

A Pre-estimate Approach to Find a Good Initial Point to Integer Linear Programming Problems

Zhifeng Kong* and Jiaheng Wei

Institute of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, Shaanxi, China

*Corresponding author

Abstract—Integer Linear Programming problems are significant in many areas, and there exist many classic methods to deal with such problems. Based on the intuition that a proper pre-estimation of those problems may help find a good initial point that is relatively closed to the final solution, we constructed a framework to do such pre-estimate by introducing δ . We proposed two methods to estimate δ by trials and by direct estimation. In a case study our method reduces about 7% of computational expense compared to the Branch and Bound Algorithm.

Keywords-integer linear programming; relaxing methods; pre-estimate

I. INTRODUCTION

Integer Linear Programming (ILP) problems are of great importance in many areas. One classical method to solve the ILP problems is the Branch and Bound Algorithm (Branch and Cut Algorithm), and there are also many other approaches such as the cutting plane algorithm and heuristic methods. Many researchers have done a great number of works in this area. Linderoth JT, et al made a survey on existed strategies of branch and bound methods and presented novel search strategies [1]. Still C, et al developed a sequential cutting plane method to solve convex mixed integer nonlinear programming problems [2]. Cornuejols G, et al did prediction on the size of branch and bound trees [3]. Morrison David R, et al made a survey of recent advances in searching, branching and pruning [4]. Akrotirianakis I, et al developed an outer approximation based brand and bound algorithm for convex zero-one Mixed Integer Nonlinear Programming problems [5]. Wang HF, et al developed a hybrid approach to solve differentiable integer programming problems, using a genetic algorithm after approximation of the objective function and constraints [6].

In this paper, we implemented a novel approach to Integer Linear Programming problems, based on the intuition that an appropriate pre-estimation may provide an initial point probably closed to the solution. We translated the idea into mathematical form by introducing δ and then proposed two methods to estimate δ . A case study shows that our method reduces 7% of computational expense compared to the original Branch and Bound Algorithm.

II. FRAMEWORK AND IDEA

Contrast to the classic Branch and Bound Method, our method provides another view to deal with the Integer Linear Programming (ILP) problems. For an ILP problem whose objective function is g , suppose the solution of its relaxed problem is $z^* = g(x^*)$, where z^* is the minimum value. Then

there exists $\delta > 0$ such that the solution (minimum value) of the original ILP problem $z_I < z^* + \delta$, or $\exists x_I \in N^n$ such that $g(x_I) < z^* + \delta$. The idea, as well as the idea of the Branch and Bound Algorithm, is shown in Figure I.

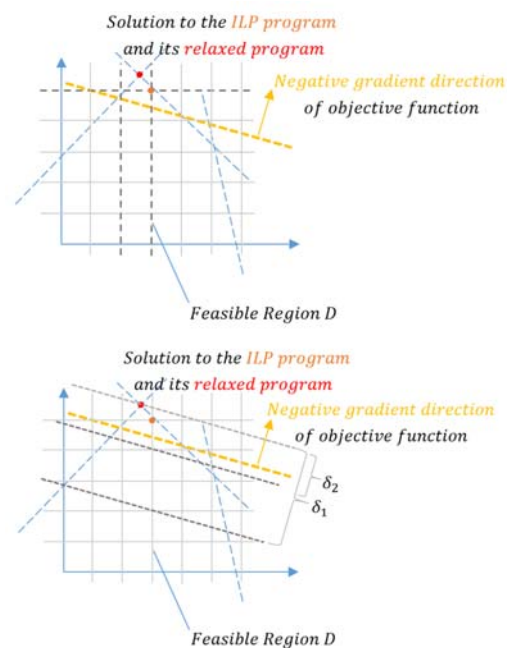


FIGURE I. IDEAS OF THE BRANCH AND BOUND ALGORITHM (TOP) AND PRE-ESTIMATE APPROACH (BOTTOM)

Suppose the ILP problem is

$$\min z = g(x) \text{ s. t. } Ax = 0, x \geq 0, x \in Z^n$$

Then solve the relaxed LP problem

$$z^* = \operatorname{argmin} g(x) \text{ s. t. } Ax = 0, x \geq 0$$

which is easy. Add a constraint to the ILP problem and construct ILP' problem

$$\min z = g(x) \text{ s. t. } Ax = 0, g(x) \leq z^* + \delta, x \geq 0, x \in Z^n$$

If δ is small, any x that satisfies the constraints of ILP' would be a good initial point to apply other classical methods, such as the Branch and Bound Method. Framework of our method is shown in Figure II.

