Bull. Nov. Comp. Center, Comp. Science, 10 (1999), 93–99

Numerical experiments with one ILP algorithm*

G.I. Zabinyako

In this paper, a branch and bound algorithm with branching in depth for problems of the integer linear programming (ILP) is considered. The results of solution to large problems of integer and mixed-integer linear programming are presented. A brief information is given on the software for solution to large ILP problems with sparse matrices.

1. The algorithm

Let a problem of the ILP be given in the form:

find

$$\max f(x) = (c, x)$$

with constraints

$$Ax = b, \qquad \alpha \leq x \leq \beta,$$

where $c, x, \alpha, \beta \in R^n$ and $b \in R^m$ are vectors, A is $m \times n$ matrix, and the components x_j of the vector x are integer numbers.

One also may consider the components of the vectors α and β to be integers too.

The algorithms with in-depth branching are attractive because they make it possible to construct compact lists of estimation problems [1]. We consider an algorithm in which any problem is described by five elements of the array.

The choice of a variable for branching is made by analyzing the penalties [1, 2] for increasing $\bar{\bar{P}}_j$ (decreasing $\bar{P}_j$) of the basic non-integer variable x_j to an integer value. In the scheme given below, the array h corresponds to the list of problems.

P0. Let $i = 0$, $k = 0$ and the value of the incumbent $r^0 = -\infty$, $\alpha^0 = \alpha$, $\beta^0 = \beta$.

P1. Solve the current problem of LP.

---

*Supported by the Russian Foundation for Basic Research under Grant 99-07-90422.

(a) If $f^i \leq r^i$ (f^i is an optimal value of the objective function) or the system of constraints is inconsistent, then assume $r^{i+1} = r^i$, $i = i + 1$ and go to P3.

(b) The integer solution is consistent. Let $r^{i+1} = f^i$, $i = i + 1$. If $f^i = f^0$, then go to P5, else go to P3.

P2. Calculate penalties. Among the penalty estimations for basic non-integer variables find the minimal $P_{\min}$.

(a) If $f^i - P_{\min} \leq r^i$, then go to P3.

(b) Execute branching for the basic variable j that is matched by the maximum penalty $\bar{P}_j$ or $\bar{\bar{P}}_j$. For consistency let $\bar{P}_j \leq \bar{\bar{P}}_j$. Set a variable

$$l = \begin{cases} 1, & \text{if } f^i - \bar{\bar{P}}_j \leq r^i, \\ 0, & \text{otherwise.} \end{cases}$$

Increase $k = k + 1$. Then perform the assignments

$$h(1,k) = j, \quad h(2,k) = \beta_j^i, \quad h(3,k) = l,$$
$$h(4,k) = f^i, \quad h(5,k) = \bar{\bar{P}}_j.$$

Set the upper boundary of the variable j equal to $[x_j^i]$, go to P1. If $\bar{\bar{P}}_j < \bar{P}_j$, then perform the assignments

$$h(1,k) = -j, \quad h(2,k) = \alpha_j^i, \quad h(3,k) = l,$$
$$h(4,k) = f^i, \quad h(5,k) = \bar{P}_j.$$

Here the parameter l is defined as mentioned above and the lower boundary for the variable j is taken to be equal $[x_j^i] + 1$.

P3. If $k = 0$, then go to P5. Select $l = h(3,k)$.

(a) If $l = 1$, then go to P4. If $F - P \leq r^i$, where $F = h(4,k)$ and $P = h(5,k)$, then go to P4.

(b) Using the array h, construct an alternative problem (which is alternative to the problem formed in P2.b), assign $h(3,k) = 1$ and go to P1.

P4. Restore the values of constraints on the variable x_j using the information from the array h ($h(1,k)$ and $h(2,k)$). Assign $k = k - 1$ and go to P3.

P5. Stop.

During the calculation of the penalties (P2) there may occur a degenerate case when the maximum value among all $\bar{P}_j$ and $\bar{\bar{P}}_j$ is equal to zero. In this case branching is performed using the variable x_j selected among the basic ones as it has a value most deviating from integer.

In the schemes with one-sided branching, one can realize an efficient passage to the next estimation problem. One should note two cases arising in the formation of the problem (P2.b):

1) the variable x_j is assigned a fixed value $x_j = \alpha_j^i = \beta_j^i$;

2) the variable must vary in the ranges $\alpha_j^i \leq x_j \leq \beta_j^i$ and $\alpha_j^i < \beta_j^i$.

