

Sequential Lifting of General Integer Variables for Integer Programs

Todd Easton^{1*} and Talia Gutierrez²

¹Industrial and Manufacturing Systems Engineering, Kansas State University, 2037 Durland Hall, Manhattan, Kansas 66506, USA

²XTO Energy, Kansas State University, 2037 Durland Hall, Manhattan, Kansas 66506, USA

Abstract

Lifting integer variables is a widely used technique to create strong cutting planes. In 1975, Wolsey introduced a method to compute the exact sequential lifting coefficients of bounded integer variables by solving many integer programs. This paper presents a new technique to perform exact sequentially up and down lifting of general integer variables. The technique requires solving only a single branching tree. Some computational results demonstrate that this new sequential lifting technique performed approximately 11 times faster than Wolsey's technique.

Keywords: Integer Programming; Lifting; Polyhedral theory; Facets

Introduction

Define a bounded integer program (IP) as $\max c^T x$ subject to $Ax \leq b$, $0 \leq x \leq u$, $x \in \mathbb{Z}^n$. IPs are NP-complete [1] and IPs are typically solved by branch and bound [2], which has an exponential run time. A common method that typically decreases the computational effort of branch and bound involves the addition of valid inequalities to the IP formulation. This paper presents a new technique to perform exact sequential lifting, which modifies a weak valid inequality into a stronger valid inequality.

First, define an IP's feasible region to be $P = \{x \in \mathbb{Z}^n : Ax \leq b, 0 \leq x \leq u\}$. (If $x \geq l$, then through translation one may assume $x \geq 0$.) Let the convex hull of solutions be denoted as $P^{IP} = \text{conv}(P)$. Define the restricted space of P when $x_i = k$ to be $P_{x_i=k} = \{x \in P : x_i = k\}$ and $P_{x_i=k}^{IP} = \text{conv}(P_{x_i=k}^{IP})$. For convenience, this restricted space definition can be extended to any number of equalities or inequalities in the obvious manner. That is, let $K = \{k_1, \dots, k_{|F|}\} \subset \mathbb{Z}^{|F|}$ and $F \subset N$ where $N = \{1, \dots, n\}$, then the restricted space of P when $x_j = k_j$ for all $j \in F$ is $P_{F=K} = \{x \in P : x_j = k_j \text{ for all } j \in F\}$ with $P_{F=K}^{IP}$ as the respective convex hull.

An inequality $\sum_{i=1}^n \alpha_i x_i \leq \beta$ is said to be valid for P^{IP} if and only if every $x \in P$ satisfies this inequality. A valid inequality induces a face of dimension r in P^{IP} if and only if the maximum number of affinely independent points in P that satisfy $\sum_{i=1}^n \alpha_i x_i \leq \beta$ is $r+1$. A face of dimension r is a facet if the dimension of P^{IP} is equal to $r+1$. Nemhauser and Wolsey [3] provide more information about polyhedral theory with respect to integer programs.

Generating a weak valid inequality is trivial. Lifting, introduced by Gomory [4], is a technique used to modify a weak inequality into a stronger inequality. Numerous researchers have used lifting to create useful cutting planes for various problems. A small subset of these articles is [4-25]. Briefly, let $E \subseteq N = \{1, 2, \dots, n\}$ and $F \subseteq N \setminus E$ be any nonempty set. Now let $\sum_{i \in F} \alpha_i x_i + \sum_{i \in E} \alpha_i x_i \leq \beta$ be a valid inequality of P^{IP} where $0 \leq k_j \leq u_j$ for all $j \in F$. Lifting seeks to create a valid inequality of P^{IP} , which takes the form $\sum_{i \in F} \alpha_i' x_i + \sum_{i \in E} \alpha_i x_i \leq \beta'$.

There are various types of lifting, such as sequential, simultaneous, approximate and exact, up, middle and down lifting techniques. For distinction, α_\wedge and α_\vee denote the coefficients that are obtained through up and down lifting, respectively, and α represents a lifted coefficient that is either up or down lifted.

Of these lifting techniques, sequential up lifting is the most widely

used technique [7, 8, 25-29]. Sequential lifting requires $|F|=1$ and up lifting assumes $\sum_{i \in E} \alpha_i x_i \leq \beta$ is valid inequality for $P_{x_j=0}^{IP}$, where $k_j=0$ for all $j \in F$. Therefore, sequential up lifting assumes that $\sum_{i=2}^n \alpha_i x_i \leq \beta$ is valid for $P_{x_1=0}^{IP}$ and seeks to create an inequality of the form $\alpha_1 x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta$.

Sequential down lifting assumes $\sum_{i=2}^n \alpha_i x_i \leq \beta$ is valid for $P_{x_1=u_1}^{IP}$ and obtains an inequality of the form $\alpha_\vee x_1 + \sum_{i=2}^n \alpha_i x_i \leq \gamma$ where γ is typically equal to $\beta - \alpha_\vee u_1$. There is also a sequential lifting when $\sum_{i=2}^n \alpha_i x_i \leq \beta$ is valid for $P_{x_1=k^*}^{IP}$ where $k^* \in \{1, \dots, u_1-1\}$, which is roughly a combination of both up and down lifting.

Lifting can be approximate or exact. Exact lifting finds the strongest α' and/or β' possible. Thus, exact sequential up lifting finds the maximum value of $\alpha_{1\wedge}$ that still maintains the validity of the inequality. If such a value of $\alpha_{1\wedge}$ is obtained, then the dimension of the face induced by the sequentially lifted inequality increases by at least 1 in the unrestricted polyhedron.

