Cutting plane algorithm:
an interactive web application

Riccardo Locci

July 2020

Abstract

In mathematical optimization, the cutting-plane method is any of a variety of optimization methods that iteratively refine a feasible set or objective function by means of linear inequalities, termed cuts. Such procedures are commonly used to find integer solutions to mixed integer linear programming (MILP) problems, as well as to solve general, not necessarily differentiable convex optimization problems. The use of cutting planes to solve MILP was introduced by Ralph E. Gomory.[1]

Focus of this work is to propose the implementation of the cutting plane algorithm as an interactive web application that shows the algorithm's execution flow step-by-step. This work will result in a auxiliary learning tool to better understand and visualize how the algorithm works thanks to the grafical user interface exposed by the application. Since the developed application focuses on a geometric representation of the algorithm, it is thought to work with two decisional variables. The application was develop and tested on Google Chrome on the following machines:

- Desktop:
  - OS: Windows 10 Pro 1903,
  - CPU: Intel®Core i5-4440 3.10 GHz
  - RAM: 16 GB DDR3
- MacBook Air:
  - OS: macOS Catalina 10.15.5,
  - CPU: Intel®Core i5 dual-core 1.6 GHz
  - RAM: 8 GB LPDDR3

1 Integer Linear Programming formal definition

Linear programming (LP, also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. Linear programming is a special case of mathematical programming (also known as mathematical optimization). [2]

More formally, linear programming is a technique for the optimization of a linear objective function, subject to linear equality and linear inequality constraints. Its feasible region is a convex polytope, which is a set defined as the intersection of finitely many half spaces, each of which is defined by a linear inequality. Its objective function is a real-valued affine (linear) function defined on this polyhedron. A linear programming algorithm finds a point in the polytope where this function has the smallest (or largest) value if such a point exists. [2]

Linear programs are problems that can be expressed in canonical form as:

$$\begin{array}{ll} \text{maximize} & c'x \\ \text{subject to} & Ax \leq b \\ \text{and} & x \geq 0 \end{array}$$

where x represents the vector of variables (to be determined), c and b are vectors of (known) coefficients and A is a (known) matrix of coefficients. The expression to be maximized or minimized is called the objective function ($c'x$ in this case). The inequalities $Ax \leq b$ and $x \geq 0$ are the constraints which specify a convex polytope over which the objective function is to be optimized. In this context, two vectors are comparable when they have the same dimensions. If every entry in the first is less-than or equal-to the corresponding entry in the second, then it can be said that the first vector is less-than or equal-to the second vector. [2]

Linear programming can be applied to various fields of study. It is widely used in mathematics, and to a lesser extent in business, economics, and for some engineering problems. Industries that use linear programming models include transportation, energy, telecommunications, and manufacturing. It has proven useful in modeling diverse types of problems in planning, routing, scheduling, assignment, and design. [2]

An integer programming problem is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers. In many settings the term refers to integer linear programming (ILP), in which the objective function and the constraints (other than the integer constraints) are linear.

An integer linear program in canonical form is expressed as:

$$\begin{array}{ll} \text{maximize} & c'x \\ \text{subject to} & Ax \leq b \\ \text{and} & x \geq 0 \\ \text{and} & x \in \mathbb{Z}^n \end{array}$$

2 Overview about Cutting Plane Method

3 Implementation of the algorithm

3.1 Data structures

3.2 Custom implementation

3.2.1 Preprocessing

3.2.2 Processing

3.2.3 Postprocessing

3.3 Web application GUI

3.3.1 Input choices

3.3.2 Canvas

3.3.3 Interaction section

3.3.4 Info section

4 Results and performances

5 Conclusions

References

[1] Wikipedia, *Cutting plane algorithm*, `https://en.wikipedia.org/wiki/Cutting-plane_method`

[2] Wikipedia, *Linear programming*, `https://en.wikipedia.org/wiki/Linear_programming`

[3] Wikipedia, *Simplex algorithm*, `https://en.wikipedia.org/wiki/Simplex_algorithm`

[4] Matteo Fischetti *Lezioni di Ricerca Operativa*

[5] GitHub, *React Plotly*: `https://github.com/plotly/react-plotly.js/`

[6] Project Web Application: `https://riccardolocci.github.io/DS`

[7] CPLEX, *IBM ILOG CPLEX Optimization Studio*, `https://www.ibm.com/it-it/products/ilog-cplex-optimization-studio`

[8] GitHub, *Project Repository*, `https://github.com/riccardolocci/NFO`