# On Generalized Branching Methods for Mixed Integer Programming

Sanjay Mehrotra and Zhifeng Li *

December 31, 2004

Department of Industrial Engineering and Management Sciences,
Robert R. McCormick School of Engineering,
Northwestern University, Evanston, Illinois 60208
{mehrotra, zhifeng}@iems.northwestern.edu

**Abstract**

In this paper we present a restructuring of the computations in Lenstra's methods for solving mixed integer linear programs. We show that the problem of finding a good branching hyperplane can be formulated on an adjoint lattice of the Kernel lattice of the equality constraints without requiring any dimension reduction. As a consequence the short lattice vector finding algorithms, such as Lenstra, Lenstra, Lovász (LLL) [15] or the generalized basis reduction algorithm of Lovász and Scarf [18] are described in the space of original variables. Based on these results we give a new natural heuristic way of generating branching hyperplanes, and discuss its relationship with recent reformulation techniques of Aardal and Lenstra [1]. We show that the reduced basis available at the root node has useful information on the branching hyperplanes for the generalized branch-and-bound tree. Based on these results algorithms are also given for solving mixed convex integer programs.

**Key Words:** Linear Programming, Volumetric Center, Analytic Center, Interior Point Methods, Convex Programming, Mixed Integer Programming, Lattice Basis Reduction.

*Both authors were supported by NSF grant DMI-0200151 and ONR grant N00014-01-1-0048

## 1. Introduction

Lenstra's result that the linear mixed integer programming problem is solvable in polynomial time in fixed number of variables is the most significant theoretical result for linear mixed integer programs [16, 25]. In order to achieve this result Lenstra constructed a novel algorithm for finding a feasible integer solution of a polyhedral set. The most important aspect of this algorithm is that it branches on hyperplanes, instead of the standard approach of branching on variables. The class of algorithms that are designed to allow branching on a general hyperplane (or half-space) are called generalized branch and bound (GBB) algorithms.

Lenstra [16] showed that we can either generate a feasible integer solution of a polytope, or it is possible to generate a direction (normal to the branching hyperplane) along which the width of the polytope is bounded only as a function of the dimension of the polytope. This is important because for the general integer programs when branching on a single variable the width of the branch-and-bound tree can be exponential in the size of problem data [29]. Lenstra's algorithm as described in Schrijver [25] has four basic steps: (i) Ellipsoidal rounding of the set; (ii) a lattice basis reduction in ellipsoidal norm; (iii) Feasibility check; (iv) dimension reduction of the set. Lattice basis reduction is at the core of the overall construction, and the algorithm of Lenstra, Lenstra, and Lovász (LLL algorithm) [15] (or other related algorithms, see Mehrotra and Li [19]) are used for this purpose. The iterations of LLL algorithm use matrix computations.

The algorithm by Lovász and Scarf for integer programming [18] (LS algorithm) is related to Lenstra's algorithm with the important difference that it does not require an ellipsoidal rounding of the feasible set. The lattice basis reduction is done using the generalized basis reduction (GBR) method also given in Lovász and Scarf [18]. Each iteration of the GBR algorithm requires solutions of mathematical programs to compute the generalized norm. Furthermore, the number of iterations required in the algorithm are not polynomial. These two properties of the algorithm make it potentially expensive.

Although Lenstra's theoretical result is seminal, his algorithm has found limited value in practice. In our present understanding there are three major reasons for this. First, the LLL algorithm or its variants are expensive to execute for two reasons: (i) during LLL we may generate large numbers requiring high (or arbitrary) precision; (ii) during LLL execution it is difficult to exploit sparsity of the original problem. Second, after a branching hyperplane is added in the algorithm a dimension reduction step is performed in the algorithm. This step is theoretically convenient, however, it destroys the sparsity structure of the original problem and the reduced problem may involve numbers much larger in size than those in the original problem. A third reason is that Lenstra's algorithm solves a feasibility problem, and to solve an optimization problem a sequence of feasibility problems are solved in a bisection method.