Approximate lifting techniques obtain coefficients that maintain valid inequalities, but at times these coefficients could be strengthened. These approximate lifting techniques do not necessarily increase the dimension of the induced face, but rather trade the computational effort required for exact lifting for a theoretically weaker inequality. Some such approximate lifting results include sequential up lifting [7] and sequence independent lifting [6,30-32]. Other work has used a linear relaxation or just a portion of the original problem to approximate the lifting coefficients [29,33].

Simultaneous lifting requires $|F| \geq 2$. Zemel [34] provided an exact technique to simultaneously up lift sets of binary integer variables. This technique solves an exponential number of integer programs and then finds extreme points of the polar created from the solutions to these integer programs. This method yields numerous inequalities,

***Corresponding author:** Todd Easton, Industrial and Manufacturing Systems Engineering, Kansas State University, 2037 Durland Hall, Manhattan, Kansas 66506, USA, Tel: 785 532-3478; E-mail: teaston@ksu.edu

Received February 18, 2015; Accepted May 12, 2015; Published May 14, 2015

Citation: Easton T, Gutierrez T (2015) Sequential Lifting of General Integer Variables for Integer Programs. Ind Eng Manage 4: 158. doi:10.4172/2169-0316.1000158

Copyright: © 2015 Easton T, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

but is computationally intensive. Recently, [17] created a linear-time algorithm that exactly up lifts sets of binary variables into a cover inequality generated from a single binary knapsack constraint (a single nonnegative less than or equal to constraint, $\{x \in \{0, 1\}^n: ax \leq b, a \geq 0\}$). Observe that the aforementioned sequence independent lifting references could also be viewed as an approximate method to up lift sets of integer variables.

Prior to this research Wolsey [29] provided the only known technique to exactly perform sequential lifting of general integer variables. His method requires the solution to u_i integer programs. This paper presents a new technique to perform exact sequential lifting of general integer variables. This new technique requires the solution to only a single branching tree for both up and down lifting and two branching trees if lifting over a valid inequality when $x_i = k^*$ where $k^* \in \{1, \dots, u_i - 1\}$.

The remainder of the paper is organized as follows. Section 2 presents the new technique to sequentially up lift integer variables and compares this to Wolsey's existing technique. Some computational results are contained in Section 3 that demonstrate that this new method is faster than Wolsey's method. A conclusion and some directions for future research are discussed in Section 4.

Up Lifting Integer Variables

Wolsey [29] introduced a method to exactly lift general integer variables. This technique is derived directly from his theorem which states:

Theorem 2.1: Given a general integer programming instance, let $\sum_{i=2}^n \alpha_i x_i \leq \beta$ be a valid inequality when $x_1 = k^*$ where k^* is integer and $0 \leq k^* \leq u_1$. For each $0 \leq k \leq u_1$ and $k \neq k^*$ define

$$Z_W^k = \begin{array}{ll} \text{Maximize} & \sum_{i=2}^n \alpha_i x_i \\ \text{Subject to} & Ax \leq b \\ & x_1 = k \\ & 0 \leq x_i \leq u_i \quad \text{and } x_i \in \mathbb{Z}_+ \quad \forall i = 1, \dots, n. \end{array}$$

If, for a specific k , the problem is infeasible, then let $Z_W^k = \infty$. Now let

$$\alpha_{1^*} = \min_k \frac{Z_W^k - \beta}{k - k^*} \text{ or } \alpha_{1^*} = \infty \text{ if } k^* = u_1$$

and

$$\alpha_{1^*} = \min_{k < k^*} \frac{Z_W^k - \beta}{k^* - k} \text{ or } \alpha_{1^*} = -\infty \text{ if } k^* = 0.$$

The sequentially lifted inequalities are $\alpha_1 x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta + k^* \alpha_{1^*}$ and such an inequality is valid for any α_1 such that $\alpha_{1^*} \leq \alpha_1 \leq \alpha_{1^*}$. Furthermore, if $\sum_{i=2}^n \alpha_i x_i \leq \beta$ defines an r dimensional face of $P_{x_1=k^*}^{IP}$, then $\alpha_1 x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta + k^* \alpha_{1^*}$ defines a face of dimension at least $r+1$ as long as α_{1^*} is finite and $\alpha_1 = \alpha_{1^*}$ or α_{1^*} .

As mentioned in the introduction, of the many possible values of k^* the most frequently used is $k^*=0$, which is called up lifting. That is, a valid inequality is obtained when $x_1=0$, and x_i is up lifted into this inequality. From Theorem 2.1 it is easy to see that Wolsey's up lifting method requires the solution to u_i integer programs and the right hand side β does not change.

Now the attention turns toward the purpose of this paper, which produces a new technique to up lift general integer variables. The algorithm creates an inequality that increases the dimension of the face, but may not necessarily be a valid inequality (guess too high of a value for α_{1^*}). A modifiable branching tree is then used to check to see if the inequality is valid. If the inequality is not valid or equivalently there exists a point violating the proposed inequality, then α_{1^*} is decreased according to this feasible point and the objective function is reset with this new α_{1^*} . This process continues until all nodes are fathomed.

The input to the Sequential Lifting Algorithm (SLA) is composed of the constraints and bounds of a general integer program and an inequality $\sum_{i=2}^n \alpha_i x_i \leq \beta$ that is valid for $P_{x_1=0}^{IP}$. SLA up lifts x_1 by finding an α_{1^*} that is the maximum value such that $\alpha_{1^*} x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta$ is valid for P^{IP} .

