = F^{(k)} \cup \bigcup_{i \in \mathscr{M}} \{j \in N^s - (F^{(k)} \cup G^{(k)}) \mid r_i^{(k)} + a_{ij} < 0\},$$

$$G^{(k+1)} = G^{(k)} \cup \bigcup_{i \in \mathscr{M}} \{j \in N^s - (F^{(k+1)} \cup G^{(k)}) \mid r_i^{(k)} - a_{ij} < 0\}.$$

As soon as $r_i^{(k)} < 0$ for some k and i, go to DBL. Otherwise, after the process has been stabilized, i.e. when $F^{(k+1)} = F^{(k)}$ and $G^{(k+1)} = G^{(k)}$ for some k, put

$$F^s = F^{(k)}, \quad G^s = G^{(k)}, \quad r_i^s = r_i^{(k)} \quad (i \in \mathscr{M}).$$

If then $F^s = \emptyset$, $N^s \cup G^s = \mathscr{N}$, go to DBL; otherwise terminate the test.

The sets $F^{(k)}$, $G^{(k)}$ have the following meaning: $F^{(k)}$ contains the elements which must be present so that combinations in the branches on J^s may be feasible. The same applies to $G^{(k)}$ if replacing 'must' by 'must not'.

Justification. Symbol $\mathsf{B}(J^s, k)$ will denote a nonredundant branch on J^s with the outset $J^s \cup \{k\}$; further we define $\mathsf{B}(J^s, H) = \bigcup_{j \in H} \mathsf{B}(J^s, j)$. Let us assume that the sets $F^{(k)}$, $G^{(k)}$ have the meaning described above. Then the quantity $r_i^{(k)}$ obviously represents the largest value that the variable y_i can assume in the branches on J^s:

$$r_i^{(k)} = \max_J \{b_i - \sum_{j \in J} a_{ij} \mid J \subseteq \mathsf{B}(J^s, H^{(k)})\}$$

where $H^{(k)} = N^s - G^{(k)}$. Thus the inequality $r_i^{(k)} < 0$ indicates the impossibility of reaching a feasible solution in branches on J^s.

Now we shall suppose $r_i^{(k)} \geqq 0$. Let $j_\times \in H^{(k)} - F^{(k)}$ be such that $r_i^{(k)} + a_{ij_\times} < 0$ (so that $j_\times \in \mathscr{S}_i^-$). Then we can write

$$r_i^{(k)} + a_{ij_\times} = y_i^s - \sum a_{ij} \langle j \in ((N^s - G^{(k)} - \{j_\times\}) \cap \mathscr{S}_i^-) \cup$$
$$\cup (F^{(k)} \cap \mathscr{S}_i^+) \rangle = \max_J \{b_i - \sum_{j \in J - \{j_\times\}} a_{ij} \mid J \subseteq \mathsf{B}(J^s, H^{(k)})\} < 0,$$

which makes apparent that the element $j_\times$ can be added to the set $F^{(k)}$ since its absence in combinations would cause their infeasibility.

Similarly, let $j_{\times\times} \in H^{(k)} - F^{(k+1)}$ be such that $r_i^{(k)} - a_{ij_{\times\times}} < 0$ (so that $j_{\times\times} \in \mathscr{S}_i^+$). From the relation

$$r_i^{(k)} - a_{ij_{\times\times}} = \max_J \{b_i - \sum_{j \in J} a_{ij} \mid J \subseteq \mathsf{B}(J^s \cup \{j_{\times\times}\},\ H^{(k)} - \{j_{\times\times}\})\} < 0$$

it is now apparent that the element $j_{\times\times}$ can be added to the set $G^{(k)}$ since its presence in combinations would cause their infeasibility.

Finally, as the sets $F^{(0)} = G^{(0)} = \emptyset$ satisfy the assumption trivially, we have proved our assertion about the meaning of $F^{(k)}$ and $G^{(k)}$ for any k. The recursive process is finite because the sets involved are finite and disjoint. Concerning the second possibility of DBL, there is no need for comments.