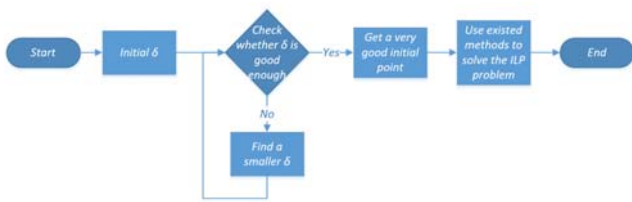


FIGURE II. FRAMEWORK

In some situations, δ can be estimated theoretically. However, in most situations, δ is hard to estimate. In the next section two methods are introduced to estimate δ .

Since it is hard to estimate integral points that satisfy the constraints of ILP', it is not time efficient to look for a δ that is excessively small.

III. TWO METHODS TO ESTIMATE δ

Usually it is very hard to provide a proper δ , since in some occasions the solution to the ILP problem and solution to its relaxed LP problem may be greatly different. But when the objective function and constraints are not complex, two methods are provided to estimate a proper δ .

A. Estimation by Trials

A vital point is that δ should not be small, which at least satisfies that there exists a solution to the constraints of ILP'. To ensure this, let x_0 be a solution to the constraints of the ILP problem, and set $\delta = g(x_0) - x^*$, then δ will not make ILP' unsolvable. However, the δ above may be quite large, so it is more reasonable to select several solutions x_1, x_2, \dots, x_m to the constraints of ILP problem and find the minimum value of $g(x_i), i = 1, 2, \dots, m$. Then let

$$\delta = \min_{i \in \{1, \dots, m\}} g(x_i) - x^*$$

This δ may be much smaller. For the efficiency of the algorithm, m should also be proper. For simplification, suppose there are N integral points that satisfy the constraints of the ILP problem. In order to reduce the computational cost, we want

$$m + \beta N(g(x) \leq \min_{i \in \{1, \dots, m\}} g(x_i))$$

to be relatively small, where β is a parameter which refers to the computational cost of the remaining integral points, and $N(\cdot)$ refers to the number of integral points that satisfy the constraints of ILP as well as the inequality above. Since a decrease will make β much larger, set $\beta = e^{-m}$ for simplification.

To estimate m , suppose the distribution of g is totally unknown. So for generalization, assume

$$\frac{N(g(x) \leq k)}{N(D)} = \frac{k - \min_{x \in D, x \in \mathbb{Z}^n} g(x)}{\max_{x \in D, x \in \mathbb{Z}^n} g(x) - \min_{x \in D, x \in \mathbb{Z}^n} g(x)}$$

where D refers to all points that satisfy the constraints of the relaxed LP problem, and $N(D)$ the integral points in D . In order to calculate the expectation of

$$N(g(x) \leq \min_{i \in \{1, \dots, m\}} g(x_i)) / N(D)$$

Suppose

$$\min_{x \in D, x \in \mathbb{Z}^n} g(x) = 0, \max_{x \in D, x \in \mathbb{Z}^n} g(x) = 1$$

Let the interval $[0,1]$ be cut into t slices, and let value $N(g(x) \leq \min_{i \in \{1, \dots, m\}} g(x_i))$ can only be $\frac{0}{t}, \frac{1}{t}, \dots, \frac{t}{t}$. Then the probability the value equals to $\frac{i}{t}$ is

$$\frac{(t+1-i)^m - (t-i)^m}{(t+1)^m}$$

So we have

$$E\left(\frac{N(g(x) \leq \min_{i \in \{1, \dots, m\}} g(x_i))}{N(D)}\right) = \lim_{t \rightarrow \infty} \sum_{i=0}^t \frac{i}{t} \cdot \frac{(t+1-i)^m - (t-i)^m}{(t+1)^m} = \frac{1}{m+1}$$

Then the problem is to minimize $m + e^{-m} \cdot \frac{N(D)}{m+1}$. This could be easily solved by letting its derivative equal to zero. At last, the proper m is approximately equal to $\ln N(D) - \ln m$, or $m \approx \ln N(D) - \ln \ln N(D)$. When $N(D) = N$ is large, it is not necessary to calculate it precisely; a rough estimation would be enough.

