

**Alternative Assignment Models for
Time Constrained Local Fleet Assignment in which
Service and Travel Times are Stochastic**

UCI-ITS-LI-WP-00-4

Xiubin Wang ¹
Amelia C. Regan ²

¹ Institute of Transportation Studies, University of California, Irvine

² Department of Civil and Environmental Engineering, Graduate School of Management
and Institute of Transportation Studies, University of California, Irvine, aregan@uci.edu

November 2000

Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600, U.S.A.
<http://www.its.uci.edu>

A Cutting Plane Method for Integer Programming Problems with Binary Variables

Xiubin Wang

Amelia C. Regan

Institute of Transportation Studies

University of California, Irvine

Irvine, CA 92697

Abstract

In this report, a new cutting plane method is proposed. This method shows advantages over Gomory's method. The number of cuts added is bounded by the number of variables. It can obtain deeper cuts by adopting branch and bound method as well. Simple analysis is presented. It is a start of a series of research in solution algorithms for integer programming problems.

1. Introduction

Consider an integer programming problem (IP) in which the integer variables are restricted to binary values. The problem can be stated as follows.

$$\begin{array}{ll} \text{Min} & CX \\ \text{s.t.} & AX = B \\ & X \text{ binary} \end{array}$$

This is the typical IP formulation. Two types of algorithms have been developed to solve these problems. They are cutting plane methods, of which the most well known and often used is Gomory's method, and the branch and bound method. In this section of the dissertation we present a new cutting plane method. Simple analysis of it is performed. Its advantage can be stated as follows.

1. The number of constraints added is no more than the number of binary variables;
2. Can be combined with branching method to obtain a deeper cut.

2. The Cutting Plane Method

The new cutting plane method follows the steps listed below

- i) Select a variable x_i that is fractional in the solution to the LP relaxation of the original IP problem. Solve two formulations with this variable being set to be 1 and 0 respectively. Suppose the two formulations are represented by $F_i(1)$ and $F_i(0)$ with minimums M_i^1 and M_i^0 correspondingly. Then add a constraint,

$$CX \geq M_i^0 + (M_i^1 - M_i^0)x_i$$

This provides us with a new formulation with this additional constraint. For convenience, we call this constraint a Twin-Min cut. If either $F_i(1)$ or $F_i(0)$ has no solution, then fix x_i to be 0 or 1 at all the subsequent iterations. This implies that no Twin-Min cut from x_i will be added later.

- ii) Select another variable that is fractional in the solution of either $F_i(1)$ or $F_i(0)$. Repeat the same process to add constraints. If there are no fractional variables in the LP solution, stop.

It should be clear that in the case of maximization, the added constraints are in the following form: $CX \leq M_i^0 + (M_i^1 - M_i^0)x_i$. Without explicit explanation, we refer to the case of minimization in the following sections.

Observation

A new Twin-Min cut makes old ones of the same variable in the formulation redundant.

An argument follows. With iterations, both M_i^1 and M_i^0 associated with the variable x_i are monotonically non-decreasing. As a result, for each value of the variable x_i , its associated objective value is also monotonically non-decreasing after its Twin-Min cut is added. Therefore, the new cut replaces the old one. (End of argument.)

As a result, the maximum number of added constraints is equal to the number of variables. On the other hand, in Gomory's cutting plane method, there is no such a tight limit on the number of constraints added. In that method, the number of added constraints explodes with iterations. As the algorithm progresses, many constraints become redundant. However, there is no efficient way to identify redundant constraints. The identification of the redundant constraints is itself an NP-hard problem (as indicated in Nemhauser and Wolsey, 1988). Therefore an advantage of our new method is that the number of added constraints does not explode with iterations. In addition, it could be combined with a typical branch and bound method to obtain deeper cuts. In the next section we examine some of the analytical properties of the method.

3. Analytical Properties

Proposition I

For an IP formulation with two binary variables, at most two Twin-Min cuts are added.

