Journal of Computational Mathematics, Vol.17, No.5, 1999, 545–552.

CALCULATION OF PENALTIES IN ALGORITHM OF MIXED INTEGER PROGRAMMING SOLVING WITH REVISED DUAL SIMPLEX METHOD FOR BOUNDED VARIABLES*1)

Yi-ming Wei
(*Institute of Policy and Management, CAS, Beijing 100080, P.R. China*)

Qing-huai Hu
(*Wuhan Institute of Chemical Technology, Wuhan 430073, P.R. China*)

Abstract

The branch-and-bound method with the revised dual simplex for bounded variables is very effective in solving relatively large-size integer linear programming problems. This paper, based on the general forms of the penalties by Beale and Small and the stronger penalties by Tomlin, describes the modifications of these penalties used for the method of bounded variables. The same examples from Petersen are taken and the satisfactory results are shown in comparison with those obtained by Tomlin.

Key words: Penalties, Stronger penalties, The revised dual simplex method for bounded variables.

1. Introduction

The studies on the branch-and-bound algorithm of integer programming have been carried out since 60's. The efforts in improving the algorithm are mainly concentrated on speeding up the related LP solution for each node and making better selection of node and branch for examing in order to approach the optimal solution as quick as possible. As a better strategy to estimate the problem bound and to select branch, Beals and Small proposed the penalties in 1965[2], and then Tomlin made modifications or extensions in 1969 by criterion for abandoning unprofitable branches. This stronger criterion is obtained by making use of Gomory cutting-plane constraints. These modifications have been incorporated into the famous UMPIRT system and are used successfully to solve many large practical mixed integer programming problems[3,7]. Moreover in aspect of speeding up solution of the LP problem for each node selected, the author would recommend the revised dual simplex method for bounded variables in which a branch is treated by introducing an additive bounded restriction as that with a lower or upper bound change. Thus a procedure of sensitivity analysis to this change is carried out in a continuous way based on the present basis inverse. The method works

* Received October 8, 1997.

1) This work was supported by the Chinese Postdoctoral Science Fundation.

very fast and becomes one of the main reasons of satisfactory solution speed. Since the penalties deduced by Beale and Small and th estronger penalties by Tomlin are all general formulas used for the dual simplex or revised dual simplex method without consideration of bounded variables. As further modifications or extensions, this paper describes calculation of penalties and stronger penalties for the branch-and-bound algorithm of mixed integer linear programming solving with the revised dual simplex method for bounded variables. In manifesting the effectiveness of the algorithm for bounded variable not only. But also the penalties and stronger penalties deduced by the author, the same examples from Petersen[5] are taken and the results are compared with those obtained by Tomolin.

2. Branch-and-Bound Algorithm with Revised Dual Simplex Method for Boundd Variables

The general mixed integer linear programming model with bounded variables can be put in matrix and vector forms as follows:

$$\text{Minimize } Z = CX$$

Subject to

$$\begin{aligned} &AX = b \\ &L \leq X \leq U \\ &X_k \text{ integer } K \in I \end{aligned} \tag{1}$$

Where I is the notation set of integer variables which are placed first and followed by the other continuous variables as vector elements in X.

Deducing the lower boundes as zeros by transforming $X' = X - L$ and using the same notations in (1), the problem becomes as follows:

$$\text{Minimize } Z = CX$$

Subject to

$$\begin{aligned} &AX = b \\ &0 \leq X \leq U \\ &X_k \text{ integer } K \in I \end{aligned} \tag{2}$$

For the problem at some selected node, let A be decomposed into $[B, N_1, N_2]$, where B is basis, and N_1 and N_2 consist of nonbasic columns corresponding to nonbasic variables at their lower bounds $X_{N1} = 0$ and upper bounds $X_{N2} = U$ respectively. Accordingly let R_1 being the notation set of nonbasic variables at their lower bounds, and R_2, the notation set of nobasic variables at their upper bounds. Thus the basic variables X_B and the related objective function value Z can be expressed as follows:

$$X_B = B^{-1}b - B^{-1}N_1X_{N1} - B^{-1}N_2X_{N2}$$
$$Z = C_BB^{-1}b + (C_{N1} - C_BB^{-1}N_1)X_{N1} + (C_{N2} - C_BB^{-1}N_2)X_{N2}$$

To use the branch-and-bound method, now some basic integer variable with a fractional value at the moment is selected to be branched by introducing one of following additive restrictions:

$$X_k \leq [X_k^*] \tag{4}$$

or

$$X_k \geq [X_k^*] + 1 \tag{5}$$

Where $[X_k^*]$ is the nearest integer value of X_k, that is

$$X_k = [X_k^*] + f_k \quad 0 \leq f_k \leq 1 \tag{6}$$

Actually the restriction (4) means an upper bound change for the selected basic variable X_k, but restriction (5), a lower bound change for X_k in the problem at the node. Thus a treatment of sensitivity analysis can be processed by use of the revised dual simplex method for bounded variables in a continuous way based on the basis inverse at hand. It works very fast and becomes one of the main reasons of satisfactory solution speed.