B. Direct Estimation

Suppose the objective function of the ILP problem is $g(x) = c^T x$. For most ordinary ILP problems, the solution to the ILP problem and that to the relaxed LP problem may be much closed. For example, when $c \geq 0 (c_i \geq 0)$ and $A \geq 0 (A_{ij} \geq 0)$, a theoretical estimation is $\delta = \sum_{i=1}^n c_i$, which ensures that the ILP' has at least a solution. In fact, more general estimation can be written as $\delta = k \|c\|_1$. A small k will lead to a better initial point, while taking a higher risk of losing all solutions; a large k always ensures that solution of ILP' exists, but the estimation may be not that good. The scale of k depends on the shape of D , which is derived from A . k should be relatively large when D has very sharp parts, which means two rows of A must be similar. Thus, by calculating the correlation matrix of A , k can be judged by whether there is strong linear relationship between two different rows of A .

For more general cases, δ should be approximately equal to the mean value of $g(x)$ where x falls into a ball with radius 1 and centroid x^* . That is,

$$\delta \approx E_{\|x-x^*\|_2=1} (|g(x^* + \alpha) - g(x^*)|)$$

Since the precise expectation is hard to know, an approximation would reduce the computational expense without losing much information. One approach could be selecting α to be $\pm e_i, i = 1, 2, \dots, n$, where

$$e_i = \left(\underbrace{0, \dots, 0, 1}_{i \text{ elements}}, 0, \dots, 0 \right)^T$$

IV. CASE STUDY

Consider the Integer Programming problem

$$\begin{aligned} \min z = & x_1^2 + x_2^2 + 3x_3^2 + 4x_4^2 + 2x_5^2 \\ & + 8x_1 + 2x_2 + 3x_3 - x_4 - 2x_5 \end{aligned}$$

Such that

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 6 \\ 2 & 1 & 3 & 0 & 0 \\ 0 & 0 & 1 & 1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \leq \begin{pmatrix} 400 \\ 800 \\ 200 \\ 200 \end{pmatrix}$$

$$1 \leq x_i \leq 99, x_i \text{ is integer}, i = 1, \dots, 5$$

Using Lingo software to solve this problem directly, 1042 iterations are needed. And its relaxed programming problem has solution $z^* = -0.5625$ and

$$x^* = \left(0, 0, 0, \frac{1}{8}, \frac{1}{2} \right)^T$$

with 56 iterations.

Now estimate δ with method introduced in section III A: we have $\delta = 11.5$ and suppose it is good enough. Then only 912 iterations are needed to solve the problem. Let δ equal to some other values between 1 and 20, no more than 950 iterations in all occasions are needed to solve the problem. Count the former 56 iterations on, our method reduces more than 7% of computational expense. Results are similar if δ is slightly changed. Thus, our method indeed provides a better initial point than the original Branch and Bound Algorithm.

V. CONCLUSION

In this paper, we considered a novel approach to deal with integer programming, based on the intuition that a good pre-estimate method could help find an initial point that may be very closed to the solution. Then we proposed two methods to do the estimation, one by trials, and another by direct estimation. With such initial point, time expense of existed methods to solve the programming problem could be reduced. In the case we studied, 7% of computational expense was decreased. In the future, we plan to find out which kind of integer programs are more likely to be improved by our method. We also plan to apply our method to other problems, such as mixed integer programming problems.

REFERENCES

- [1] Linderoth JT, Savelsbergh MWP, "Computational study of search strategies for mixed integer programming", *Inform Journal On Computing*, vol.11, pages173-187, 1999.
- [2] Still C, Westerlund T, "Solving convex MINLP optimization problems using a sequential cutting plane algorithm", *Computational Optimization and Applications*, vol.34, pages63-83, 2006.
- [3] Cornuejols G, Karamanov M, Li YJ, "Early estimates of the size of branch-and-bound trees", *Inform Journal on Computing*, vol.18, pages86-96, 2006.
- [4] Morrison David R, Jacobson Sheldon H, Sauppe Jason J, Sewell Edward C, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning", *Discrete Optimization*, vol.19, pages79-102, 2016.
- [5] Akrotirianakis I, Maros I, Rustem B, "An outer approximation based branch and cut algorithm for convex 0-1 MINLP problems", *Optimization Methods & Software*, vol.16, pages21-47, 2001.
- [6] Wang HF, Liao YC, "A hybrid approach to resolving a differentiable integer program", *Computers & Operations Research*, vol.25, pages505-517, 1998.