The sets F^s and G^s created by Test BF are the largest ones of the kind, i.e. no element $j \in N^s - (F^s \cup G^s)$ exists such that either $r_i^s + a_{ij} < 0$ or $r_i^s - a_{ij} < 0$ for some $i \in \mathscr{M}$. That makes a difference from the tests of Balas [1], [2] and Fleischmann [6], a generalization of which Test BF appears to be. Still in other words: whenever $F^{(k)}$ or $G^{(k)}$ are extended on some constraint, then it influences back the constraints already examined. Such approach to a problem, when the constraints are considered as a system of *simultaneous* conditions, makes it quite natural to examine all of them, not only the "infeasible" ones. (An a_{ij}, $j \in F^{(k)} \cap \mathscr{S}_i^+$ in (3.0) can cause $r_i^{(k)} < 0$ even for $y_i^s \geqq 0$.)

Regarding the principles, Test BF makes use of (1α) and (2). The reduction effect of the latter in Algorithm BG is accomplished by formula (2.1) and the quantities v_l.

Remark 1. Test BF could be modified to include the following criterion: The set $G^{(k+1)}$ will not be disjoint to $F^{(k+1)}$, but

$$G^{(k+1)} = G^{(k)} \cup \bigcup_{i \in \mathscr{M}} \{j \in N^s - (F^{(k)} \cup G^{(k)}) \mid r_i^{(k)} - a_{ij} < 0\}.$$

As soon as $(F^{(k+1)} - F^{(k)}) \cap (G^{(k+1)} - G^{(k)}) \neq \emptyset$, then DBL. Since this arrangement would probably cause a difficulty in programming, we do not consider it any more.

Remark 2. Suppose that the sets G^s and F^s are produced only by Test BF. If in the s-th iteration $F^s \neq \emptyset$ and IBL is to follow, then in the next iteration the test need not be applied. Indeed, equations (3.0), (2.2), and $N^{s+1} = N^s - (F^s \cup G^s)$ imply that $r_i^{(0)}$ for the $(s + 1)$-st iteration are identical with r_i^s resulting from the s-th iteration, but there all possibilities to increase F, G were already exhausted.

Regarding the computational efficiency, the algorithm of Test BF should be worked out more carefully. Therefore, we present

A practical version of Test BF. The sets F and G are now developed gradually along with i. After some of them is augmented on the λ-th constraint $(\lambda \leqq i)$ then all implied corrections of r_ι's $(\iota \leqq i)$ and, as the case may be, further augmentation of F or G are immediately realized. We do not proceed to the $(i + 1)$-st row before we have obtained the largest sets F, G by means of the 0-th through i-th constraints.

The information about r_ι's which were corrected is kept by quantities-indicators χ_ι (moreover, they can predetermine which rows are to be tested at all):

$$\chi_\iota = \begin{cases} 1 \ \dots \ \iota\text{-th row will be examined by the test,} \\ 0 \ \dots \ \iota\text{-th row was examined by the test,} \\ -1 \ \dots \ \iota\text{-th row will be ignored by the test.} \end{cases}$$

Such arrangement of the test algorithm guarantees the system of constraints to be passed through at most once; only those rows are treated repeatedly in which r_ι are corrected.

Another saving of computation is introduced by quantities ω_i. Their purpose is to suppress the examination of such a row in which none of the inequalities $r_\lambda \pm a_{\lambda j} < 0$ can hold or there is a little chance of it.

The test is again an iteration process, now depending on indices $\varkappa \geqq 0$ and $0 \leqq i \leqq m$ (index s is usually omitted). In symbolics we shall pursue the same convention as in § 2: Each new value of an index will pass to all quantities involved in the process; however, the test (below) usually deals with it in its "old" fashion $\varkappa$ or i. — We shall start by describing a current iteration (in a form suitable for immediate coding).

BEGIN: Determine $\min\limits_{\iota \leqq i} \{\iota \mid \chi_\iota^{(\varkappa)} = 1\} = \iota_*$ and distinguish the possibilities:

(a) No minimum exists — there are no units among $\chi_0^{(\varkappa)}, \dots, \chi_i^{(\varkappa)}$.

(aα) If $i = m$, then put $F^s = F^{(\varkappa)}$, $G^s = G^{(\varkappa)}$, and $r_i^s = r_i^{(\varkappa)}$ $(i \in \mathscr{M})$. If $F^s = \emptyset$, $N^s \cup G^s = \mathscr{N}$, then go to DBL; else terminate the test.