Proof

Let us start with the case of maximization for graphical clarity. Suppose the two variables are x_1 and x_2 . We first consider adding the Twin-Min cut of x_1 . For the formulations with x_1 being binary, there are two possibilities when the LP relaxation of the original formulation does not have an integer solution. In one case there is an optimal integer solution and a fractional solution to $F_1(1)$ and $F_1(0)$ respectively. In the other x_2 is fractional in both $F_1(1)$ and $F_1(0)$. In the first case, we will obtain a Twin-Min cut passing through an integer point and a point with x_2 being fractional while in the second case, we will end with a Twin-Min cut passing through two points with x_2 being fractional. In the second case, the variable x_2 would be set to a fixed integer (1 or 0) finally in the formulation when additional Twin-Min cut of x_2 is further considered. Then one more branch over x_1 will solve the problem ultimately. In the first case, the two cuts can be shown in the diagram as Figure 5.3. (End of proof.)

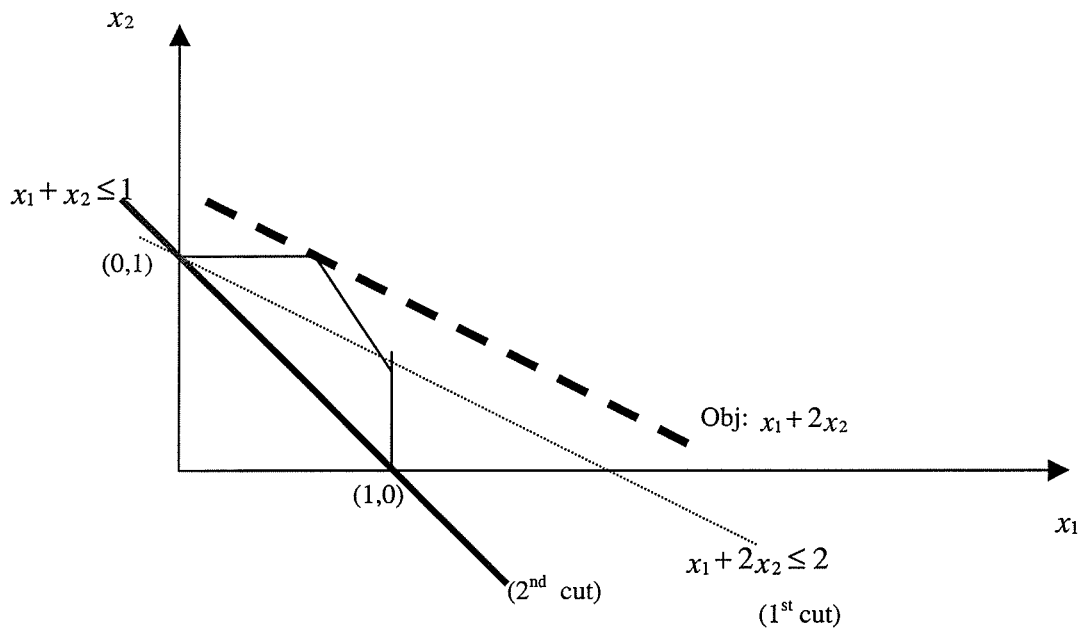


Figure 5.3) Example of cuts added

In the example of Figure 5.3, the first cut corresponds to the variable x_1 , which was set to be 1 and 0 respectively while the second cut corresponds to x_2 in the formulation with the first cut added.

Gomory's cutting plane method does not possess so small a limit on the number of constraints added even in this simple case.

Proposition II

The new cutting plane method indicated in step i) and ii) guarantees finite convergence.

Argument

A formal proof of this proposition would be lengthy. Here a brief argument based on intuition is given instead. First, the solution space is bounded at 1 and 0 at every dimension. So the optimal integer solution for a feasible problem is also bounded by certain values. The optimal fractional solution must be a limited value in the same reason. As a result, the difference between the optimal integer solution and the optimal fractional solution is limited and could be bounded by a limited value. We only need to show that the optimal fractional solution at each iteration monotonically increases by a value, which is no less than zero. This can be shown to be true. At each iteration, if the value of a variable x_i is fractional in the optimal solution, then to increase or decrease the value of this variable leads to either a bigger or equal objective value. Therefore finite convergence is guaranteed. (End of argument.)