The Sequential Lifting Algorithm (SLA)

$$\text{Set } \alpha_{1^*} := M \text{ where } M > \beta + \sum_{i=2}^n |\alpha_i| u_i$$

Begin a modified branch and bound tree by letting the following LP be the unfathomed root node of the tree.

$$\begin{array}{ll} \text{Maximize} & \alpha_{1^*} x_1 + \sum_{i=2}^n \alpha_i x_i \\ \text{Subject to} & Ax \leq b \\ & x_1 \geq 1 \\ & 0 \leq x_i \leq u_i \quad \text{and } x_i \in \mathbb{Z}_+ \quad \forall i = 1, \dots, n. \end{array}$$

While there exist unfathomed nodes in the branch and bound tree, begin. Select an unfathomed node and solve the linear relaxation with the solution denoted by Z_{LR}^* and x_{LR}^* .

If $Z_{LR}^* \leq \beta$ or the linear relaxation is infeasible, then fathom the node

If the solution to the node is an integer solution with $Z_{LR}^* > \beta$ then begin

$$\text{Let } \alpha_{1^*} := \frac{\beta - \sum_{i=2}^n \alpha_i x_{LRi}^*}{x_{LR1}^*}$$

Change x_1 's objective coefficient in every pendant node to α_{1^*} . Create a single child constraint with no additional constraints.

end if

If $Z_{LR}^* > \beta$ and x^* is non-integer, then begin

Create two new children nodes by branching on any x_{LRi}^* that is non-integer.

One node has the parent's LP with the added inequality $x_i \leq \lfloor x_{LRi}^* \rfloor$

The other child's node has the parent's LP and adds the inequality $x_i \geq \lceil x_{LRi}^* \rceil$.

end if

end while

Output

If $\alpha_{1^*} = M$, then $\alpha_{1^*} := \infty$.

Report $\alpha_{1^*} x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta$, a valid inequality for P^{IP} .

The following example demonstrates this algorithm and provides some fundamental insights into the differences between Wolsey’s method and SLA. The branching tree is explored according to a depth first left strategy.

Example 2.2: Consider an integer program with feasible region defined by

$$\begin{aligned} 6x_1 + 5x_2 &\leq 27 \\ 5x_1 + 6x_2 &\leq 28 \\ x_1, x_2 &\in \mathbb{Z}_+. \end{aligned}$$

First observe that $x_2 \leq 4$ is a valid inequality of $P_{x_1}^{IP}$ that defines a face of dimension 0 from the point (0, 4). Figure 1 provides the branching tree to exactly up lift x_1 . SLA begins with the root node given by the LP

$$\begin{aligned} \text{Max } Mx_1 + x_2 \\ 6x_1 + 5x_2 &\leq 27 \\ 5x_1 + 6x_2 &\leq 28 \\ x_1, x_2 &\in \mathbb{R}_+. \end{aligned}$$

Solving this LP results in $Z_{LR}^* = 4.5M$ and $x_{LR}^* = (4.5, 0)$. Two new child nodes are created. One adds on the constraint $x_1 \leq 4$ and the other $x_1 \geq 5$. The second node’s solution is $Z_{LR}^* = 4M + .6$ and $x_{LR}^* = (4, .6)$. Two new nodes are added, one with the constraint $x_2 \leq 0$ and the other with $x_2 \geq 1$. The LP solution to the left node, node 3, is $Z_{LR}^* = 4M$ and $x_{LR}^* = (4, 0)$. This is an integer solution and $4M > \beta = 4$. So α_{1A} is changed so that $4 \times \alpha_{1A} + 0 = 4$ and $\alpha_{1A} = 1$. The objective function of all remaining LPs is changed to $x_1 + x_2$. Unlike branch and bound, this node is not fathomed; instead a single child node is added with the same LP as its parent, but with the updated objective value. The solution to node 4 is $Z_{LR}^* = 4$ and $x_{LR}^* = (4, 0)$. This node is fathomed, not because it is integer, but because $Z_{LR}^* \leq 4$.

Node 5’s solution is $Z_{LR}^* = 5 > 4$ and $x_{LR}^* = (2, 3)$. The α_{1A} is updated so that $2 \times \alpha + 3 = 4$ and $\alpha_{1A} = \frac{1}{2}$. All remaining nodes have the new objective of $\frac{1}{2}x_1 + x_2$ and a single child node is added. The complete tree is shown in Figure 1. The final value of $\alpha_{1A} = \frac{1}{2}$ sequentially up lifted inequality being $\frac{1}{2}x_1 + x_2 \leq 4$.

Figure 1 provides a graphical view of lifting x_1 into $x_2 \leq 4$ and is used to discuss the difference between this new technique and Wolsey’s technique. Wolsey’s method uses a single IP to find the maximum

value of $\sum_{i=2}^n \alpha_i x_i \leq \beta$ for each $x_i = k$ for $k=1, 2, \dots, u_i$. In this example the points that give the optimal solutions are (1, 3), (2, 3), (3, 1) and (4, 0). Wolsey’s method uses this point to obtain the candidate α_{1A} where this point would meet the new inequality at equality. In this case, the candidate α_{1A} values would have been $1, \frac{1}{2}, 1$ and 1 , respectively.

The method then takes the minimum of these values as $\alpha_{1A} = 1$, which guarantees a valid inequality and increases the dimension by at least 1. In Figure 1, $x_1=1, x_1=3$ and $x_1=4$ all have identical candidate α_{1A} values, because they are on the same line. Essentially, Wolsey’s method checks every possible extreme point and accepts the best α_{1A} value that maintains validity. Clearly some work is typically wasted.

In contrast, SLA begins by starting with an objective function that is nearly parallel to the x_2 axis. As soon as an integer solution is discovered with value larger than 4, α_{1A} is changed to intersect this point. In this case, it happens at node 4 with the point (4, 0). The inequality now tested for validity is $x_1 + x_2 \leq 4$, which intersects (4,0). If no integer solutions satisfy $x_1 + x_2 \leq 4$, then the inequality is valid. However, node 5 finds such a violating point (2,3) and the objective function changes to $\frac{1}{2}x_1 + x_2 \leq 4$. Observe that (2, 3) meets this inequality at equality. Eventually, the tree is fathomed and there are no integer solutions with a value larger than 4 to the objective function. Thus, the inequality is valid.