(aβ) If $i < m$, increase the index i by 1. If then $\chi_i^{(\varkappa)} = -1$, go to BEGIN; else put

$$r_i^{(\varkappa+1)} = y_i^s - \sum a_{ij} \langle j \in ((N^s - G^{(\varkappa)}) \cap \mathscr{S}_i^-) \cup (F^{(\varkappa)} \cap \mathscr{S}_i^+) \rangle \,. \tag{3.1}$$

If $r_i^{(\varkappa+1)} < 0$, go to DBL; else go to BEGIN.

(b) $\iota_* = \lambda$. If $r_\lambda^{(\varkappa)} \geqq \omega_\lambda^{(\varkappa)}$, set $\chi_\lambda^{(\varkappa+1)} = 0$ and return to BEGIN; otherwise examine the λ-th row of the constraint matrix:

(bα) If for some $j = \mu \in N^s - (F^{(\varkappa)} \cup G^{(\varkappa)})$ the inequality $r_\lambda^{(\varkappa)} + a_{\lambda\mu} < 0$ holds, then put

$$F^{(\varkappa+1)} = F^{(\varkappa)} \cup \{\mu\}$$

and explore the μ-th column (for $\iota \leqq i$ such that $\chi_\iota^{(\varkappa)} \geqq 0$): If it is $a_{\iota\mu} > 0$, then

$$r_\iota^{(\varkappa+1)} = r_\iota^{(\varkappa)} - a_{\iota\mu}, \quad \chi_\iota^{(\varkappa+1)} = 1, \tag{3.2}$$

and if $r_\iota^{(\varkappa+1)} < 0$, go to DBL; else continue to examine the λ-th row.

(bβ) If for some $j = \mu \in N^s - (F^{(\varkappa)} \cup G^{(\varkappa)})$ the inequality $r_\lambda^{(\varkappa)} - a_{\lambda\mu} < 0$ holds, then put

$$G^{(\varkappa+1)} = G^{(\varkappa)} \cup \{\mu\}$$

and explore the μ-th column (for $\iota \leqq i$ such that $\chi_\iota^{(\varkappa)} \geqq 0$): If it is $a_{\iota\mu} < 0$, then

$$r_\iota^{(\varkappa+1)} = r_\iota^{(\varkappa)} + a_{\iota\mu}, \quad \chi_\iota^{(\varkappa+1)} = 1, \tag{3.3}$$

and if $r_\iota^{(\varkappa+1)} < 0$, go to DBL; else continue to examine the λ-th row.

After the λ-th row has been examined, set $\chi_\lambda^{(\varkappa+1)} = 0$ and go on back to BEGIN.

We start the iteration process with the value $i = -1$ (for the 0-th row to be included in it). The initial values of χ_i's may be chosen by one of the ways:

(1) $\chi_i^{(0)} = 1$ for all $i \in \mathscr{M}$,
(2) $\chi_i^{(0)} = 1$ if $y_i^s < 0$; else $\chi_i^{(0)} = -1$.

The parameters ω_i may be prepared, for instance:

(1) $\omega_i^{(\varkappa)} = \max\limits_{j \in \mathscr{N}} |a_{ij}|$,

(2) $\omega_i^{(\varkappa)} = \dfrac{\theta}{n} \sum\limits_{j \in \mathscr{N}} |a_{ij}| \quad (\theta > 0)$,

(3) $\omega_i^{(\varkappa)} = \max\limits_{j} \{|a_{ij}| \mid j \in N^s - (F^{(\varkappa)} \cup G^{(\varkappa)})\}$.

Remark 3. If it occurs $N^s - (F^{(\varkappa)} \cup G^{(\varkappa)}) = \emptyset$ during the test, we continue limiting ourselves to the evaluation of (3.1) – the remaining part of the test will be dummy. The test terminates in this case either by DBL after (3.1), or DBL when $N^s \cup G^s = \mathscr{N}$, or normally after performing all examinations (the combination $J^s \cup F^s$ is then feasible).

The evaluation of quantities r_i according to (3.1) is the most laborious part of Test BF. Therefore it was suggested in [17] that r_i should be derived *recursively*, by analogy to y_i's. Clearly, this requires that χ_i be chosen according to (1).

Also in [17] some numerical properties of Tets BF are demonstrated. The advantage of choosing χ_i according to (1) is particularly interesting, as well as the computation economy due to the recursive preparation of r_i's (up to 20%). Of course, a substantial improvement with regard to the results obtained elsewhere could not be expected.