Though we have shown that this algorithm has finite convergence, methods to raise its convergence speed are needed. A possible way to improve it is to obtain a deeper Twin-Min cut with an improved M_i^1 and M_i^0 . The following section explains this idea.

4. Obtaining Deeper Cuts

As an example, for the formulation $F_i(1)$, if we further split it into two new formulations with one *more* fractional variable, assume it is x_j , being 1 and 0 respectively we could end with two improved (or at least not worse) objective values for M_i^1 . Select the smaller

one as M_i^1 . This M_i^1 must be a better bound than the one obtained without fixing x_j to be integer. In the same way, we can obtain a better bound M_i^0 for $F_i(0)$. With both M_i^1 and M_i^0 being improved, we have a deeper cut corresponding to the variable x_i . Better bounds are available by further branching over more variables. In the extreme, if branching over all the variables could be performed, only one cut is needed to generate an optimal integer solution. An advantage of this cutting plane method in getting a better cut is that its branching depth (the number of variables over which branching is made) is controllable.

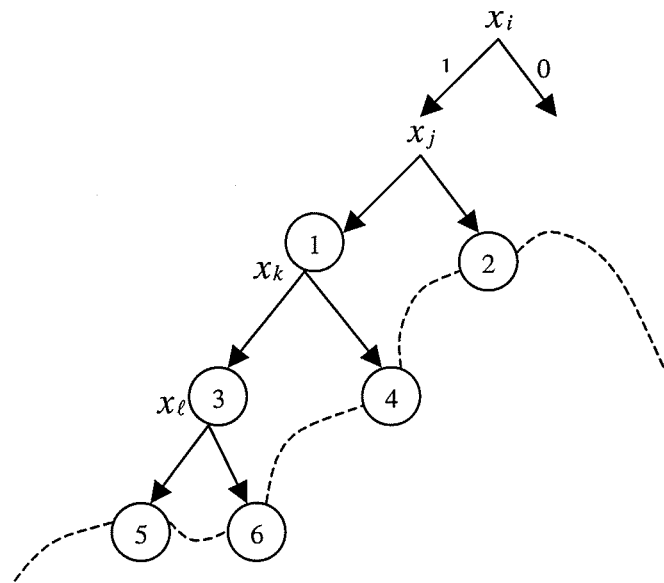


Figure 5.4) A branching procedure to get deeper cut

The method to get an improved M_i^1 or M_i^0 for x_i is shown in Figure 5.4. The left side, which contains the nodes from 1 to 4, is for the value M_i^1 . Here let φ_i represent the objective value at node i . M_i^1 is evaluated in the following way $M_i^1 = \min\{\varphi_i\}$, where i is

the index of the node on a cut across the tree at the side with x_i being equal to 1. A cut is shown in Figure 5.4. For clarification we provide the following definition.

Definition

A set of nodes in the branching tree is defined as a *cut* if the complimentary set of the nodes is split into two sets without linkage and if there are no paths between the nodes of the cut.

If the objective value of a node on a cut is not the minimum of all the nodes on this cut, it is not necessary to branch further from this node. In short, it is only meaningful to branch over the node on a cut with the minimum value in order to get a deeper cut. In this way, a bigger M_i^1 is available, which is critical for a deeper cut. Therefore the process of looking for deeper cut associated with a variable involves finding a cut of largest minimum value for the tree under a certain variable.

5. Conclusion

In this section, a new cutting plane method is provided. Some simple properties are analyzed. Preliminary analysis shows it possesses some advantage over Gomory's method. One of them is that it guarantees the number of constraints added in the formulation is bounded by the number of binary variables in the problem. We show that for the simple case in which only two binary variables are present, our method guarantees

that the number of iterations required is less than Gomory's method guarantees. Though this algorithm is discussed in the context of IP problems, it applies to the mixed IP problems (MIP) too. In addition, it is possible to extend this result to the general IP/MIP problems that do not require the integer variables to be binary. An immediate task, however, is to test the performance of the proposed algorithm on typical IP formulations