Historically there could also be a fourth reason for practical difficulty in implementing Lenstra's

original algorithm, namely the difficulty in finding an ellipsoidal rounding of the polyhedral set. However, since the development of interior point methods, we now know that a good ellipsoidal rounding or a polyhedral (or convex sets) can be computed efficiently.

This paper studies the mixed integer feasibility problems. It makes the following contributions:

- It shows that the problem of finding a good branching hyperplane can be formulated on an adjoint lattice of the kernel lattice of the equality constraints without requiring any dimension reduction. Adjoint lattices (at least in the context of integer programming) are introduced later in this paper. The data in the hyperplanes generated from an adjoint lattice is 'better behaved' compared to the data in problems obtained after dimension reduction or projections.

- It allows the possibility of finding ellipsoidal rounding in the space of original variables without a dimension reduction in the pure integer case, or a projection computation in the mixed integer case. Consequently, all center finding computations may benefit from the sparsity of original data. Furthermore, one may use practical log-barrier methods for the purpose of finding a center.

- It gives a relationship between reduced kernel lattice basis and its adjoint, hence computation of reduced kernel lattice basis is possible from the adjoint lattice basis and vice-versa. This relationship allows the possibility of working with kernel lattice basis as well. Methods are given that use the kernel lattice basis for rounding the center to a nearby integer solution. Based on these relationships this paper also establishes a worst case complexity results when working in the space of original variables.

- It describes the possibility of running LLL and GBR algorithms at selective nodes instead of all the nodes of a generalized branch-and-bound tree. Consequently, the burden of running LLL or GBR algorithms is significantly reduced. We discuss why a basis available at the root node may provide 'good' information for generating branching hyperplanes at all children nodes. In this context a relationship with the recent reformulation proposal of Aardal and Lenstra [1] and Aardal, et. al. [2] is also established. The LLL and GBR algorithms at any node have a natural description in the space of original variables.

- It gives a natural extension of branching hyperplane algorithms for pure and mixed convex integer programs.

A companion paper (Mehrotra, *et. al.* [20]) gives the computational results using an implementation based on the proposed scheme for solving mixed integer optimization problem.

We now review previous research on this topic. Cook, *et. al.* [9] implemented LS algorithm for solving mixed integer linear programming problems. Cook, *et. al.* [9] assumed full dimensionality of problems, and they transformed data in the original constraint matrix in order to record unimodular operations in GBR algorithm. Cook *et. al.* solved several difficult mixed

integer problems and found that the number of nodes in the branch and bound tree were significantly fewer than those present in the standard approach that branches on one variable at a time. Moreover, the overall computational performance was better on several difficult problems in comparison with the CPLEX mixed integer programming solver available at the time. Wang [29] presented a refinement of the algorithm implemented by Cook *et. al.*. In particular, he replaced the bisection search with a more refined approach, and solved several convex integer programs using the generalized basis reduction algorithm.

Surprisingly, since the publication of Cook, *et. al.* [9] and the thesis of Wang [29] not much is reported on these methods till very recently. Owen and Mehrotra [27] heuristically generated the branching disjunctions (where at each node only two branches are generated) at the optimal solution and reached conclusions very similar to those reported in the work of Cook *et. al.* [9]. The interesting aspect of the results in Owen and Mehrotra is that the hyperplanes are not generated from the minimum width hyperplane finding problem; instead they are generated from the desire to improve the lower bound on the objective value as much as possible. Gao and Zhang [10] reported their experience with implementing Lenstra's algorithm where ellipsoidal rounding was used for finding the branching hyperplane. They performed dimension reduction at root, and possibly at other nodes in the branch and bound tree, to maintain full dimensionality of the polyhedral set. The ellipsoidal rounding was obtained by computing a maximum volume ellipsoid approximating the polyhedral set using an interior point method.

For hard knapsack problems Aardal and Lenstra [1] proposed a reformulation technique, and showed that branching on single variables in the reformulated problem requires significantly fewer branches than those required to solve the original problem using CPLEX version 6.5.3. Their reformulation reduces the problem dimension. The LLL basis reduction method is used to obtain such as basis and feasible solution. Our developments suggest that such reformulations are not required. A similar reformulation technique was studied in Aardal, *et. al.* [2] while solving difficult market split problems.

