Available online at www.sciencedirect.com

SCIENCE DIRECT®

Discrete Optimization 2 (2005) 219–228

DISCRETE OPTIMIZATION

www.elsevier.com/locate/disopt

Simple lifted cover inequalities and hard knapsack problems

Brady Hunsaker[a], Craig A. Tovey[b,*]

[a]*Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA*
[b]*School of ISYE and College by Computing, Georgia Institute of Technology, Atlanta 303320205, USA*

Received 29 April 2004; received in revised form 22 June 2005; accepted 27 June 2005
Available online 15 September 2005

Abstract

We consider a class of random knapsack instances described by Chvátal, who showed that with probability going to 1, such instances require an exponential number of branch-and-bound nodes. We show that even with the use of simple lifted cover inequalities, an exponential number of nodes is required with probability going to 1.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Integer programming; Branch and bound; Average case; Cover inequality; Knapsack; Lifting

0. Introduction

It is not surprising that there exist integer programming (IP) instances for which solution by branch-and-bound requires an exponential number of nodes, since integer programming is an NP-complete problem while the linear programs solved at each branch-and-bound node are polynomially solvable. Examples of such instances were given by Jeroslow [5], who presented a set of simple instances of the knapsack problem which require an exponential number of branch-and-bound nodes when branching on variables, and by Chvátal [1], who considered a class of random instances of the knapsack problem and showed that with probability converging to 1, such a random instance requires exponentially many branch-and-bound nodes to solve.

Most modern IP solvers use branch-and-cut algorithms, which combine branch-and-bound with the use of cutting planes. Gu et al. [4] considered solving the knapsack problem with branch-and-cut. They presented a set of instances that require an exponential number of branch-and-bound nodes even with the addition of simple lifted cover inequalities. More recent work in proving exponential worst-case bounds in the presence of various cutting planes has been done by Dash [2], who proved worst-case exponential bounds in the presence of lift-and-project cuts, Chvátal–Gomory inequalities, and matrix cuts as described by Lovász and Schrijver.

The work of Gu et al. and Dash is similar to Jeroslow's work in that specific "worst-case" examples are presented. In this paper we build on Chvátal's results, which are concerned with average-case performance over a class of random instances. We add all simple lifted cover inequalities to his formulation and show that an exponential number of branch-and-bound nodes is required with probability converging to 1. This result is not suggested by the NP-hardness of binary knapsack problems, because cover inequality separation for these problems is NP-hard [6].

* Corresponding author.
E-mail address: ctovey@isye.gatech.edu (C.A. Tovey).

1572-5286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.disopt.2005.06.003

1. Statement of the result

Following Chvátal [1], we consider the following class of knapsack instances:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} a_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i \leqslant \left\lfloor \frac{\sum_{i=1}^{n} a_i}{2} \right\rfloor \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, n, \end{aligned} \tag{1}$$

where the coefficients a_i are integers selected independently and uniformly such that $1 \leqslant a_i \leqslant 10^{n/2}$.

For ease of discussion, we denote the right-hand-side of the inequality by $r \equiv \lfloor \sum_{i=1}^{n} a_i / 2 \rfloor$ and the upper bound on coefficients by $B \equiv 10^{n/2}$.

Rather than a standard branch-and-bound framework, Chvátal considered a slight generalization, a class of algorithms that he called *recursive algorithms*. These have the capabilities of branching, fathoming, dominance, and improving the current solution. In particular, branching is performed on a single variable, though the selection of branching variable and the process of exploring nodes may be arbitrary. In terms of branch-and-bound, dominance allows the removal of a node if there is another node with the same set of fixed variables that has—considering only the fixed variables—at least as much slack in the constraint and at least as good an objective value. For a precise definition of this class of algorithms, see [1].

We will present our results using the language of branch-and-bound, though our results do apply to Chvátal's class of recursive algorithms.

Theorem 1 (*Chvátal*). *With probability converging to* 1 *as* $n \to \infty$, *every recursive algorithm* (*as described in the previous paragraph*) *operating on an instance of* (1) *will create at least* $2^{n/10}$ *nodes in the process of solving.*

For a knapsack problem with constraint $\sum_{i=1}^{n} a_i x_i \leqslant b$, a *cover* is a set $C \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in C} a_i > b$. A *minimal cover* is a cover C such that no subsets of C are covers. A minimal cover C defines the following *cover inequality*, which is a valid inequality for the knapsack problem:

$$\sum_{i \in C} x_i \leqslant |C| - 1.$$

Although cover inequalities are not facet-defining in general, they can be strengthened to form facet-defining inequalities through a process called *lifting*. We will consider a special case of a lifted cover inequality called a *simple lifted cover*. Given a cover C, a simple lifted cover inequality has the form

$$\sum_{i \in C} x_i + \sum_{i \notin C} \alpha_i x_i \leqslant |C| - 1.$$

The values α_i are called *lifted coefficients* and are determined through a process called *sequential lifting*. See Gu et al. [3] or Wolsey [8] for discussions of lifted cover inequalities. Here we describe the process briefly for simple lifted cover inequalities.

Definition 2. The sequential lifting process for simple cover inequalities is as follows. Let C be the cover. Let the indices not in C be ordered arbitrarily $i_1, i_2, \ldots, i_m$.

1. Initialize $K = \emptyset$, $a = 1$.
2. Let $j = i_a$.
3. Determine lifted coefficient α_j as follows:

$$\alpha_j = |C| - 1 - \max \left\{ \sum_{i \in C} x_i + \sum_{k \in K} \alpha_k x_k : x \in S, x_j = 1 \right\}, \tag{2}$$

   where S is the set of feasible integer solutions to the original knapsack problem.

Download English Version:

https://daneshyari.com/en/article/10523843

Download Persian Version:

https://daneshyari.com/article/10523843

Daneshyari.com