Essentially SLA maintains inequalities that guarantee to increase the dimension of the face (an additional point meets the inequality at equality), but may not necessarily be valid. Once the algorithm terminates, the inequality is clearly valid since there does not exist a feasible point with a value larger than β . Since α_{1A} is calculated from a feasible point, the dimension of the lifted inequality’s face also increases in the unrestricted polyhedron. SLA seeks to obtain the lifting coefficient by coming through the inside of the polyhedron; whereas, Wolsey’s method seeks to obtain the lifting coefficient by identifying all extreme points.

One may attempt to incorrectly argue that SLA is Wolsey’s method. The erroneous argument states that the root node should have u children with each branch having x_i set to a different integer between 0 and u . Such an IP would only generate a single objective function. This single IP could not accurately calculate the lifting coefficient. Thus, SLA is not merely an extension of Wolsey’s method, but a new method to perform sequential lifting.

The main theoretical result of the paper, which states that SLA terminates with a valid inequality, can now be presented. Furthermore, this inequality results in the same coefficient as Wolsey’s method, and, under a feasibility assumption, the dimension of the inequality increases over the non-restricted polyhedron.

Theorem 2.3: Let $\sum_{i=2}^n \alpha_i x_i \leq \beta$ be a valid inequality of $P_{x_1=0}^{IP}$. Given a bounded integer program, the Sequential Lifting Algorithm terminates and reports a valid inequality, $\alpha_{1A} x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta$, of P^{IP} . Furthermore, the coefficient α_{1A} returned from SLA is the equal to the α_{1A} generated from Wolsey’s method. If $\sum_{i=2}^n \alpha_i x_i \leq \beta$ defines a face of dimension r in $P_{x_1=0}^{IP}$, then $\alpha_{1A} x_1 + \sum_{i=2}^n \alpha_i x_i \leq \beta$ defines a face of dimension at least $r+1$ in P^{IP} as long as α_{1A} is finite.

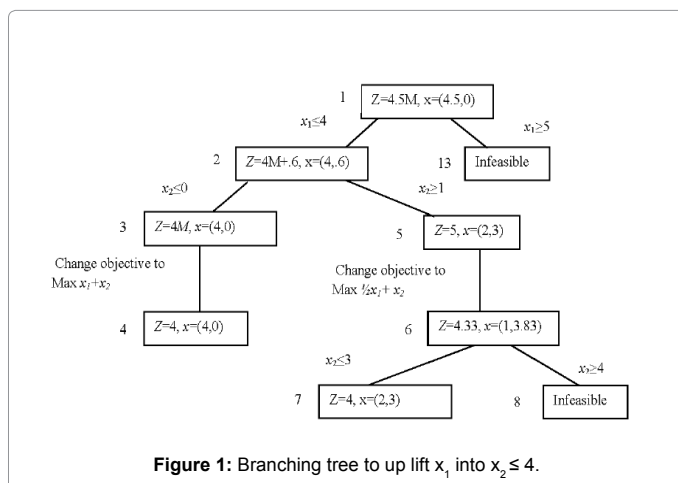


Figure 1: Branching tree to up lift x_1 into $x_2 \leq 4$.

Proof: Assume a node in the branching tree has an integer solution with $z > \beta$. Then $\frac{\beta}{\sum_{i=1}^n \alpha_i x_i}$. Clearly, this new value of α_{i_A} is strictly less than the previous value of α_{i_A} . Thus, the values of α_{i_A} monotonically decrease as the branching tree progresses.

Due to the monotonically decreasing nature of the α_{i_A} , any integer solution contained in the feasible solution space of a node that is fathomed because $z \leq \beta$ satisfies the returned inequality $\alpha_{i_A} x_i + \sum_{i=2}^n \alpha_i x_i \leq \beta$. There are no integer solutions in the solution space of an infeasible node. Since SLA only fathoms nodes under these two conditions and SLA terminates when every node is fathomed, there does not exist an $x \in P$ such that $\alpha_{i_A} x_i + \sum_{i=2}^n \alpha_i x_i > \beta$. Consequently, SLA generates a valid inequality upon termination.

To show termination, observe that the IP is bounded. Thus, there can be at most $\prod_{i=1}^n (u_i + 1)$ integer solutions. The branching step can be implemented at most $2 \prod_{i=1}^n (u_i + 1)$ times. In such a scenario, the solution to every child would either be integer or infeasible. Any solution to an integer with value larger than β , creates an additional child. The solution x values of this child node must be identical to its parent's solution and thus the objective value is equal to β . This new node is then fathomed. Thus, each node is fathomed in a finite number of steps and SLA terminates.

Assume SLA terminates with $\alpha_{i_A} = \infty$. In such a case, no integer solutions were encountered in the branching tree. Thus, the solution to Maximize $\sum_{i=2}^n \alpha_i x_i$ subject to $Ax \leq b, x_i \geq 1, 0 \leq x_i \leq u_i$ and $x_i \in \mathbb{Z}_+, \forall i = 1, \dots, n$ is infeasible and both Wolsey's method and SLA sets $\alpha_{i_A} = \infty$.