This paper is organized as follows. In the next section we formally introduce the problem under consideration. We give notations, definitions, and assumptions used through out this paper. In Section 3 we review and critique Lenstra and Lovász-Scarf algorithms. In Appendix 10.1 we describe LLL basis reduction algorithm in the original space. A similar description of GBR algorithm is given in Appendix 10.2. The sections in the appendix are presented for completeness. Sections 4 and 5 consider the pure integer feasibility problem. In Section 4 we describe our modifications to Lenstra-type algorithms, and give a reformulation of the hyperplane finding problem. Rounding to a feasible integer solution is discussed in Section 4.3. In Section 5 we compare our approach on generating branching hyperplane with the work of Aardal and Lenstra [1]. Section 6 discusses the properties of a reduced basis at the root node, and shows why such a basis can already give a sequence of branching hyperplanes for generating a good quality branching tree. A generalization of the proposed approach to the mixed integer linear programs is given in

Section 7. Section 8 extends this development to mixed integer convex problems. Here we show that $\epsilon$-feasibility versions of these problems can be solved with a complexity similar to the linear case. The discussion on feasibility mixed integer convex programs assumes that the constraints are twice differentiable, and they admit a self-concordant barrier (see next section). Section 9 gives concluding remarks and future research.

## 2. Problems, Assumptions, Notations, and Definitions.

**Feasibility Integer Programs**

Below we give four problems that are of interest in this paper. The pure feasibility integer linear programming problem (FILP) is to

$$\text{Find } \{x \in \mathbb{Z}^n_+ \mid Ax = a\}, \tag{2.1}$$

where $A \in \mathbb{Z}^{m\times n}$, $a \in \mathbb{Z}^m$, and $A$ is assumed to have full row rank. If $A$ does not have a full row rank, then the linearly dependent rows of $A$ are deleted. The feasibility mixed integer linear problem (FMILP) is to

$$\text{find} \quad \left\{ x = \begin{bmatrix} x_z \\ x_c \end{bmatrix},\ x_z \in \mathbb{Z}^n_+,\ x_c \in \mathbb{R}^{\bar{n}}_+ \mid Rx = r \right\}, \tag{2.2}$$

where $R$ has the form $R = \begin{bmatrix} B : C \\ A : 0 \end{bmatrix}$, $r = \begin{bmatrix} b \\ a \end{bmatrix}$, $B \in \mathbb{Z}^{\bar{m}\times n}$, and $C \in \mathbb{Z}^{\bar{m}\times \bar{n}}$. The matrix $A$ is defined as before. The columns of $A$ correspond to the integer variables, and the columns of $C$ correspond to the real variables. Without loss of generality we assume that $C$ has full row rank. If this is not the case, then we have a $\pi$ such that $\pi^T C = 0$, allowing us to delete a constraint for which $\pi_i \neq 0$, and replace it with the constraint $\pi^T B = \pi^T b$. Our developments in this paper suggest that this transformation is needed only in the pre-processing stage while computing the initial adjoint lattice. All subsequent computations can be performed using the original data.

The feasibility mixed integer convex program (FMICP) is to

$$\text{find} \quad \{x_z \in \mathbb{Z}^n_+, x_c \in \mathbb{R}^{\bar{n}}_+ \mid Rx = r, c_i(x) \leq 0, i = 1, \ldots, l\}. \tag{2.3}$$

The structure and assumptions on $R$ are same as in (FMILP). The functions $c_i(x) : \mathbb{R}^n \to \mathbb{R}$ are convex. An $\epsilon$ feasibility version of these problems satisfies constraints $c_i(x) \leq 0$ by relaxing them to $c_i(x) \leq \epsilon$, $i = 1, \ldots, l$, for a given $\epsilon > 0$. These problems are denoted by $\epsilon$-FMICP.