In the first case, a fictitious variable x_ν is introduced to the basis (the optimal basis of the preceding problem with the basic matrix B) instead of the variable x_j. For this purpose a vector $z^T = e_p^T B^{-1}$ is calculated, where e_p is the p-th ort in R^m and p is the number assigned to the variable x_j in the basis. A number k is found, on which the maximum of $|z_i|$ for $i = 1, \ldots, m$ is reached. The ort k from R^m corresponds to the variable $x_\nu(\nu = n + k)$. Then the matrix B^{-1} is corrected in the standard manner with allowance for the substitution of the variable x_ν for the variable x_j in the basis, and the values of basic variables are calculated in a new basis. In the solution, if such exists in the estimation problem, x_ν is equal to zero.

In the second case, there is no need to correct B^{-1}. One always starts a solution of any estimation problem with eliminating the infeasibility of the values of basic variables. For this purpose the expression

$$\Phi(x) = \sum_{x_j < \alpha_j^i} (\alpha_j^i - x_j) + \sum_{x_j > \beta_j^i} (x_j - \beta_j^i)$$

is minimized on the set $Ax = b$. While eliminating the infeasibility of basic variables, a special method [3] of the shift value calculation on the simplex method iterations is used which makes it possible to minimize a piecewise linear function $\Phi(x + \lambda s)$ in λ along a selected direction.

After the elimination of the infeasibility of basic variables in defining the value of a shift the concept of "an expanding sequence of tolerances" [4] is used. For any variable x_j the fulfillment of the conditions $\alpha_j - \delta \leq x_j \leq \beta_j + \delta$ is controlled, where the value of a parameter δ is increasing by a small amount $\tau > 0$ at each iteration of the simplex method.

2. Solution to problems

To test the algorithm, the problems of integer and mixed-integer LP from various fields of application were solved. All the problems were obtained on the electronic media in the MPS-format (for the format see, e.g. [2]).

Table 1

№	The problem name	m	m_e	n	n_i	n_b	n_z
1	air01	23	23	771	771	771	4215
2	air02	50	50	6774	6774	6774	61555
3	air03	124	124	10757	10757	10757	91028
4	air05	310	310	6200	6200	6200	32740
5	boeing1	351	98	473	150	16	3574
6	boeing2	166	23	162	50	–	1215
7	cracpb1	143	90	573	572	572	4159
8	dsbmio	1182	343	1928	183	160	7417
9	gen	780	150	870	150	144	2592
10	modglob	291	95	422	98	98	968

In Table 1, a description of the problems is presented, where m is the number of constraints of which m_e is that on equalities; n is the number of variables (columns) of which n_i is that of integer, and n_b is that of the Boolean variables; nz is the total number of nonzero elements in the matrix A. In all the problems it is necessary to determine the minimum of the objective function. In addition to the data in the MPS-format, the electron media contains the comments indicating to the field of application and the authorship. In Problems 1–4, it is required to find an optimal time–table of flights for the crews (G. Astfalk). Problems 5 and 6 are the engineering problems on aircraft construction (N. Gould); these two problems are included in the library of NETLIB tests [5]. The application field of Problem 7 is unknown (N. Growder, E.A. Boyd). Problem 8 optimizes the production, processing, and distribution of gas (J.J.H. Forrest). Problem 9 consists in determining the time–table of operation for technical devices (L.A. Wolsey, M.W.P. Savelsbergh), and Problem 10 refers to designing heating systems (L.A. Wolsey, M.W.P. Savelsbergh).

Table 2 presents the results of solutions to the problems. The problems were solved in two variants which differ in the way of constructing the initial basis in the original LP problems with no allowance for integers [6]. In Table 2, the following denotations are used:

N is the problem number;

it is the total number of iterations;

it_0 is the number of the iterations at which the solution to the LP problem was obtained with no allowance for integers;

it_s is the number of the iterations at which an optimal integer solution was obtained;

k is the maximum depth of the tree of variants (the maximum value of the index k in the array h);

Table 2

№	it	it_0	it_s	k	k_s	k_i	P1	P2	t
				Initial basis $B0$					
1	75	42	75	1	2	1	1	–	0.84
2	798	262	564	5	7	1	4	–	26.2
3	1319	843	1319	1	2	1	1	–	95.7
4*	1000000	10076	431822	69	102	2	2	–	20412.0
5*	1000000	879	664568	185	43879	36	22452	351	3237.0
6	180014	206	104713	48	11683	7	4754	53	358.9
7	2520	766	2520	56	57	1	–	–	11.28
8	29107	1519	29107	46	140	11	32	–	616.3
9	113226	2152	55178	34	670	6	79	–	1088.0
10*	1000000	170	990149	96	30613	115	11169	145	1983.0
				Initial basis $B1$					
1	76	60	76	1	2	1	1	–	0.68
2	51449	230	51262	42	345	2	28	1	698.8
3	1750	1076	1750	1	2	1	1	–	43.38
4*	1000000	8176	386851	68	69	1	–	–	22243.0
5*	1000000	1069	707371	189	58126	14	29562	342	3534.0
6	285923	159	208334	50	17836	6	6951	62	561.9
7	3854	678	3854	57	77	2	3	–	19.36
8	14097	2949	14097	48	503	15	43	–	4783.0
9	148803	1945	71621	38	1009	11	92	–	1390.0
10*	1000000	168	170932	89	39656	65	16971	41	2150.0

k_s is the total number of formulated LP problems;

k_i is the number of solutions obtained with the integer requirement fulfilled;