Assume SLA terminates with α_{i_A} being finite. Let x_{SLA}^* be the integer solution that determined the reported value of α_{i_A} . Since $x_{SLA}^* = k$ an integer between 1 and u_1 , Wolsey's method solves an integer program of the form Maximize $\sum_{i=2}^n \alpha_i x_i$ subject to $Ax \leq b, x_i \geq 1, 0 \leq x_i \leq u_i$ and $x_i \in \mathbb{Z}_+, \forall i = 1, \dots, n$. The solution to this integer program must be identical to x_{SLA}^* or there is a contradiction to optimality of either an LP or IP. Thus, Wolsey's coefficient is at most $\frac{\beta - Z_W^k}{k} = \frac{\beta - (\sum_{i=2}^n \alpha_i x_{i,SLA}^*)}{x_{i,SLA}^*} = \alpha_{i_A}$.

For contradiction, assume Wolsey's coefficient is less than α_{i_A} . Then there exists $k' \in \mathbb{Z}_+$ such that $\frac{\beta - Z_W^{k'}}{k'} < \alpha_{i_A}$. Let $x^{*k'}$ be the solution to Wolsey's IP when $x_1 = k'$. Clearly $\beta < k' \alpha_{i_A} + Z_W^{k'} = k' \alpha_{i_A} + \sum_{i=2}^n \alpha_i x_i^{*k'}$, since $x^{*k'}$ is integer, this contradicts the prior argument on the validity of SLA. Consequently, SLA and Wolsey's method both produce identical coefficients.

Finally, assume $\sum_{i=2}^n \alpha_i x_i \leq \beta$ defines a face of dimension r in $P_{x_1=0}^{IP}$ and so there exists $r+1$ affinely independent points in $P_{x_1=0}$ that satisfy $\sum_{i=2}^n \alpha_i x_i = \beta$. Assume SLA terminates with a finite α_{i_A} . Then there exists some x_{SLA}^* that determined the reported value of α_{i_A} .

Furthermore, $x_{SLA}^* \geq 1$ and $\alpha_{i_A} x_{SLA}^* + \sum_{i=2}^n \alpha_i x_{i,SLA}^* = \beta$. Clearly, x_{SLA}^* is affinely independent from these other $r+1$ points and thus the face of the lifted inequality is at least $r+1$ and the result follows.

A few comments can now be made that describe how to improve SLA. Theoretically, SLA requires $M > \beta + \sum_{i=2}^n |\alpha_i| u_i$. Thus any integer solution is larger than β and α_{i_A} changes. From a computational standpoint, if a feasible point with $x_1 \geq 1$ is known, then the starting value of α_{i_A} can be changed so that this feasible point satisfies the up lifted in equality at equality. Multiple knapsack problems (all nonnegative coefficients and less than or equal to constraints) provide an excellent example. In such problems, the point $(u_1, 0, 0, \dots, 0)$ is always feasible and the starting value of α_0 can be set equal to $\frac{\beta}{u_1}$. In Example 2.2, the point $(4, 0)$ is trivially valid and α_{i_A} could have been set to 1 to start SLA. This may reduce computational effort.

An additional advantage to SLA is that it may enable exact up lifting over unbounded integer programs. Wolsey's method does not enable this type of lifting as solving an infinite number of IPs is not possible. However, SLA may not terminate for an unbounded integer program as a critical assumption for the finite termination proof is no longer valid. For instance, implementing SLA to lift x_1 into $x_2 \leq 0$ for $\{x \in \mathbb{Z}^2 : x_1 - \sqrt{2}x_2 \leq 0, x_1, x_2 \geq 0\}$ would never terminate. Consequently, if SLA terminates for an unbounded integer program, then SLA determines the correct up lifting coefficient.

Sequential Down Lifting and Lifting when $x_1 = k^*$

With a fundamental understanding of up lifting, it is now straightforward to modify SLA into algorithms that can down lift and lift over a valid inequality when $x_1 = k^*$. For brevity two new algorithms and theorems for these other types of lifting are not presented. Rather changes are described that enable the reader to trivially extend SLA to these other two sequential lifting types through variable substitution. An example of each algorithm is also provided to aid with comprehension.

Down lifting can be viewed as substituting x_1' for x_1 where $x_1' = u_1 - x_1$ and then up lifting x_1' . Thus, down lifting creates a valid inequality of the form $\alpha_{i_A} (u_1 - x_1) + \sum_{i=2}^n \alpha_i x_i \leq \beta$. Now as in the up lifting case, the left hand side of this inequality becomes the objective function, Maximize $\alpha_{i_A} (u_1 - x_1) + \sum_{i=2}^n \alpha_i x_i$, and the constraints are the constraints of the integer program with the additional constraint of $x_1 \leq u_1 - 1$.

When an integer solution is found that is larger than β , the linear relaxation point, x_{LR}^* , is substituted into the equation $\alpha_{i_A} (u_1 - x_{LR1}^*) + \sum_{i=2}^n \alpha_i x_i^* = \beta$ to solve for the new estimate of α_{i_A} . Everything else is identical and trivial to determine with knowledge of this substitution. The following example provides a demonstration of down lifting using SLA.

Example 3.1: Returning to Example 2.2, if $x_1 = 4$, then $x_2 \leq 0$ is a valid inequality that defines a face of dimension 0 due to the point $(4, 0)$. The down lifted branching tree is shown in Figure 2 using a depth first left node evaluation strategy. The first integer solution is found at node 2 with a $Z = 4M + 4 > 0 = \beta$ and $x = (0, 4)$. Solving for α_{i_A} in $\alpha_{i_A} (4 - 0) + 4 = 0$ results in $\alpha_{i_A} = -1$.