**Continuous Relaxation, Bounded and Non-empty Interior Assumption**

Let $x = \begin{bmatrix} x_z \\ x_c \end{bmatrix}$. The set obtained from relaxing the integrality requirements on $x_z$ in (FMILP) and (FMICP) are represented by $\mathcal{P}$ and $\mathcal{C}$ respectively, i.e.,

$$\mathcal{P} := \{x \mid Rx = r, x \geq 0\},$$

and

$$\mathcal{C} := \{x \mid Rx = r, c_i(x) \leq 0, i = 1, \ldots, l\}.$$

The continuous relaxation of (FILP) is also represented by $\mathcal{P}$. We assume that the set $\mathcal{P}$ and more generally the set $\mathcal{C}$ is bounded with non-empty relative interior, i.e., the set

$$\mathcal{C}^0 := \{x \mid Rx = r, c_i(x) < 0, i = 1, \ldots, l\},$$

has a feasible solution. For a polyhedral set $\mathcal{P}$ verification of non-empty relative interior can be done in polynomial time. In fact, it is a by-product of an optimization problem that finds a feasible solution of $\mathcal{P}$ using interior point methods. This is because the entire set $\mathcal{P}$ is the optimal face for the feasibility/optimization problem, and interior point methods converge to a point in the interior of this face. All inequality constraints that are tight at this solution can be permanently set to equality constraints (see for example, Mehrotra and Ye [22], Ye [30]). The assumption that the feasible sets are bounded can be ensured by introducing an explicit constraint bounding all the variables.

Polynomial time solutions of general convex programs are possible only within a specified tolerance [24, 23]. This is our reason for introducing ($\epsilon$-FMICP).

**General Notation**

The superscript $^T$ represents transpose of a vector or a matrix. For $x \in \mathbb{R}^n$, $\|x\|$ represents the $l_2$ norm, and $\|x\|_Q$ represents the ellipsoidal norm $\sqrt{x^T Q x}$. In our developments $Q$ may be a positive semi-definite or definite matrix. $\nabla\phi(x), \nabla^2\phi(x)$ represent the gradient and Hessian of a multivariate function $\phi(\cdot)$ at $x$. $\lfloor\alpha\rceil$ represents an integer nearest to the real number $\alpha$. $\lfloor\alpha\rfloor$ denotes the integer number less or equal to $\alpha$, and $\lceil\alpha\rceil$ the number greater or equal to $\alpha$. $\lfloor x\rfloor$ and $\lceil x\rceil$ represent integral vectors obtained by rounding each component of a real vector $x$ as described above. $I$ represents an Identity matrix of an appropriate size. $e_i$ represents $i$-th column of an Identity matrix. We will use the symbol $\oplus$ to represent sum of sets, i.e., $\mathcal{X} \oplus \mathcal{Y} := \{x + y \,|\, x \in \mathcal{X}, y \in \mathcal{Y}\}$. $gcd(a)$ is the greatest common divisor of the components of $a$, and $a$ is called *primitive* if $gcd(a) = 1$.

**Width of a Convex Set**

The width of a convex set $\mathcal{C}$ along a vector $u$ is defined to be

$$\mathcal{W}(u, \mathcal{C}) := \max_{u \in \mathcal{C}} u^T x - \min_{u \in \mathcal{C}} u^T x.$$

In our case the vector $u$ is integral.

**Range and Null Spaces, Lattice, Kernel Lattice and its Dual**

The range space of a $m \times n$ matrix $A$, $\{x \in \mathbb{R}^n \mid x = \sum_{i=1}^m \mathbb{R} A^T e_i\}$, is represented by $\mathcal{R}(A)$. The null space of $A$ is given by $\mathcal{N}(A) := \{p \in \mathbb{R}^n \mid Ap = 0\}$. Given $B = [b_1, \ldots, b_k]$, $\mathcal{L}(B) := \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^k \mathbb{Z} b_i\}$, is the lattice generated by column vectors $b_i$, $i = 1, \ldots, k$. A lattice is called integral if all vectors in $\mathcal{L}(B)$ are integer vectors. An integral lattice has an associated

unique integral kernel lattice $\mathcal{K}(B) := \{u \in \mathbb{Z}^n \mid u^T b = 0 \text{ for all } b \in \mathcal{L}(B)\}$. The lattice $\mathcal{K}(A^T)$ is represented by $\Lambda$. The existence of $\Lambda$ is well known. The dual of $\Lambda$ is defined as the set

$$\Lambda^\perp := \{z \in \mathbb{R}^n | Az = 0, z^T x \in \mathbb{Z}, \text{ for all } x \in \Lambda\}.$$

