Exact Linear and Integer Programming

Tutorial Abstract

Daniel E. Steffy
Department of Mathematics and Statistics
Oakland University
2200 N. Squirrel Rd.
Rochester, Michigan 48309
steffy@oakland.edu

ABSTRACT

This tutorial surveys state-of-the-art algorithms and computational methods for computing exact solutions to linear and mixed-integer programming problems.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Numerical Analysis—*Optimization*; I.1.2 [**Computing methodologies**]: Symbolic and algebraic manipulation—*Algorithms*

Keywords

Linear programming; Integer programming; Hybrid symbolic-numeric computation

TUTORIAL OVERVIEW

A *linear program* (LP) is an optimization problem of minimizing a linear function subject to a finite set of linear inequalities and can be expressed as: $\min_{x\in\mathbb{R}^n}\{c^T x : Ax \geq b\}$, where x is a vector of decision variables. A *mixed-integer program* (MIP) additionally restricts a subset of the decision variables to take integer values, if all decision variables are required to be integers it is called an *integer program* (IP). Many decision problems can be modeled in these formats and a steady line of research has focused on developing and improving algorithms and numerical methods to solve them. Over the past 20 years some commercial software packages have increased their speed by a factor of more than ten thousand due to algorithmic improvements alone.

Most high-performance optimization software used today relies on inexact floating-point arithmetic. Although numerical solutions are adequate for many applications, exact conclusions are sometimes necessary. For example, one component of Thomas Hales' proof of the Kepler Conjecture relies on bounding the value of certain LPs and certain industrial applications, including design verification, also require exact solutions. A number of recent efforts have focused on developing efficient techniques to compute exact solutions and/or rigorous bounds; the primary goal of this tutorial is to survey these efforts, many of which use a hybrid of numerical and exact computation.

Of the aforementioned classes of problems, LPs are considered the easiest. One widely used algorithm, the simplex method, works by pivoting between vertices of the feasible region, a convex polyhedron, until an optimal vertex is found. It computes not only a solution, but a structural description of that solution. One way to solve linear programs exactly is thus to apply numerical computation to identify candidate optimal vertices of the polyhedron and recompute the solution by solving a system of equations exactly. Even if the vertex returned by the numerical solver is not feasible or optimal, it often provides useful information which can be used to warm start additional computations.

Algorithms for mixed-integer programming often rely on solving LPs as a subroutine. The *branch-and-bound* method starts with the LP relaxation of the problem, dropping the integrality constraints, and successively subdivides the search space, solving a sequence of related LPs to arrive at the solution. Computation of LP dual bounds often allows large portions of the search space to be eliminated based on the objective value of known feasible solutions. The *cutting plane* method also starts with the LP relaxation; instead of subdividing the feasible region, it iteratively adds additional inequality constraints in an effort to approximate the convex hull of the feasible solutions, always working with a relaxation of the original feasible region. It continues solving LPs and adding new inequalities until a solution satisfying the integrality constraints is discovered. In practice these methods are integrated together as a *branch-and-cut* method, also incorporating a number of other modules such as preprocessing and heuristics for finding feasible solutions. Due to the complex nature of the overall solution algorithm, there are many different ways in which numerical errors can creep in and invalidate the result.

Avoiding errors by calling an exact LP solver at every step of the branch-and-bound method is a natural strategy. However, the computation of safe dual bounds using interval arithmetic or other fast methods can allow many portions of the search space to be safely eliminated, only leaving a relatively small number of exact LPs to be solved. For the cutting plane method, rounding errors must be avoided when computing the newly added inequalities, otherwise they may cut off feasible solutions. Some recent studies have applied directed rounding to safely generate valid cutting planes using floating-point arithmetic. In many cases exact solutions and safe bounds can be computed with a surprisingly low overhead beyond the cost of the existing numerical methods. Still, there are many open questions and further areas for improvement that will also be discussed in this tutorial.

Little prior experience with optimization is assumed and,

Copyright is held by the author/owner(s).
ISSAC'13, June 26–29, 2013, Boston, Massachusetts, USA.
ACM 978-1-4503-2059-7/13/06.

in addition to describing exact methods, this tutorial also serves as a self-contained introduction to linear and integer programming algorithms. With this goal in mind, part of the tutorial discusses ideas widely applied to tailor the general algorithms into highly efficient problem specific algorithms, including the dynamic generation of variables and constraints and methods for reformulation and decomposition.