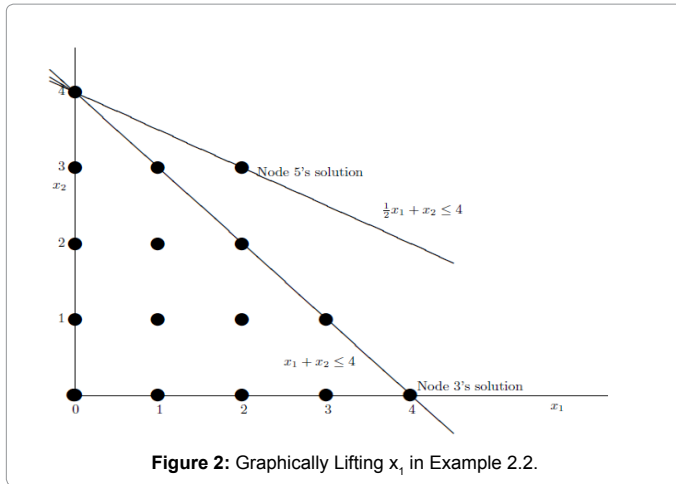


Figure 2: Graphically Lifting x_1 in Example 2.2.

All objective functions are now changed to $-1(4-x_1)+x_2$.

Node 3 also is integer with $Z=1>\beta$ and $x=(2, 3)$. Solving for α_{1v} in $\alpha_{1v}(4-2)+3=0$ results in $\alpha_{1v} = \frac{-3}{2}$. Nodes 4 and 6 require branching, but eventually every node is either infeasible or $Z_{LR}^* \leq 0$. Thus, SLA terminates with $\alpha_{1v} = \frac{-3}{2}$. The reported valid inequality is $\frac{-3}{2}(4-x_1) + x_2 \leq 0$ or equivalently $\frac{3}{2}x_1 + x_2 \leq 6$. Observe that this inequality has dimension 1 because the point (2, 3) meets this inequality at equality and is affinely independent from (4, 0).

The attention now turns toward sequentially lifting over a valid inequality when $x_i = k^*$ where $0 < k^* < u_i$, which is a combination of both down and up lifting. The substitution for down lifting is $x_i' = k^* - x_i$. The down lifting portion adds on the constraint $x_i \leq k^* - 1$ to the root node. For an integer solution larger than β , a new value of α_{1v} is obtained by solving $\alpha_{1v}(k^* - x_{LR1}^*) + \sum_{i=2}^n \alpha_i x_{LRi}^* = \beta$.

In contrast, the up lifting adds on the constraint $x_i \geq k^* + 1$ to the root node and results in a lifted inequality taking the form $\alpha_{1u}(x_1 - k^*) + \sum_{i=2}^n \alpha_i x_i = \beta$. When an integer solution is encountered with a $Z_{LR}^* > \beta$, a new α_{1u} is obtained by solving $\alpha_{1u}(x_{LR1}^* - k^*) + \sum_{i=2}^n \alpha_i x_{LRi}^* = \beta$.

In contrast to up and down lifting, middle lifting may not produce a valid inequality. In fact, a middle lifted inequality takes the form $\alpha_1(x_1 - k^*) + \sum_{i=2}^n \alpha_i x_i = \beta$. This inequality is only valid when $\alpha_1 \in [-\alpha_{1v}, \alpha_{1u}]$. Notice that if $-\alpha_{1v} > \alpha_{1u}$, and then one cannot sequentially lift x_1 using middle lifting as any such inequality would be invalid. The following example demonstrates these concepts.

Example 3.2: Reexamining example 2.2 and if $x_1 = k^* = 2$, then $x_2 \leq 3$ is valid inequality that defines a face of dimension 0 from the point (2, 3). To find the proposed up lifted coefficient, set the objective function to $\alpha_{1u}(x_1 - 2) + x_2$ and add the constraint $x_1 \geq k^* + 1 = 3$. The optimal integer solution occurs at (4, 0), which sets $\alpha_{1u} = \frac{3}{2}$. Any remaining nodes are easily fathomed. For the down lifting portion, set the objective function to $\alpha_{1v}(2 - x_1) + x_2$ and add the constraint $x_1 \leq k^* -$

$1 = 1$. The optimal integer solution is (0, 4), which sets $\alpha_{1v} = \frac{-1}{2}$. Since $-\alpha_{1v} = \frac{1}{2} < \frac{3}{2} = \alpha_{1u}$, any inequality of the form $\alpha_1 x_1 + x_2 \leq 3 + 2\alpha_1$ is valid as long as $\alpha_1 \in [\frac{1}{2}, \frac{3}{2}]$.

To help clarify the idea that middle lifting may not result in a valid inequality, observe that $x_2 \leq 1$ is valid if $x_1 = k^* = 3$. This inequality defines a face of dimension 0 from the point (3, 1). To find the proposed up lifted coefficient, set the objective function to $\alpha_{1u}(x_1 - 3) + x_2$ and add the constraint $x_1 \geq k^* + 1 = 4$. The optimal integer solution is (0, 4), which sets $\alpha_{1u} = 1$. Any remaining nodes are easily fathomed. For the down lifting portion, the optimal integer solution is (2, 3), which sets $\alpha_{1v} = -2$. Since $-\alpha_{1v} = 2 > 1 = \alpha_{1u}$, no middle lifted inequality is valid. This is easily verifiable in Figure 3 as the point (3,1) is not an extreme point of P^{IP} .

Computational Results

This section demonstrates that SLA is substantially faster than Wolsey's method. All computational results were performed on an Intel Core i7-2600 chip at a 3.4 GHz processor with 8 Gb of RAM and the integer programs were solved using CPLEX 10.0 [35] at its default settings.

The problems chosen for this study were generated from random multiple knapsack polytope $\{x \in \mathbb{Z}_+^n : \sum_{i=1}^n a_{ij}x_j \leq b_i\}$. The a_{ij} were integers selected uniformly between 25 and 1,000. (Note: When $a_{ij} < 25$, then the u_i 's became extremely large and Wolsey's could not finish in a reasonable time. So each $a_{ij} \geq 25$.) The slackness, s , of the constraint determines the right hand side, $b_i = \sum_{j=1}^n s * a_{ij}$. Various values of n were chosen as shown in Table 1.