Test BF stands for and generalizes almost all the tests of Balas' additive algorithm [1]. The quantities r_i (Fleischmann's notation) and the sets F have the corresponding meaning. The function of Balas' sets D is accomplished by our G. Level decreasing after obtaining a feasible solution in case of U1, $a_{0j} > 0$ $(j \in N^s)$ results from $N^s \cup G^s = \mathcal{N}$ and in case of U1', $a_{0j} \geqq 0$ $(j \in N^s)$ from $r_0^{(0)} < 0$. Only Balas' sets E have no counterpart in Test BF; but this is no loss because they produce a test of small effect since it generally eliminates only individual combinations (see about that [15]).

2. Test GZ. Let us have indices $p \in \mathcal{M}$, $q \in \mathcal{M}$ for which

$$y_p^s \geqq 0\,, \quad a_{pj} \geqq 0 \ (j \in N^s)\,,$$
$$y_q^s < 0\,,$$

and a set $H = N^s \cap \mathcal{S}_q^-$; let $H \neq \emptyset$[7]). If for all $j \in H$

$$a_{pj} y_q^s < a_{qj} y_p^s\,, \tag{3.4}$$

then go to DBL.

Justification. Let us assume that some combination J^t $(t > s)$ in a branch on J^s yields $y_q^t \geqq 0$. Denoting $J^t = J^s \cup D$ we can write

$$\sum_{j \in D} a_{qj} \leqq y_q^s - y_q^t \leqq y_q^s < 0 \tag{3.5}$$

where the first inequality expresses the possibility that between the s-th and the t-th iteration y_q will be reduced in Algorithm BG1 (after scoring a feasible solution when $q = 0$). Below, this circumstance is considered for the quantity y_p^t, too. From (3.5) it follows $D \cap H \neq \emptyset$ so that we can deduce

$$y_p^t \leqq y_p^s - \sum_{j \in D} a_{pj} \leqq y_p^s - \sum a_{pj} \langle j \in D \cap H \rangle =$$
$$= y_p^s - \frac{1}{y_q^s} \sum a_{pj} y_q^s \langle j \in D \cap H \rangle < y_p^s - \frac{1}{y_q^s} \sum a_{qj} y_p^s \langle j \in D \cap H \rangle \leqq 0\,.$$

The last but one inequality is a consequence of (3.4), and the last one follows from (3.5) because of

$$\sum a_{qj} \langle j \in D \cap H \rangle \leqq \sum_{j \in D} a_{qj}\,.$$

Thus, our initial assumption leads to $y_p^t < 0$.

---

[7]) We hope it causes no trouble if we do not specify how newly introduced symbols depend on the indices s and q.

Test GZ is constructed according to the principle (1β) as a mild generalization of a test suggested by Glover and Zionts in [12]. Their original test corresponds to GZ with $p = 0$.

Test GZ may be applied only to problems with both positive and negative elements among a_{ij}. (This is always the case in Problems U1 and U1′ if these are not trivial.)

For practical use of the test, of course, it is necessary to specify how to select the indices p, q. Let $M^{++} \subset \mathscr{M}$ and $M^{-} \subset \mathscr{M}$ stand for the sets of all indices of the types p and q, respectively. We introduce a suitable *ordering* on these sets and then take their elements in turn for testing, up to a prescribed number of them. We recommend an ordering based on monotonically increasing ratios $y_i^s / \bar{a}_i$ where $\bar{a}_i > 0$ is some magnitude characteristic of the elements a_{ij} $(j \in N^s)$, e.g.

(1) $\bar{a}_i = \left|\min\limits_{j \in N^s} a_{ij}\right| + \eta\,,$

(2) $\bar{a}_i = \sum\limits_{j \in N^s} |a_{ij}|\,.$

Here η is a small positive number guaranteeing the characteristic to be non-zero. (For $i \in M^{-}$ it suffices to consider only $N^s \cap \mathscr{S}_i^-$ instead of the whole N^s.) Such an ordering is motivated by the endeavour to test first the elements "more hopeful" with respect to the criterion (3.4). We use the following *heuristic* consideration (demonstrated for the characteristic (1)): In the ordering of M^{++} the elements p for which y_p^s is small and $\bar{a}_p$ large come first; all the more every a_{pj} $(j \in N^s)$ is large. Similarly for M^{-}, the elements q for which $|y_q^s|$ is large and

$$\bar{a}_q = \left|\min_{j \in N^s} a_{qj}\right| = \left|\min_{j \in H} a_{qj}\right| = -\min_{j \in H} a_{qj} =$$
$$= \max_{j \in H}\,(-a_{qj}) = \max_{j \in H} |a_{qj}|$$

is small come first; all the more every $|a_{qj}|$ $(j \in H)$ is small. The validity of this consideration was proved by numerical experiments [17].

