Cardinality Cuts: New Cutting Planes for 0-1 Programming

Osman Oguz
Department of Industrial Engineering, Bilkent University, Ankara, Turkey
E-mail: ooguz@bilkent.edu.tr

Abstract: We present new valid inequalities that work in similar ways to well known cover inequalities.The differences exist in three aspects. First difference is in the generation of the inequalities. The method used in generation of the new cuts is more practical in contrast to classical cover inequalities. Second difference is the more general applicability, i.e., being useful for problems like TSP. The third aspect is superior efficiency as indicated by our preliminary experiments.

Keywords: *0-1 Integer programming, Valid inequalities, Branch-and-cut*

1.Introduction

Crowder et al. (1983) demonstrated the important role of cover inequalities used as cutting planes in the branch-and-cut methods for solving integer programming problems. Relatively recent works by Gu et al. (1999) and Johnson et al. (2000) provide extensive dicussions of available strategic choices for using cover inequalities in the branch-and-cut process for
0-1 programming. Nemhauser and Wolsey (1988), and Wolsey (1998) may be seen for basic expositions of the subject and related issues.

We work on the 0-1 integer programming problem given below:

Maximize $z=\sum_{j=1}^{n} c_j x_j$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \text{ for } i=1,..,m \qquad \text{(IP)}$$

$x_j \in \{0,1\}$ for all $j=1,..,n$

There are no restrictions on the signs of the parameters c_j , a_{ij} and b_i .

The next section consists of the description and the generation method of the inequalities together with the proof of validity. Some numerical examples and the results of the preliminary numerical experiments are discussed in Section 3. Conclusions and comments follow in Section 4.

2. The New Cut

Consider the problem (IP) and let $X_{LP}=(x_1,..,x_n)$ denote a solution to the linear programming (LP) relaxation of this problem. Also let $S_0=\{j|\ x_j>0\}$and solve the following problem:

$$z_0= \max \sum_{j=1}^{n} x_j \ : \ \sum_{j=1}^{n} a_{ij} x_j \le b_i \text{ for } i=1,..,m, \text{ and } x_j \in \{0,1\} \text{ for all } j=1,..,n.$$

The following inequality obviously holds:

$$\sum_{j \in S_0} x_j \le z_0$$

Also, this inequality holds for all possible values x_j such that $j \in S_0$ in any solution of the LP relaxation for any objective function. Moreover, the inequality holds in the form:

$$\sum_{j \in S_0} x_j \le \lfloor z_0 \rfloor$$

for any integer solution of the problem for any objective function. In fact, this last inequality may be an effective cut to make some noninteger solutions infeasible. However, its use can be limited to very few instances and it becomes ineffective very soon in a cutting plane framework. It may even be useless if is integer z_0 valued. None the less, it is the starting point of our proposal for a new type of cuts.

Starting with the solution to the LP relaxation of the problem (IP), we partition S_0 into two subsets:
$S_1 = \{j | j \in S_0 \cap x_j = 1\}$ and $S_2 = S_0 \setminus S_1$ .Then, naming the LP relaxation of the orginal integer program (IP) as (P1), define the following linear program (P2):

$$z_1 = \max \sum_{j \in S_2} x_j$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \text{ for } i=1,..,m \qquad \text{(P2)}$$

$$\sum_{j \in S_1} x_j = | S_1 |$$

and

$$0 \le x_j \le 1 \text{ for } j=1,..,n$$

The the following is true:

Proposition 1: The inequality given below is valid for the solution set of the problem (IP) for sufficienly large values of the parameter r;

$$\sum_{j \in S_1} r x_j + \sum_{j \in S_2} x_j \le r | S_1 | + \lfloor z_1 \rfloor$$

is a valid inequality for the solution set of the problem (IP) for sufficienly large values of the parameter r.

Proof: The first summation term in the left hand side of the inequality can attain its maximum value when all x_j for $j \in S_1$ are equal to 1. At this instance, the number of x_j for $j \in S_2$ being equal to 1 cannot be more than $\lfloor z_1 \rfloor$ by the definition of z_1 in (P2). Thus we conclude that the number of x_j for $j \in S_2$ being equal to 1 can be greater than $\lfloor z_1 \rfloor$, only if at least one x_j variable in S_1 is equal to 0.
Assigning a very large value to r will allow the inequality to hold even when all variables in S_2 are equal to 1, thus eliminating no feasible solutions to (IP).■

Consider the cases where the values of x_j in S_2 for some X_{LP} happen so that $\sum_{j \in S_2} x_j > \lfloor z_1 \rfloor$.

The valid inequality defined above will make the relaxed solution infeasible. That is, the valid inequality represents an effective cutting plane for the problem (IP).

Assigning an arbitrarily large value to the parameter r is not desirable since the quality of the cut will be low, i.e., the cut will separate a very small part of the underlying polyhedron containing no integer solutions. A more reasonable approach is to choose r more carefully.

Consider a slight variation of (P2) given below:

$$z_2 = \max \sum_{j \in S_2} x_j$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ for } i=1,..,m$$

$$\sum_{j \in S_1} x_j = | S_1 | - 1$$

and

$$0 \leq x_j \leq 1 \text{ for } j=1,..,n$$

It is expected that $z_2 > z_1$, otherwise value of r will be immaterial, and $z_2 - \lfloor z_1 \rfloor$ is an upperbound on how many more x_j in S_2 can take a value of 1 when we decrease the cardinality of S_1 by 1. Note that when we replace the right hand side of the equation $\sum_{j \in S_1} x_j = | S_1 | - 1$ by $| S_1 | - 2$ the difference $z_2 - \lfloor z_1 \rfloor$ will less than double since z_2 parametrized on $| S_1 | - k$, for $k \geq 0$ is a concave piecewise linear function. Thus, setting $r = z_2 - \lfloor z_1 \rfloor$ gives a relaxation sufficiently tight for the purpose of cutting deeper into the underlying polyhedron. We set r=1 when $z_2 - \lfloor z_1 \rfloor \leq 1$.

