Balas algorithm for linear 0-1 programming

Giovanni Righini

Operations Research Complements

UNIVERSITÀ DEGLI STUDI DI MILANO

References:
E. Balas, *An Additive Algorithm for Solving Linear Programs with Zero-One Variables*, Operations Research 13, 4 (1965) 517-546.
F. Glover, S. Zionts, *A Note on the Additive Algorithm of Balas*, Operations Research 13, 4 (1965) 546-549.

Linear 0-1 programming

Consider a linear 0-1 programming problem P such as

$$\begin{aligned} \text{minimize } z = \sum_{j \in N} c_j x_j & \\ \text{s.t. } \sum_{j \in N} a_{ij} x_j \le b_i & \qquad \forall i \in M \\ x_j \in \{0, 1\} & \qquad \forall j \in N \end{aligned}$$

where M is the set of rows, N is the set of columns and x is a binary solution vector.

It is not necessary to assume that the components of A, b and c are integers.

Balas algorithm

Balas algorithm is a branch-and-bound algorithm, also known as "additive algorithm", because it requires only additions and subtractions but no multiplications or divisions.

It can be described by

- branch strategy (branching on a binary variable)
- search strategy
- fathoming rules
- variable fixing rules
- lower bounding
- upper bounding

Standard form

First, the problem must be put in a standard form, so that all components of c have the same sign: either minimization of z with $c \geq 0$ or maximization of z with $c \leq 0$.

Let select the first option.

If $c_j < 0$ for some $j \in N$, then replace x_j with $1 - \hat{x}_j$ everywhere in P.

The constant terms that are generated in the objective function are disregarded and added back to the final optimal value.

Example.

minimize $z = 3x_1 - 5x_2 + x_3 - 2x_4 \Rightarrow$ minimize $z = 3x_1 - 5 + 5\hat{x}_2 + x_3 - 2 + 2\hat{x}_4$.

Standard form

Then, all constraints must be written in the same form, either as $\geq$ or as $\leq$ (let select $\leq$, for instance).

Equality constraints are replaced by pairs of inequality constraints.

Example.

$$\begin{cases} 3x_1 - 2x_2 + 8x_3 \leq 7 \\ 4x_1 + 6x_2 - 5x_3 \geq 5 \\ 2x_1 - 3x_2 - 4x_3 = -5 \end{cases}$$

$$\begin{cases} 3x_1 - 2x_2 + 8x_3 \leq 7 \\ -4x_1 - 6x_2 + 5x_3 \leq -5 \\ 2x_1 - 3x_2 - 4x_3 \leq -5 \\ -2x_1 + 3x_2 + 4x_3 \leq 5 \end{cases}$$

Partial solution and 0-completion

While developing a branch-and-bound binary tree, at each node of the tree, some binary variables have been fixed to 0, some to 1 and some are still free.

They identify a partial solution, S.

We indicate by $N^0(S)$, $N^1(S)$ and $N^{free}(S)$ their sets, respectively: $N = N^0(S) \cup N^1(S) \cup N^{free}(S)$ for any S in the tree.

The 0-completion of a partial solution S is the solution obtained by setting all variables in $N^{free}(S)$ to 0.

It is the best solution of the subtree rooted at S, since $c \geq 0$ (but it may be infeasible): its value, $z^0(S)$, is a valid lower bound for the subtree.

Feasibility

For each partial solution S we have a corresponding sub-problem whose constraints are

$$\sum_{j \in N^0(S)} a_{ij} x_j + \sum_{j \in N^1(S)} a_{ij} x_j + \sum_{j \in N^{free}(S)} a_{ij} x_j \leq b_i \;\; \forall i \in M$$

i.e.

$$\sum_{j \in N^{free}(S)} a_{ij} x_j \leq b_i - \sum_{j \in N^1(S)} a_{ij} \;\; \forall i \in M.$$

Let define the right-hand-sides of S as

$$r_i(S) = b_i - \sum_{j \in N^1(S)} a_{ij}$$

so that the constraints are

$$\sum_{j \in N^{free}(S)} a_{ij} x_j \leq r_i(S) \;\; \forall i \in M.$$

If $r_i(S) \geq 0 \; \forall i \in M$, then the 0-completion of S is feasible: $z^0(S)$ is an upper bound.

Upper bounding

A best incumbent value $\overline{z}$ is kept at all times during the execution of the algorithm.

Initially $\overline{z}$ is set to a very large value: $\overline{z} \leftarrow +\infty$.

Every time the feasibility test on the 0-completion of a partial solution S succeeds, a new upper bound $z^0(S)$ is found and it is compared with the best incumbent.

If $z^0(S) < \overline{z}$, then the best incumbent value is updated: $\overline{z} \leftarrow z^0(S)$.