Before Test GZ itself some *preliminary probe* can be inserted: Let us determine

$$\alpha_i = \begin{cases} \min\limits_j \{a_{ij} \mid j \in N^s\} & \text{for} \quad i \in M^{++}\,, \\ \min\limits_j \{a_{ij} \mid j \in N^s \cap \mathscr{S}_i^-\} & \text{for} \quad i \in M^{-}\,. \end{cases}$$

If some of the inequalities

$$\alpha_p y_q^s < \alpha_q y_p^s \quad (p \in M^{++},\ q \in M^{-}) \tag{3.4*}$$

holds, then the criterion (3.4) is satisfied, too. This preliminary probe can easily be performed even for all pairs of M^{++} and M^{-} elements. In case it does not "succeed" the α_i's can well serve for preparing the characteristics $\bar{a}_i$.

Remark 4. If Test GZ is placed after Test BF, then necessarily $H \neq \emptyset$ since otherwise Test BF would lead to DBL, and it is possible to use $N^s - G^s$ instead of N^s.

Another possibility for applying Test GZ is provided by *surrogate constraints* [10], [2], [9], [11], i.e. auxiliary constraints

$$\sum_{j\in\mathcal{N}} \Big(\sum_{i\in M} w_i a_{ij} \Big) x_j \leqq \sum_{i\in M} w_i b_i \quad (w_i \geqq 0,\ M \subseteq \mathcal{M}),$$

which are necessary conditions for the genuine constraints (1.1) to be satisfied. Now, we shall apply the test only to one pair of constraints in an iteration. These may be either one genuine and one surrogate constraint or both of them surrogate. For a "mixed" pair it is of advantage to have $p = 0$ since it can always be achieved $a_{0j} \geqq 0$ $(j \in \mathcal{N})$, $y_0^s \geqq 0$ by a transformation of the problem and earlier execution of Test BF. Moreover, the 0-th constraint has a considerable reduction effect for Problem U1. Surrogate constraints can be obtained, for instance, according to Geoffrion's method [9] or prepared as specially fitting for Test GZ. In the latter case, an extremum requirement follows from the nature of the test:

$$(3.6)\qquad \begin{aligned} &\sigma \to \min\,, \\ &\sum_{i\in M} a_{ij} w_i + \sigma \geqq 0 \quad (j \in N^s)\,, \\ &\sum_{i\in M} y_i^s\, w_i \qquad = \gamma\,, \\ &\qquad w_i \qquad \geqq 0 \quad (i \in M)\,. \end{aligned}$$

This is an *auxiliary linear programming problem.* The last but one constraint represents a normalization condition. To obtain a 'type q' surrogate constraint, we may set $\gamma = -1$. If $M \cap M^- \neq \emptyset$, the auxiliary problem is feasible. We construct a surrogate constraint using an optimal solution $(\bar{w}, \bar{\sigma})$ if $\bar{\sigma} > 0$; in other cases DBL may immediately follow since there exists a surrogate constraint infeasible in a branch on J^s $(\sum_i w_i a_{ij} \geqq 0$ for $j \in N^s$, $\sum_i w_i y_i^s < 0)$. To obtain a 'type p' constraint, we may set $\gamma = +1$. If $M \cap M^+ \neq \emptyset$ where $M^+ = \{i \in \mathcal{M} \mid y_i^s > 0\}$, the auxiliary problem is feasible. An unbounded solution implies DBL (there exist w, σ such that $\sum w_i a_{ij} > 1 = \sum w_i y_i^s$ for $j \in N^s$). A finite optimal solution is useful for us only if $\bar{\sigma} \leqq 0$ (in case of $\bar{\sigma} > 0$ some $\sum \bar{w}_i a_{ij}$ must be negative).