The set $\Lambda^\perp$ is also a lattice, however, it may not be integral. If $Z$ is a basis for $\Lambda$, then $Z^\perp := Z(Z^T Z)^{-1}$ is a basis for $\Lambda^\perp$. This is shown as follows. Clearly all vectors of the form $v = Z(Z^T Z)^{-1} p$, where $p$ is integral satisfy $v^T x \in \mathbb{Z}$ for all $x \in \Lambda$. Now, take a $v$ such that $Av = 0$ and assume that $v^T x \in \mathbb{Z}$ for all $x \in \Lambda$. Since $Z$ is a basis of $\Lambda$, we have $Z^T v = u$, where $u$ is an integral vector. Also, since $Av = 0$, there exists a unique $y \in \mathbb{Z}^k$ such that $v = Zy$. Therefore, $Z^T Zy = u$, and $y = (Z^T Z)^{-1} u$, which gives $v = Z(Z^T Z)^{-1} u$. This also proves that $\Lambda^\perp$ is a lattice, and it is unique.

**Adjoint Lattice**
A lattice $\mathcal{K}^*(A^T)$ is called an adjoint lattice of $A$ if for any basis $Z$ of $\Lambda$ there exist a basis $Z^*$ of $\mathcal{K}^*(A^T)$ such that

$$Z^T Z^* = I. \tag{2.4}$$

An adjoint lattice is integral if all its elements are integral. Integral adjoint lattices play a fundamental role in the developments of this paper. Henceforth the prefix "integral" is dropped when considering integral adjoint lattices.

Obviously the dual lattice gives an adjoint lattice, however, it may not be integral. We now show the existence of an integral adjoint lattice, and give a way of computing it. Let $U$ be a unimodular matrix such that $AU = [H : 0]$, where $H$ is the Hermite normal form of $A$ (see Schrijver [25]). The last $k$ columns ($k = n - m$) of $U$ give a basis of $\Lambda$. We denote these columns of $U$ by $Z \in \mathbb{Z}^{n \times k}$. Let us represent the last $k$ columns of $U^{-T}$ by $Z^*$. The matrix $U^{-T}$ is integral because $U$ is unimodular, hence $Z^*$ is integral, and $Z^T Z^* = I$. We just showed the existence of an adjoint lattice corresponding to any basis of $\Lambda$ generated in a HNF calculation. The following proposition extends the existence result to any basis of $\Lambda$.

**Proposition 2.1** *Let $\tilde{Z}$ be a basis of the lattice $\Lambda$. Then there exists an integral matrix $\tilde{Z}^*$ such that*

$$\tilde{Z}^{*T} \tilde{Z} = I. \tag{2.5}$$

**Proof:** Assume that a basis $Z$ of $\Lambda$ is available from the HNF computations of $A$, and $Z^*$ is constructed as above. Since $Z$ and $\tilde{Z}$ are basis for $\Lambda$ we have a unimodular matrix $V$ such that $\tilde{Z} = ZV$. Now take $\tilde{Z}^* = Z^* V^{-T}$. Since $V$ is unimodular, $V^{-T}$ is also unimodular. □

Note that an integral adjoint lattice associated with $A$ is not unique. For example any matrix in the set $Z^* \oplus \mathcal{L}(A^T)$ satisfies (2.4), and the lattices generated are different. We represent an

instance of $\mathcal{K}^*(A^T)$ by $\Lambda^*$. $Z^*$ represents a basis of $\Lambda^*$ satisfying $Z^T Z^* = I$ for a basis $Z$ of $\Lambda$.