There will, of course, be instances where the valid inequality will be useless, i.e., will fail to eliminate a fractional solution. We can try a few more things before giving up and starting branching. The most obvious thing to do is to play around the partition of S_0 to S_1 and S_2 . We have tried two strategies with partial success. First one is to move few variables from S_2 to S_1 picking those with values close to 1. Second strategy is to eliminate some variables in S_2 from consideration, i.e., not including them in the valid inequalities, or in the objective function of problem (P2). We may end up with effective cuts as aresult of these changes.

3.Two numerical examples and the results from preliminary computational experiments

To illustrate the use of the cut explained in the previous section, we implement it on two small scale problems. The first one is a 4 variable 0-1 integer programming problem given below:

Maximize z= 5x1+11x2+4x3+10x4
Subject to:

7x1+3x2-4x3-2x4 ≤1
-2x1+7x2+3x3+x4 ≤ 6
2x2+3x3+6x4 ≤ 5
-3x1+2x3 ≤ 1

x1,x2,x3,x4 are binary.

The LP relaxation of this problem gives the following solution:
x1=.78, x2=.5, x3=x4=1 with z=23.51

The value of $\lfloor z_1 \rfloor$ is 1, and r=1. The resulting cut is:
x1+x2+x3+x4 ≤3
Adding this cut to the original problem, we resolve the LP relaxation again and get:
x1=.64, x2=.55, x3=.79, x4=1 and z=22.54
Setting S_1={4} and S_2={1,2,3} doesn't work because it requires setting x4=0, which is infeasible and r is undefined.
Trying S_1 ={3,4} and S_2 ={1,2} is useless since it gives the previous inequality. Choosing S_1 ={3,4} and S_2={1} gives the inequality x1+x3+x4 ≤ 3 which is satisfied by the current solution. The last choice remaining is S_1={3,4}and S_2 ={2} which gives the cut x2+x3+x4≤ 2. Adding this to the problem as the 6th constraint, we solve the LP relaxation one more time to get:
x1=.33, x2=.66, x3=.33, x4=1 and z=20.33
Trying S_1={2,4}results in an infeasible (P2), which implies x2+x4≤ 1 is a valid inequality violated by the current solution.We add this constraint to the problem and solve the LP relaxation to obtain the integer solution of the problem:
x1=x3=x4=1, x2=0, and z=19.

The second example is a 15 city symmetric traveling salesman problem. We worked on a two matching formulation of the problem. The following
noninteger solution is obtained when the LP relaxation of the problem is solved:

x1,2= 1
x1,5=.5
x1,6=.5
x2,12=1
x3,10=.5
x3,11=1
x3,15=.5
x4,8=1
x4,13=1
x5,6=.5
x5,8=1
x6,14=1
x7,9=1
x7,15=1
x9,10=1
x10,15=.5
x11,14=1
x12,13=1
The corresponding value of the objective function is 4956.5. The cardinality cut generated for this solution is:

1.8(x1,2+x2,12+x3,11+x4,8+x4,13+x5,8+x6,14+x7,9+x7,15+x9,10+x11,14+x12,13+x1,5)+x1,6+x3,10+x3,15+ x5,6+x10,15≤ 24.4

When the LP relaxation solved with this constraint, the objective function value raises to 4956.71. If we add a subtour breaking constraint instead, the objective function value is 4956.75.

Actually, we were able to solve this 15 city problem to optimality using only cardinality cuts and subtour breaking constraints. Total number of such constraints was 19.

In another experiment, we tested the cardinality constraints on a (100 variables) X (5 constraints) multidimensional knapsack problem.
(The problem is *mknap1.lp* taken from the OR Library maintained by Beasley(1998)). We first run an exact cover inequality generating program to find all cover cuts at the root. That is, we first solve the LP relaxation and find the best cut for this solution, and then add the cut to the problem, and resolve the relaxation to find a new cut. We continue until no cover cuts can be found. We repeated the same experiment a second time using cardinality cuts.

Here are the results:

Number of cover inequalities generated (without lifting): 2
Number of cover inequalities generated (with lifting): 7
Upper bound improvement:from 24585.9027 to 24583.64.70

Number of cardinality cuts generated : 9
Upper bound improvement: 24585.9027 to 24577.2824

4. Conclusions and Comments

The proposed cuts in this study seem to have some potential. There is need for more time and experimentations to prove their viability. The results presented are very skimpy in this respect. Also, more work is needed to better formalize the suggested methods.

References:

Crowder, H., E. L. Johnson, M. W. Padberg. 1983. Solving large scale 0-1 linear programming problems. *Oper. Res*. 31, 803-834.

Gu, Z., G. L. Nemhauser, M. W. P. Savelsbergh. 1999. Lifted Cover Inequalities for 0-1 Integer Programs: Complexity, *INFORMS Journal on Computing,* 11, 117-123

Johnson, E. L., G. L. Nemhauser, M. W. P. Savelsbergh. 2000. Progress in Linear Programming-Based Algorithms for Integer Programming: An Exposition. *INFORMS Journal on Computing* 12, 2-23.

Nemhauser, G. L., L. A. Wolsey. 1988. Integer and Combinatorial Optimization. Wiley, New York.

Wolsey, L. A. 1998. Integer Programming. Wiley, New York.