3. Test P1. Again let us have indices $p \in \mathcal{M}$, $q \in \mathcal{M}$ for which

$$\begin{aligned} &y_p^s \geqq 0\,, \quad a_{pj} \geqq 0 \;(j \in N^s)\,, \\ &y_q^s < 0\,, \end{aligned}$$

a set $H = N^s \cap \mathcal{S}_q^-$, and let v stand for the number of its elements. If $v = 0$, then go to DBL; else introduce an *ordering* $\{h_1, \ldots, h_v\}$ into H so that

$$(3.7)\qquad 0 \geqq \frac{a_{ph_1}}{a_{qh_1}} \geqq \frac{a_{ph_2}}{a_{qh_2}} \geqq \ldots \geqq \frac{a_{phv}}{a_{qhv}}\,.$$

Let us further denote $H^{(\tau)} = \{h_1, \ldots, h_\tau\}$. For $\tau = 1, 2, \ldots$ form successively sums

$$S_p^{(\tau)} = \sum_{j \in H^{(\tau)}} a_{pj}\,, \quad S_q^{(\tau)} = \sum_{j \in H^{(\tau)}} a_{qj}$$

and test:

(α_1) If $S_p^{(\tau)} \leqq y_p^s$, $S_q^{(\tau)} > y_q^s$, and $H - H^{(\tau)} = \emptyset$, then DBL.

(α_2) If $S_p^{(\tau)} \leqq y_p^s$, $S_q^{(\tau)} > y_q^s$, and $H - H^{(\tau)} \neq \emptyset$, then new τ.

(β) If $S_p^{(\tau)} > y_p^s$, $S_q^{(\tau)} \geqq y_q^s$, then DBL.

In other possible cases terminate the test for the given p and q.

Justification. Only case (β) is to be dealt with. Let us assume that a combination $J^t = J^s \cup D$ $(t > s)$ exists in a branch on J^s for which $y_q^t \geqq 0$. We are going to show that this assumption implies $y_p^t < 0$. Let us denote the sets

$$\begin{aligned} D^- &= D \cap \mathscr{S}_q^-\,, \quad D^+ = D - D^-\,, \\ K &= H^{(\tau)} - \left(H^{(\tau)} \cap D^-\right), \\ L &= D^- - \left(H^{(\tau)} \cap D^-\right). \end{aligned}$$

Obviously,

$$D^- \neq \emptyset\,, \quad D^- \subseteq H\,, \quad L \subseteq H - H^{(\tau)}\,, \quad K \cap L = \emptyset\,. \tag{3.8}$$

Now we consider three possibilities:

(1) $K = \emptyset$, i.e. $H^{(\tau)} \cap D^- = H^{(\tau)}$. For the increment of y_p^s we may write

$$\begin{aligned} \sum_{j \in D} a_{pj} &\geqq \sum_{j \in D^-} a_{pj} = \sum a_{pj} \langle j \in L \cup \left(H^{(\tau)} \cap D^-\right)\rangle = \\ &= \sum a_{pj} \langle j \in L \cup H^{(\tau)}\rangle \geqq \sum_{j \in H^{(\tau)}} a_{pj} = S_p^{(\tau)} > y_p^s \end{aligned}$$

so that $y_p^t \leqq y_p^s - \sum_{j \in D} a_{pj} < 0$. (The first inequality respects a possible reduction of the quantity y_p after reaching a feasible solution in Algorithm BG1; similarly for y_q^t below.)

(2) $K \neq \emptyset$, $L \neq \emptyset$. It holds

$$\sum_{j \in H^{(\tau)}} a_{qj} = S_q^{(\tau)} \geqq y_q^s \geqq y_q^s - \sum_{j \in D^+} a_{qj} \geqq \sum_{j \in D^-} a_{qj}$$

where the last inequality follows from our assumption about y_q^t. Hence we can obtain

$$\sum_{j \in K} a_{qj} \geqq \sum_{j \in L} a_{qj}\,. \tag{3.9}$$

For all pairs of $\varkappa \in K$ and $\lambda \in L$, owing to (3.8) and (3.7), it is

$$\frac{a_{p\varkappa}}{a_{q\varkappa}} \geqq \frac{a_{p\lambda}}{a_{q\lambda}}$$

so that we have

$$\sum_{\lambda\in L} a_{q\lambda} \cdot \sum_{\varkappa\in K} a_{p\varkappa} \geqq \sum_{\lambda\in L} a_{p\lambda} \cdot \sum_{\varkappa\in K} a_{q\varkappa} \, .$$