The computational study starts with the valid inequality $x_1 \leq u_1$ and sequentially up lifts every other variable until the inequality becomes facet-defining. Fifty random knapsack instances are created for each problem size. Table 1 reports the total time in seconds required to lift all of the variables. Thus, when $n=10$, the total time columns represent the time required to sequentially up lift 9 variables for each of the 50 problems or 450 variables with either Wolsey's Algorithm or SLA. In

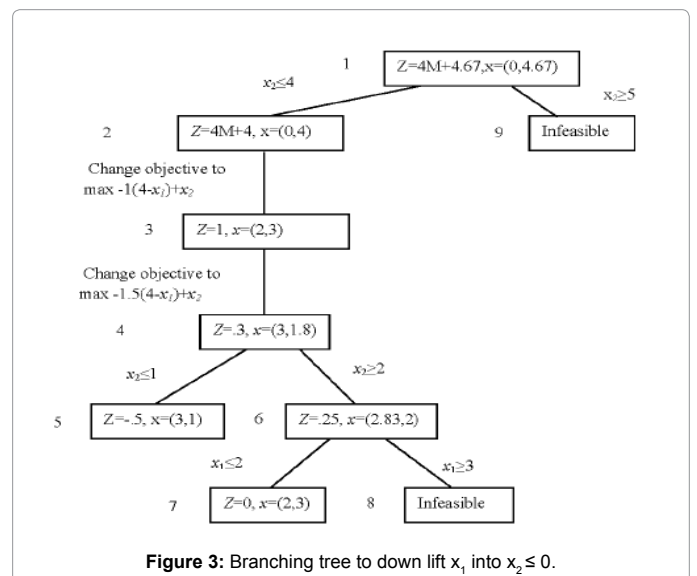


Figure 3: Branching tree to down lift x_1 into $x_2 \leq 3$.

Num of Var	Num of Cons	Slack Coef	Num Var Lifted	Avg Upper Bound	Wolsey's Method		SLA	Proportion SLA Time Improvement
					Total Time Sec.	Total IPs Solved	Total Time Sec.	
10	1	.5	225	8.4	.405	1,901	.143	.353
30	1	.5	725	28.9	54.32	20,985	7.23	.133
50	1	.5	1,225	48.25	11,697.44	59,114	2,215.6	.189
10	1	1	225	17.3	1.06	3,907	.211	.208
30	1	1	725	58.5	981.21	42,442	38.55	.0393
50	1	1	1,225	110.6	2,141.95	124,474	397.12	.1854
10	1	2	225	33.6	19.69	7,559	4.42	.224
30	1	2	725	116.7	510.52	84,609	7.72	.0151
50	1	2	1,225	193.2	14,243.12	236,673	46.52	.0033
10	10	.5	225	2.2	.121	498	.121	0
30	10	.5	725	8.0	11.26	5,763	3.46	.307
50	10	.5	1,225	13.6	41.22	16,603	4.56	.111
10	10	1	225	5	.432	1,125	.133	.308
30	10	1	725	16.4	27.71	11,902	2.98	.108
50	10	1	1,225	27.9	286.46	34,197	4.12	.0144
10	10	2	225	10.7	1.08	2,400	.16	.148
30	10	2	725	33.3	80.10	24,136	5.95	.074
50	10	2	1,225	80.5	11,132.75	98,619	1,058.42	.095
10	25	.5	225	2.0	.206	448	.156	.757
30	25	.5	725	7.2	9.71	5,226	2.98	.307
50	25	.5	1,225	11.7	44.23	14,278	4.56	.103
10	25	1	225	4.1	.403	927	.144	.357
30	25	1	725	9.6	13.78	6,931	1.65	.120
50	25	1	1,225	24.3	650.0	29,766	79.56	.122
10	25	2	225	9.3	1.04	2,099	.221	.213
30	25	2	725	30.2	66.17	21,882	2.54	.0383
50	25	2	1,225	51.1	1,060.2	62,563	34.9	.0329
Average			725	36	1,596	34,112	145	.091

Table 1: Comparing Up Lifting Techniques.

addition, the total number of integer programs that Wolsey's method solved is also reported.

Table 1 shows the computational superiority of SLA. On average SLA lifted variables eleven times faster than Wolsey's method. SLA was at least as fast in every instance. As expected, the larger the average upper bound the greater improvement of SLA. For instance, the largest bounds occurred with the slack coefficient of 2 and when there are fifty variables. The average of these 150 instances shows that SLA is about 25 times faster than Wolsey's method. Some additional computational results [36] demonstrate SLA's superiority over Wolsey's method.

Conclusion and Future Research

Since Wolsey's paper in 1975 and prior to this work, there has only been one method to perform exact sequential lifting. This existing method requires solving u_{i+1} integer programs where u_i is the upper bound for variable x_i . This paper presents a new technique, SLA, to perform exact sequential lifting of bounded integer programs. SLA requires solving a single branching tree for both up and down lifting, and only two branching trees for middle lifting. SLA, under reasonable assumptions, becomes the first method for sequential lifting of unbounded integer programs.

A computational study compared the amount of computational effort required to sequentially up lift many variables using both Wolsey's method and SLA. SLA performed 11 times faster than Wolsey's method. Thus, SLA is both theoretically and computationally superior to the existing method. It is recommended that researchers performing sequential lifting implement SLA instead of Wolsey's method.