3. Calculation of Penalties for Revised Dual Simplex Method for Bounded Variables

Starting from expressions in (3), the vector of basic variables and the related objective function value can be rewritten as follows:

$$\begin{aligned} X_B &= \bar{b} - \sum_{j\in R_1} Y_jX_j - \sum_{j\in R_2} Y_jX_j \\ Z &= \bar{z} + \sum_{j\in R_1} (C_j - Z_j)X_j + \sum_{j\in R_2} (C_j - Z_j)X_j \end{aligned} \tag{7}$$

Taking into account the first restriction (4) which means a upper bounded change for the basic variable X_k, thus from the right hand side of (7) that the basic variable X_k goes beyond its new upper bound, and this row is then selected as the pivoting row. Some other nonbasic variable X_k may enter in the basis, substituting X_k and making it at its upper bound level. hence it results in a change (i.e. the penalty p_d) of the objective function value after the tableau pivoting. Thus P_d can be calculated by

$$p_d = t \bullet f_k \tag{8}$$

where

$$t = \text{Minimum } \{t_1, t_2\} \tag{9}$$

$$t_1 = \underset{j\in R_1}{\text{Minimum}}\Big\{\frac{c_j - z_j}{y_{kj}} : y_{kj} > 0\Big\} \tag{10}$$

$$t_2 = \underset{j\in R_2}{\text{Minimum}}\Big\{\frac{c_j - z_j}{y_{kj}} : y_{kj} < 0\Big\} \tag{11}$$

To determine penalty P_u corresponding to restriction (5) which means a lower bound change for the basic variable X_k, the basic variable X_k at that time is less than its new lower bound and this row is then selected as the povoting row. Again, using the revised dual simplex calculations, it follows that:

$$P_u = t \bullet (1 - f_k) \tag{12}$$

t is the same as that in (9), but

$$t_1 = \underset{j\in R_1}{\text{Minimum}}\Big\{\frac{c_j - z_j}{-y_{kj}} : y_{kj} < 0\Big\} \tag{13}$$

$$t_2 = \underset{j\in R_2}{\text{Minimum}}\Big\{\frac{c_j - z_j}{-y_{kj}} : y_{kj} > 0\Big\} \tag{14}$$

Once the penalties are computed, and estimated problem bound at the node can be calculated by

$$Z = Z_0 + \text{Minimum } \{P_d, P_u\} \tag{15}$$

Note that some signs in the expressions above are different from those proposed by Beale and Tomlin etc.[1,2,7]. Because the LP problem here is minimized. There should be a minus before Minimum $\{P_d, P_u\}$ in (15) and a negative sign should be added for t_1, t_2 expressions above while considering a maximizing problem.

4. Claculation of Stronger Penalties for Revised Dual Simplex Method for Bounded Variables

The stronger penalties can be deduced from the fact that some nonbasic variables (at their lower bounds or upper bounds) must be integers. Consider the equation associated with an integer X_k in the optimum tableau of the current node. When the restriction (4) is imposed, some nonbasic variable X_q having $y_{kq} > 0$ and locating at its lower bound must be increased (above zero); and another nonbasic variable X_p having $Y_{kp} < 0$ and locating at its upper bound must be decreased (below its upper bound) in order to decrease the right hand side value for this equation to remain satisfied. But if such a nonbasic variable is also integer, then its value must be increased or decreased at least one. This means that the associated penalty must be at least equal $\bar{c}_q$ (or $\bar{c}_p$), and the revised stronger penalty is then given as:

$$p'_d = \min\begin{cases} \min(t_1, t_2) \cdot f_k; & j \notin I \\ \max\{\underset{j\in R_1, y_{kj}>0}{\max}[c_j - z_j, t_1 \cdot f_k], \underset{j\in R_2, y_{kj}<0}{\max}[-c_j + z_j, t_2 \cdot f_k]\}; & j \in I\} \end{cases} \tag{16}$$

Where t_1 and t_2 are these as (10) and (11) respectively.