**Lenstra, Lenstra, and Lovász Reduced Basis**

We now define a LLL-reduced basis as used in this paper. This definition is adapted from [14, 15] for the ellipsoidal norm. Let $\hat{B} = [\hat{b}_1, \ldots, \hat{b}_n]$ be the orthogonal basis vectors computed by using the Gram-Schmidt orthogonalization procedure as follows:

$$\hat{b}_i = b_i - \sum_{j=1}^{i-1} \Gamma_{j,i} \hat{b}_j, i = 1, \ldots, n, \tag{2.6}$$

where

$$\Gamma_{j,i} = b_i^T E \hat{b}_j / \|\hat{b}_j\|_E^2, \text{ and } \hat{b}_1 = b_1. \tag{2.7}$$

**Definition 2.1** *A basis $b_1, \ldots, b_n$ of a lattice $\mathcal{L}$ is called a* LLL-reduced basis*, for $\delta \in (\frac{1}{4}, 1]$, if it has the following two properties:*

**C1. (Size Reduced)** $|\Gamma_{j,i}| \leq 1/2$ *for* $1 \leq j < i \leq n$.

**C2. (2-Reduced)** $\|\hat{b}_{i+1}\|_E^2 \geq (\delta - \Gamma_{i,i+1}^2)\|\hat{b}_i\|_E^2$, $|\Gamma_{i,i+1}| \leq 1/2$ *for* $i = 1, \ldots, n-1$.

It is assumed that $\|\cdot\|_E \neq 0$ for vectors of interest. A basis is called 2-reduced if only C2 is satisfied. It is called size-reduced if only C1 is satisfied. The upper triangular matrix $\Gamma := [\Gamma_{j,i}]$ satisfies $\Gamma^T D \Gamma = B^T E B$, where $D$ is a diagonal matrix whose elements $D_{ii}$ are $\|\hat{b}_i\|_E^2$. Note that $\Gamma$ can be computed from the Cholesky factors of $B^T E B$.

**Size Reduction of $B$.**

We now show that a size-reduced lattice basis is easily obtained from a 2−reduced lattice basis. Let us obtain basis $\tilde{b}_1, \ldots, \tilde{b}_n$ from the basis $b_1, \ldots, b_n$ as $b_1, \ldots, b_{k-1}, b_k - [\Gamma_{j,k}]b_j, \ldots, b_n$, $j < k$, i.e., $\tilde{B} = BU_{j,k}$, where $U_{k,j} = I - [\Gamma_{j,k}]e_j e_k^T$ is an elementary unimodular matrix. It is easy to see that $\tilde{B} = \hat{B}\tilde{\Gamma}$, where $\tilde{\Gamma} = \Gamma - [\Gamma_{j,k}]e_j e_k^T$. Note that $\hat{B}$ is unchanged as a result of this operation. This operation results in $|\tilde{\Gamma}_{j,k}| \leq 1/2$. This computation is called size reduction of $b_k$ against $b_j$, $j < k$. Note that $\tilde{\Gamma}$ is obtained from $\Gamma$ (i.e., $\Gamma$ is updated) in $O(n)$ arithmetic operations. After initial $\Gamma$ is computed, we can size reduce the entire basis by recursively applying this step in the order $(k, j) = (n, n-1), (n, n-2), \ldots, (n, 1), (n-1, n-2), \ldots, (2, 1)$. This is summarized in the method **SizeReduceBasis**. Note that generating a size reduced basis from a 2-reduced basis requires $O(n^3)$ arithmetic operations and only requires information from $\Gamma$.

**Lovász and Scarf [18] Reduced Basis**

Lovász and Scarf [18] developed a generalized basis reduction (GBR) algorithm which gives a reduced basis of the lattice $\mathbb{Z}^n$ with respect to a generalized distance function $F$ defined on $\mathcal{C}$: $F(x, \mathcal{C}) = \inf\{\lambda | \lambda \geq 0, x \in \lambda\mathcal{C}\}$. Let $\mathcal{C}^*$ be the dual of $\mathcal{C}$ defined as $\mathcal{C} := \{p | p^T x \leq 1 \text{ for all } x \in \mathcal{C}\}$. It can be shown that the generalized distance of a point $y$ to the dual set $\mathcal{C}^*$ is computed by