The creation of SLA also raises several important research

questions. Can SLA's new method and theory be extended to perform exact simultaneous lifting? Can the concepts behind SLA be extended into a new method to obtain approximate lifting coefficients (terminating SLA prior to each node being fathomed)? Can polynomial time methods be developed to create sequentially lifted inequalities?

Acknowledgment

The authors would like to thank Ricardo Fukasawa for some critical insight into down lifting.

References

- Karp RM (1972) Reducibility among combinatorial problems. Complexity of Computer Computations. The IBM Research Symposia Series 85-103.
- Land A, Doig A (1960) An automatic method of solving discrete programming problems. *Econometrica* 28: 497-520.
- Nemhauser GL, Wolsey LA (1999) Integer and combinatorial optimization, John Wiley and Sons, New York.
- Gomory R (1969) Some polyhedra related to combinatorial problems. *Linear Algebra and its Applications* 2: 451-588.
- Atamtürk A (2003) On the facets of the mixed-integer knapsack polyhedron. *Mathematical Programming* 98: 145-175.
- Atamtürk A (2004) Sequence independent lifting for mixed-integer programming. *Operations Research* 52: 487-490.
- Balas E (1975) Facets of the knapsack polytope, *Mathematical Programming* 8: 146-164.
- Balas, E. and E. Zemel (1978). "Facets of the knapsack polytope from minimal covers," *SIAM Journal of Applied Mathematics* 34: 119-148.
- Balas E, Zemel E (1984) Lifting and complementing yields all facets of positive zero- one programming polytopes. In: Cottle RW (ed.) *Proceedings of the International Conference on Mathematical Programming*, pp. 13-24.

10. Balas E, Ng SM (1989) On the set covering polytope. II, Lifting the facets with coefficients in 0,1,2. *Mathematical Programming* 45: 1-20.
11. Balas E, Fishetti M (1993) Lifting procedure for the asymmetric traveling salesman polytope and a large new class of facets. *Mathematical Programming* 58: 325-352.
12. Carr R (1996) Separating over classes of TSP inequalities defined by 0 node-lifting in polynomial time *Integer Programming and Combinatorial Optimization*. 5th International IPCO Conference Proceedings, pp. 460-474.
13. Cho CD, Padberg MW, Rao MR (1983) On the uncapacitated plant location problem. II. Facets and lifting theorems. *Mathematics of Operations Research* 8: 590-612.
14. Dahan X, Maza MM, Schost. E, Wu W, Xie Y (2005) Lifting techniques for triangular decompositions. *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation ISSA 05*.
15. De Farias IR JR, Johnson EL, Nemhauser GL (2002) Facets of the complementarity knap- sack polytope. *Mathematics of Operations Research* 27: 210-226.
16. De Simone C (1990) Lifting facets of the cut polytope. *Operations Research Letters* 9: 341-344.
17. Easton T, Hooker K (2008) Simultaneously lifting sets of variables and scaled multiple cover inequalities for knapsack polytopes. To appear in *Discrete Optimization* 5: 254-261.
18. Felici G, Gentile C (2003) Zero-lifting for integer blocks structured problems. *Journal of Combinatorial Optimization* 7: 161-167.
19. Gu Z, Nemhauser GL, Savelsbergh MWP (1998) Lifted cover inequalities for 0-1 integer programs: computation. *Journal of Computing* 10: 427-437.
20. Gu Z, Nemhauser GL, Savelsbergh MWP (1999) Lifted cover inequalities for 0-1 integer programs: computation. *Mathematical Programming* 85: 439-467.
21. Koster A, Hoesel S, Kolen A (1998) The partial constraint satisfaction problem: Facets and lifting. *Operations Research Letters* 23: 89-98.
22. Nemhauser GL, Vance PH (1994) Lifted cover facets of the 0-1 knapsack polytope with GUB constraints. *Operations Research Letters* 16: 255-263.
23. Park K (1997) Lifting cover inequalities for the precedence-constrained knapsack problem. *Discrete Applied Mathematics* 72: 219-241.
24. Richard J, De Farias I, Nemhauser G (2003) Lifted inequalities for 0-1 mixed integer programming: Superlinear lifting *Integer Programming* 98: 115-143.
25. Zemel E (1989) Easily computable facets of the knapsack polytope. *Mathematics of Operations Research* 14: 760-764.
26. Hammer P, Johnson E, Peled U (1975) Facets of regular 0-1 polytopes. *Mathematical Programming* 8: 179-206.
27. Padberg M (1973) On the facial structure of set packing polyhedral. *Mathematical Programming* 5: 199-215.
28. Wolsey LA (1975) Faces for a linear inequality in 0-1 variables. *Mathematical Programming* 8: 165-178.
29. Wolsey LA (1975b) Facets and strong valid inequalities for integer programs. *Operations Research* 24: 367-372.
30. Gu Z, Nemhauser GL, Savelsbergh MWP (2000) Sequence independent lifting in mixed integer programming. *Journal of Combinatorial Optimization* 4: 109-129.
31. Shebalov S, Klabjan D (2006) Sequence independent lifting for mixed integer programs with variable upper bounds. *Mathematical Programming* 105: 523-561.
32. Wolsey LA (1977) Valid inequalities and superadditivity of 0/1 integer programs. *Mathematics of Operations Research* 2: 66-77.
33. Santanu SD, Jean-Philippe PR (2006) Linear programming based lifting and its application to primal cutting plane algorithms. School of Industrial Engineering, Purdue University.
34. Zemel E (1978) Lifting the facets of 0-1 polytopes. *Mathematical Programming* 15: 268-277.
35. The CPLEX Solver on IBM's Home Page
36. Gutierrez T (2007) Sequential lifting and simultaneous up lifting of integer variables, Kansas State University, Manhattan, Kansas.