As for the stronger penalty P_u corresponding to restriction (5), it yields

$$p'_u = \min\begin{cases} \min(t_1, t_2) \cdot (1 - f_k); & j \notin I \\ \max\{\max\limits_{j \in R_1, y_{kj} < 0}[c_j - z_j, t_1 \cdot (1 - f_k)], & \\ \quad \max\limits_{j \in R_2, y_{kj} > 0}[-c_j + z_j, t_2 \cdot (1 - f_k)]\}; & j \in I\} \end{cases}$$

Where t_1 and t_2 are those as (13) and (14) respectively.

Moreover like Tomlin did, and even stronger condition can be also derived from the Gomory cutting-plane constraint in this case. Considering the restriction (4) and equation (6) as well as the equation for basic integer X_k in (7)m, a equation can be formulated, while some nonbasic variables have some changes Δx_j, as follows:

$$X_k - [X_k^*] = f_k - \sum_{j \in R_1} y_{kj} \Delta x_j - \sum_{j \in R_2} y_{kj} \Delta x_j \leq 0 \tag{18}$$

Hence

$$-\sum_{j \in R_1} y_{kj} \Delta x_j - \sum_{j \in R_2} y_{kj} \Delta x_j \leq -f_k \tag{19}$$

and furthermore that

$$-\sum_{j \in R_1, y_{kj} > 0} y_{kj} \Delta x_j - \sum_{j \in R_2, y_{kj} < 0} y_{kj} \Delta x_j \leq -f_k \tag{20}$$

However on the other hand for restriction (5), it follows that

$$-\sum_{j \in R_1, y_{kj} < 0} y_{kj} \Delta x_j - \sum_{j \in R_2, y_{kj} > 0} y_{kj} \Delta x_j \leq f_k - 1 \tag{21}$$

or

$$-\sum_{j \in R_1, y_{kj} < 0} \Big(\frac{f_k y_{kj}}{1 - f_k}\Big) \Delta x_j - \sum_{j \in R_2, y_{kj} > 0} \Big(\frac{f_k y_{kj}}{1 - f_k}\Big) \Delta x_j \leq -f_k \tag{22}$$

Combining both conditions of (20) and (22) and considering that the Δx_j is of the same cost coefficient as x_j has in the objective function, thus the cutting-plane constraint (m-cut)[6] can be constructed as follows:

$$\sum_{j \in R_1, y_{kj} < 0} \Big(\frac{f_k y_{kj}}{1 - f_k}\Big) x_j + \sum_{j \in R_2, y_{kj} > 0} \Big(\frac{f_k y_{kj}}{1 - f_k}\Big) x_j - \sum_{j \in R_1, y_{kj} < 0} (y_{kj}) x_j$$

$$- \sum_{j \in R_2, y_{kj} > 0} (y_{kj}) x_j \leq -f_k \tag{23}$$

and the stronger one[4,7] is

$$s = -f_k + \sum_{j \in N_B} \lambda_{kj} x_j \tag{24}$$

where s is an auxiliary nonnegative variable, and

$$\lambda_{kj} = \begin{cases} y_{kj} & \text{if } j \notin I \text{ and } j \in R_1, y_{kj} > 0 \\ \dfrac{f_k y_{kj}}{(f_k - 1)} & \text{if } j \notin I \text{ and } j \in R_1, y_{kj} < 0 \\ y_{kj} & \text{if } j \notin I \text{ and } j \in R_2, y_{kj} < 0 \\ \dfrac{f_k y_{kj}}{(f_k - 1)} & \text{if } j \notin I \text{ and } j \in R_2, y_{kj} > 0 \\ y_{kj} & \text{if } j \in I \text{ and } f_{kj} \leq f_k \\ \dfrac{f_k(1 - f_k)}{(f_k - 1)} & \text{if } j \in I \text{ and } f_{kj} > f_k \end{cases} \tag{25}$$

where f_{kj} is defined such that

$$f_{kj} = [y^*_{kj}] + f_{kj}, \quad 0 \leq f_{kj} \leq 1 \tag{26}$$

Thus a minimum penalty p^* is

$$p^* = \min_{j \in N_B} \Big\{ \frac{c_j - f_k}{\lambda_{kj}} \Big\} \tag{27}$$

where

$$c_j = \begin{cases} c_j - z_j & \text{if } j \in R_1 \\ -c_j + z_j & \text{if } j \in R_2 \end{cases} \tag{28}$$

The P^* is a stronger criterion which is used to test the node firstly to see whether the node must be discarded or branched.

5. Implementation

A package of mixed integer linear programming using the revised dual simplex algorithm for bounded variables and the stronger penalties duduced for bounded variables was coded by the author, which is used for solving relatively large size problems in the field with success. In order to manifest the effectiveness of the algorithm for bounded variable not only, but also the related penalties, the same examples from Petersen are taken and the results are listed in Table 1 in comparison with those obtained by Tomlin.