When the 0-completion of S is feasible, then S is solved at optimality and no branching occurs.

Fathoming

Feasibility. Let

$$t_i(S) = \sum_{j \in N^{free}(S)} \min\{0, a_{ij}\}$$

i.e. the minimum value that the left-hand-side of constraint i can take in S.

If $\exists i \in M$ such that $t_i > r_i(S)$, then S is infeasible and its node is fathomed.

Optimality. If the cost of the 0-completion of S is larger than the best incumbent upper bound, i.e.

$$z^0(S) \geq \overline{z},$$

then no optimal solution can be obtained from any completion of S; hence, its node is fathomed.

Refined fathoming

Glover and Zionts (1965) provided an even weaker sufficient condition that allows for fathoming a node S.

If $\exists i \in M$ such that $r_i(S) < 0$ and

$$r_i(S)\frac{c_j}{a_{ij}} \geq \overline{z} - z^0(S) \;\; \forall j \in N^{free}(S) : a_{ij} < 0,$$

then constraint i cannot be satisfied without making the objective function greater than or equal to the value of the best incumbent upper bound.

Hence no completion of S can be feasible and optimal; the node can be fathomed.

Multiple variables fixing

If a sub-problem S is not fathomed, then it is analyzed to possibly fix some variables and reduce its size.

If $\exists i \in M$ such that $t_i = r_i(S)$, then feasibility requires

- $x_j = 0 \; \forall j \in N^{free}(S) : a_{ij} > 0$
- $x_j = 1 \; \forall j \in N^{free}(S) : a_{ij} < 0$.

So, all variables $j \in N^{free}(S) : a_{ij} \neq 0$ can be fixed (and fathoming tests are re-done).

Otherwise, single variables are examined (in any sequence) to be possibly fixed.

Single variable fixing

Feasibility.
If $\exists i \in M, j \in N^{free}(S) : a_{ij} > 0 \wedge t_i(S) + a_{ij} > r_i(S)$, then x_j can be fixed to 0 (for each $i \in M$, Balas suggests to examine variable x_j for which a_{ij} is maximum).

If $\exists i \in M, j \in N^{free}(S) : a_{ij} < 0 \wedge t_i(S) - a_{ij} > r_i(S)$, then x_j can be fixed to 1 (for each $i \in M$, it is profitable to consider x_j for which a_{ij} is minimum).

Optimality.
Every variable $j \in N^{free}(S)$ such that $z^0(S) + c_j \geq \overline{z}$ must be set to 0 in any completion of S with a value better than the best incumbent $\overline{z}$.

Refined single variable fixing

Feasibility and optimality. Glover and Zionts (1965) refined Balas' criterion.

Consider a variable $j \in N^{free}(S)$ and a constraint $i \in M$ such that $a_{ij} > r_i(S)$: setting $x_j = 1$ requires setting to 1 some other variable with $a_{ij} < 0$ in order to achieve feasibility.

This implies at least a cost $c_p = \min\{c_k : k \in N^{free}(S) \wedge a_{ik} < 0\}$. Hence, if

$$z^0(S) + c_j + c_k \geq \overline{z},$$

then variable x_j can be set to 0.

If no variable is left after fixing, than S is solved and no branching is required.

Promising vectors

All constraints with $r_i(S) < 0$ are violated in the 0-completion of S.

To achieve feasibility we need to set some free variable x_j to 1, among the variables with $a_{ij} < 0$ in those constraints.

If there are no such variables, there is no way to achieve feasibility by completing S.

Let define

$$V(S) = \{i \in M : r_i(S) < 0\}$$

$$A(S) = \{j \in N^{free}(S) : a_{ij} \geq 0 \ \forall i \in V(S)\}$$

When completing S, variables in $A(S)$ do not help to achieve feasibility.

Promising vectors are those in $R(S) = N^{free}(S) \setminus A(S)$.
If $R(S) = \emptyset$, then S is infeasible and its node is fathomed.

Branch policy and search policy

When S is not fathomed and its 0-completion is infeasible, it is necessary to branch.

A branching variable is selected among the promising vectors and it is set to 0 in one branch and to 1 in the other.

The 1-branch is explored first, the 0-branch second.

When a node is fathomed, the algorithm backtracks: hence, the branch-and-bound tree is explored with a depth-first-search policy.

Selection of the branching variable

The branching variable is selected as the variable that yields the 0-completion with minimum total infeasiblity when it is fixed to 1.

Define

$$I_j(S) = \sum_{i \in M} \max\{0, a_{ij} - r_i(S)\}.$$

The value $a_{ij} - r_i(S)$, when it is positive, is a measure of the residual violation of constraint i when x_j is set to 1.

The branching variable x_k is selected so that

$$k = \arg \min_{j \in R(S)} \{I_j(S)\}.$$