$P1$ shows how many times the conditions $f^i - \bar{\bar{P}}_j \leq r^i$ or $f^i - \bar{P}_j \leq r^i$ were fulfilled during the solution to a problem (see P2);

$P2$ shows how many times the condition $f^i - P_{\min} \leq r^i$ was fulfilled;

t is the time of solution to a problem in seconds on Silicon Graphics (the computer of capasity with a clock rate 75 MHz).

In Table 2, the problems in which no proof has been obtained for the optimality of an integer solution are marked with the asterisk (*).

3. Analysis of peculiarities

In addition to the above-mentioned problems some standard small problems were solved for testing the programs. The process of solving the latter problems does not strongly depend on the way of constructing the initial basis and on changing the values of controlling parameters in sufficiently large ranges. The differences in Table 2 for various initial bases are essentially due to the *non-uniqueness of solutions* of the original LP problems with no allowance for integers. For instance, in Problem 7, the optimal solution with no allowance for integers has relative estimations $d_j = 0$ for 430 non-basic variables.

When solving Problem 8 there are *ill-posed basic matrices.* A standard set of values of controlling parameters which was used in the problems did not permit obtaining any solution with allowance for integers. In order to obtain the solution one had to make a special choice of controlling parameters which gave a higher accuracy of solution of the LP estimation problems.

In addition to the above-mentioned problems, two other problems air04 and air06 with a higher dimensionality than that of air05 were available. One could not nevertheless solve these problems with the reasonable expense of time. The peculiarity of these problems consists in that the *basic solutions are strongly degenerate.* In optimal solution with no allowance for integers and in the intermediate iterations several hundreds basic variables have zero values (in these problems all $\alpha_j = 0$).

4. Software

The software for solving large ILP problems with sparse matrices may be used on computers IBM PC in the environment DOS and on Silicon Graphics in the environment IRIX in the programming language Fortran-77. In addition to procedures of numerical computation, the software contains auxiliary programs for data processing. These programs provide data input, their logical control, composing dictionaries and reference tables, conversion from the outer MPS-format [2] into the inner format (used in procedures of numerical computations), and composing the output forms. The versions of batches for IBM PC and Silicon Graphics differ in their auxiliary programs which is due to the difference of their memory organization. Moreover, the batches for IBM PC contain procedures for tracing the course of program solution on the display, while Silicon Graphics has no such means.

A special file for determining controlling parameters is provided in programs. Using the corresponding values of controlling parameters one can modify the scheme of program solution. The user may define an estimation for the value of the object function thus diminishing the search. The relaxation problem being solved, f^i is compared with $\tilde{f}$ and the problem is not applied any more as a parent one if $f^i < \tilde{f}$.

For obtaining approximate solutions, an input parameter δ_f may be provided. The problem solution process is completed if for the current incumbent r^i the inequality $(f^0 - r^i)/|f^0| \leq \delta_f$ holds.

For those problems, in which general constraints have only the form of inequalities, one can apply a local algorithm, based on a coordinatewise search.

The software is discussed in more details in [6].

References

[1] Kovalev M.M. Discrete optimization (integer programming). – Minsk: Izd. Belorusk. universiteta, 1977 (in Russian).

[2] Murtagh B.A. Advanced Linear Programming. Computation and Practice. – Mc Graw-Hill International, Book Company, 1981.

[3] Fourer R. A simplex algorithm for piecewise-linear programming I: derivation and proof // Mathematical Programming. – 1985. – Vol. 33. – P. 204–233.

[4] Gill P.E., Murray W., Saunders M.A., Wright M.H. A practical anti-cycling procedure for linearly constrained optimization // Mathematical Programming. – 1989. – Vol. 45, № 3. – P. 437–474.

[5] Gay D.M. Electronic mail distribution of linear programming test problems // Mathematical Programming Society COAL Newsletter. – 1985. – Vol. 13. – P. 10–12.

[6] Zabinyako G.I. Program package of integer linear programming. – Novosibirsk, 1998. – (Report / ICMMG SO RAN; GR 01.9.30 001317, 02.9.80 005512) (in Russian).