Note that the results in the Table are these using the stronger penalties only, and the data given refer to the effort required to reach the actual optimum solution and the total effort required to complete the tree search to verify this optimum. It is known from the Table that, in comparison with those by Tomlin, the total numbers of iterations for most of examples (except problems 3 and 4) when using the package proposed by the author by use of the revised dual simplex method for bounded variables, are lower than those by Tomlin, and are obviously lower for larger problems in particular, eventhough the numbers of branches are not the case. The reason for fewer total iterations is basically due to the algorithm for bounded variables. But the reason for more branches than that by Tomlin is possibly the weaker criterion for selection of basic integer variable for branching in the package made by the authors. The method used in package by the authors is just this according to the greatest cost value in the objective function for simplicity, rather than that by considering the penalties for all candidates of basic integer variable to be selected for branching[6].

Table 1 Test Calculation on Petersen's Problems and result comparisons (optimum solution)

Pro	m	n	Results by Tomlin (Modified algorithm)			Results by Author (Stronger penalties)		
			Branches made	simplex iterations	Time* (seconds)	Branches made	simplex iterations	Time** (seconds)
1	10	6	4	12	1	5	7	3
2	10	10	8	29	2	10	27	4
3	10	15	15	81	5	25	112	13
4	10	20	17	72	6	26	116	14
5	10	28	22	136	9	23	41	9
6	5	39	45	431	19	28	115	12
7	5	50	84	855	42	110	579	52

(Complete search)

Pro	m	n	Results by Tomlin (Modified algorithm)			Results by Author (Stronger penalties)		
			Branches made	simplex iterations	Time* (seconds)	Branches made	simplex iterations	Time** (seconds)
1	10	6	4	12	1	5	15	4
2	10	10	11	49	3	11	43	6
3	10	15	20	127	8	45	160	17
4	10	20	19	87	7	26	126	15
5	10	28	22	169	10	37	135	15
6	5	39	49	525	22	38	298	25
7	5	50	86	926	44	110	605	55
Computer and package used			Univac 1108 under Exec II, early experimental version of UMPIRE			AST 386 SX/16 personal computer package made by the author		

* Include all CP time used in I/O operations (problem input, printout of all integer solutions found, and a trace of the tree search)

** Include all CP time used in I/O operations (reading data from diskfile, printout all integer solutions found on disk, and a trace of the tree search)

It is, on the other hand, obvious that the package runs fast. It takes few or several seconds to solve any of the problems even though the computer used for the testing is just an usual personal computer AST 386 SX/16. As an example of relatively large problems, it can be referred to the mailing list compilation problem in [8]. The problem is of 736 variables (among them 94 are integers) and 316 constraints. It takes just 15 minutes to solve it in a Compaq 4/33 personal computer using the package.

6. Conclusions

As described above that the combination of the revised dual simplex method for bounded variables with the penalties or, in particular, the stronger penalties derived in the paper is very effective in solving the pure integer, mixed integer or zero-one integer programming problems, especially the relatively large-size problems. Futher improvements in both the algorithm and program technique are necessary in order to obtain more satisfactory results.

Acknowledgement. The authors would like to thank two anonymous referees for their helpful suggestions and corrections on an earlier draft of the paper according to which we have improved the content and composition.

References

[1] E.M.L. Beale , Mathematical Programming in Practice, Pitmans, London, 1968.

[2] E.M.L. Beale, R.E. Small, Mixed integer programming by a branch and bound technique, *Proce. of the 3rd IFIP Congress 1965,* **2** (1965), 450–451.

[3] J.J.H. Forrest, J.P.H. Hirst, J.A. Tomlin, Practical solution of large mixed integer programming problems with UMPIRE, *Management Science,* **20**:5 (1974), 736–773.

[4] R.E. Gomory, An algorithm for mixed integer problems, RM-2597, The Land Corporation, Santa Monica, California, 1960.

[5] C.C. Petersen, Computational experience with varieties of the Balas algorithm applied to the selection of R&D projects, *Management Science,* **13** (1967), 736–750.

[6] H.A. Taha, Integer Programming, Theory, Applications and Computations, Wiley & Sons, New York, 1975.

[7] J.A. Tomlin, An improve branch-and-bound method for integer programming, *Operations Research,* **19** (1971), 1070–1075.

[8] Qing-huai Hu and Yi-ming Wei, Solving relatively larger size mixed-integer linear programming problems using personal computers (in Chinese), *J. on Numerical Methods and Computer Applications,* **4** (1996), 254–262.