This, with regard to (3.9), gives

$$\sum_{\lambda\in L} a_{p\lambda} \geqq \sum_{\varkappa\in K} a_{p\varkappa} \, .$$

The increment of y_p^s yields, similarly to (1),

$$\sum_{j\in D} a_{pj} \geqq \sum_{j\in L} a_{pj} + \sum a_{pj} \langle j \in H^{(\tau)} \cap D^- \rangle \geqq$$

$$\geqq \sum_{j\in K} a_{pj} + \sum a_{pj} \langle j \in H^{(\tau)} \cap D^- \rangle = \sum_{j\in H^{(\tau)}} a_{pj} = S_p^{(\tau)} > y_p^s \, .$$

(3) $K \neq \emptyset$, $L = \emptyset$. This possibility does not occur because it implies $D^- \subset H^{(\tau)}$ (sharp!), which contradicts the assumption as it may be seen from

$$y_q^t \leqq y_q^s - \sum_{j\in D} a_{qj} \leqq y_q^s - \sum_{j\in D^-} a_{qj} < y_q^s - \sum_{j\in H^{(\tau)}} a_{qj} = y_q^s - S_q^{(\tau)} \leqq 0 \, .$$

Test P1 rests on the principles (1). It appears to be a slight generalization of Petersen's test [20] which, in our terms, corresponds to $p = 0$.[8]) Judging from some practical results, Petersen's test represents an effective reduction means even if not always (compare [22]).

The execution of Test P1 might proceed similarly to Test GZ: The indices p, q are selected from the sets M^{++}, M^- ordered according to the same considerations as for Test GZ. Notice that a uniform adjustment of coefficients of a surrogate constraint, as it is supplied by the auxiliary problem (3.6), is a suitable compromise with respect to "antagonism" between the ordering (3.7) and the criterion (β) of the test. Thus in this situation, when we shall work only with a limited number of judiciously selected pairs of constraints, it will not be an unrealistic task to perform the ordering (3.7) currently in iterations (thereby saving a great deal of computer memory).

Remark 5. It may be easily seen that the validity of criterion (β) for $\tau = 1$ implies the validity of (3.4) in Test GZ.

Remark 6. If Test P1 is placed after Test BF, then always $H \neq \emptyset$ and $N^s - G^s$ can be used instead of N^s.

Conclusion. The elementary means that are used in Balasian enumerative algorithms should not be overvalued. It would not be very reasonable if one wanted to solve problems (of general type) with $n > 50$. The best way of making these methods more powerful seems to be the Glover's and Geoffrion's idea of surrogate constraints [11], [9]. This may be added to an enumerative algorithm as an acceleration facility.

---

[8]) Except for a not very much effective part for creating sets of type F. This may be left out because the test is intended to be used along with Test BF. The name 'P1' relates to [15].

*References**)

[1] *Balas, E.:* An additive algorithm for solving linear programs with zero-one variables. Operations Research *13* (1965), No 4, 517—546.

[2] *Balas, E.:* Discrete programming by the filter method. Operations Research *15* (1967), No 5, 915—957.

[3] *Bertier, P., Nghiem, P. T.:* Résolution de problèmes en variables bivalentes (Algorithme de Balas et procédure SEP). SEMA, Paris, Note D.S. No 33, 1965.

[4] *Brauer, K. M.:* A note on the efficiency of Balas' algorithm. Operations Research *15* (1967), No 6, 1169—1171.

[5] *Byrne, J. L., Proll, L. G.:* Initialising Geoffrion's implicit enumeration algorithm for the zero-one linear programming problem. The Computer Journal *12* (1969), No 4, 381—384.

[6] *Fleischmann, B.:* Computational experience with the algorithm of Balas. Operations Research *15* (1967), No 1, 153—155.

[7] *Freeman, R. J.:* Computational experience with a 'Balasian' integer programming algorithm. Operations Research *14* (1966), No 5, 935—941.

[8] *Geoffrion, A. M.:* Integer programming by implicit enumeration and Balas' method. SIAM Review *9* (1967), No 2, 178—190.

[9] *Geoffrion, A. M.:* An improved implicit enumeration approach for integer programming. Operations Research *17* (1969), No 3, 437—454.

[10] *Glover, F.:* A multiphase-dual algorithm for the zero-one integer programming problem. Operations Research *13* (1965), No 6, 879—919.

[11] *Glover, F.:* Surrogate constraints. Operations Research *16* (1968), No 4, 741—749.

[12] *Glover, F., Zionts, S.:* A note on the additive algorithm of Balas. Operations Research *13* (1965), No 4, 546—549.

[13] *Golomb, S. W., Baumert, L. D.:* Backtrack programming. Journal ACM *12* (1965), No 4, 516—524.

[14] *Gue, R. L., Liggett, J. C., Cain, K. C.:* Analysis of algorithms for the zero-one programming problem. Communications ACM *11* (1968), No 12, 837—844.

[15] *Hrouda, J.:* Jeden popis Balasova aditivního algoritmu se zřetelem k programování. Ekonomicko-matematický obzor *5* (1969), No 1, 45—59.

[16] *Hrouda, J.:* Staging in Balas' algorithm. This issue, 354—369.

[17] *Hrouda, J.:* Tři příspěvky k bivalentnímu lineárnímu programování. Příloha k výzkumné zprávě VZ-321/70, VÚTECHP, Praha 1970.

[18] *Корбут, А. А., Финкельштейн, Ю. Ю.:* Дискретное программирование. Наука, Москва 1969.

[19] *Lemke, C. E., Spielberg, K.:* Direct search algorithms for zero-one and mixed-integer programming. Operations Research *15* (1967), No 5, 892—914.

[20] *Petersen, C. C.:* Computational experience with variants of the Balas algorithm applied to the selection of R & D projects. Management Science *13* (1967), No 9, 736—750.

[21] *Walker, R. J.:* An enumerative technique for a class of combinatorial problems. Proceedings of Symposia in Applied Mathematics, vol. 10 (eds. R. Bellman, M. Hall), Providence 1960, 91—94.

[22] Výzkumná zpráva VZ-124/68 (řešitel J. Hrouda). VÚTECHP, Praha 1968.

[23] Výzkumná zpráva VZ-211/69 (řešitel J. Tesař). VÚTECHP, Praha 1969, 42—46.

*) A great deal of bibliography can be found in [2], [14], and [18].

Souhrn

PŘÍSPĚVEK K BALASOVU ALGORITMU

JAROSLAV HROUDA

Článek sleduje dvojí cíl: (1) Podat alternativní popis algoritmu vhodný jako propedeutika k článku [16], v němž provedeme jisté zobecnění Balasova algoritmu. (2) Usoustavnit a zobecnit některé starší testy.

V § 1 stručně popisujeme enumerační proces typu ‚backtracking' včetně způsobu jeho redukce a evidence.

V § 2 pod označením ‚algoritmus BG' presentujeme modifikaci Balasova, resp. Geoffrionova algoritmu (bez testů). K formálnímu zjednodušení přispívá pojetí účelové funkce jako 0-tého omezení s proměnnou pravou stranou. Rozlišení versí algoritmu — pro získání všech optimálních řešení (BG1) a nejvýše jednoho optimálního řešení (BG1') — se pak docílí úpravou jediného místa algoritmu. Nepožaduje se nezápornost koeficientů účelové funkce, neboť to nemusí být vždy výhodné. Gloverův způsob evidence enumeračního procesu je poněkud upraven.

V § 3 uvádíme tři poměrně silné testy, jež vznikly zobecněním starších známých testů: (1) Test BF (zobecněný Balas-Fleischmannův) pracuje důsledně s dvojicí množin F, G zahrnujících prvky, jež musí být, resp. nesmí být přítomny k dosažení přípustnosti v dané větvi. Množiny F, G jsou v našem případě konstruovány jako maximálně početné svého druhu. (2) Test GZ (zobecněný Glover-Ziontsův) se již výlučně neváže na účelovou funkci. Je k němu vypracován heuristický postup pro stanovení vhodného pořadí, v němž omezení úlohy mají vstupovat do testu, a navrženo použití zástupných omezení. (3) Test P1 (zobecněný Petersenův) aplikuje touž zobecňovací ideu i heuristiku jako GZ.

Author's address: Jaroslav Hrouda, prom. mat., Výzkumný ústav technicko-ekonomický chemického průmyslu, Štěpánská 15, Praha 2.