**Method: SizeReduceBasis** ($B$ (or $U$), $\Gamma$,$n$)

FOR $k = n, \ldots, 2$

FOR $j = k-1, \ldots, 1$

$\vartheta = [\Gamma_{k-1,j}]$;

$b_k = b_k - \vartheta b_j$;

FOR $i = 1, \ldots, j$

$\Gamma_{i,k} = \Gamma_{i,k} - \vartheta \Gamma_{i,j}$;

Figure 1: Size Reduction of a Basis

solving an optimization problem defined over $\mathcal{C}$. In particular, $F(y, \mathcal{C}^*) = \max_{x \in \mathcal{C}} y^T x$ [18]. Let us define

$$F_i(x, \mathcal{C}) = \min_{\alpha_1, \ldots, \alpha_{i-1} \in \mathbb{R}} F(x + \alpha_1 b^1 + \ldots + \alpha_{i-1} b^{i-1}, \mathcal{C}). \tag{2.8}$$

The function $F_i(x, \mathcal{C})$ is a distance function associated with the projection of $\mathcal{C}$ (call it $\mathcal{C}_i$) into the subspace generated by $\{b^i, \ldots, b^n\}$ along $\{b^1, \ldots, b^{i-1}\}$, i.e., $x = \sum_{l=i}^{n} \alpha_l b^l \in \mathcal{X}_i$ iff there exist $\alpha_i, i = 1, \ldots, i-1$ such that $x + \sum_{l=1}^{i-1} \alpha_l b^l \in \mathcal{C}$. If the constraints defining $\mathcal{C}$ are explicitly given, then the functions $F_i(x, \mathcal{C})$ can either be computed directly from (2.8) or from the solution of

$$F_i(x, \mathcal{C}) = \max\{x^T z | z \in \mathcal{C}^*, b_i^T z = 0, \ldots, b_{i-1}^T z = 0\}. \tag{2.9}$$

**Definition 2.2** *A basis $b_1, \ldots, b_n$ is called* Lovász-Scarf reduced basis *(LS-reduced for short) for a given $0 < \epsilon < \frac{1}{2}$ if the following two conditions hold for $i = 1, \ldots, n-1$:*

**(G1)** *$F_i(b_{i+1} + \mu b_i, \mathcal{C}) \geq F_i(b_{i+1}, \mathcal{C})$ for integral $\mu$,*

**(G2)** *$F_i(b_{i+1}, \mathcal{C}) \geq (1 - \epsilon) F_i(b_i, \mathcal{C})$.*

These conditions reduce to conditions C1 and C2 when $\mathcal{C}$ is replaced with an ellipsoid.

**Self-Concordant Convex Functions, Barriers, and Analytic Centers**
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable strictly convex function, and consider the set

$$\hat{\mathcal{C}} := \{x \mid c_i(x) \leq 0, i = 1, \ldots, l\},$$

where $c_i(\cdot)$ are convex functions as defined before. Let $\hat{\mathcal{C}}^0$ be the interior of $\hat{\mathcal{C}}$. Following Renegar [24] (see also Nesterov and Nemirovskii [23]) the function $f$ is called self-concordant ($SC$) if for all $x \in \hat{\mathcal{C}}$ and $y \in \{y \mid \|y - x\|_{\nabla^2 f(x)} \leq 1\}$, we have

$$1 - \|y - x\|_{\nabla^2 f(x)} \leq \frac{\|v\|_{\nabla^2 f(y)}}{\|v\|_{\nabla^2 f(x)}} \leq \frac{1}{1 - \|y - x\|_{\nabla^2 f(x)}},$$

for all $v \neq 0$. A self-concordant barrier associated with $\hat{\mathcal{C}}$ is a $SC$ function with the additional property

$$\theta := \sup_{x \in \hat{\mathcal{C}}^0} \|\nabla f(x)\|_{\nabla^2 f(x)} < \infty.$$

The parameter $\theta$ is called the complexity value of $f$. A restriction of a $SC$ barrier with complexity value $\theta$ on a subspace (or its translation) (see Renegar [24, Page 35]) is also a $SC$ barrier with complexity value $\theta$. Hence, without loss of generality, we will refer to $f$ as a barrier function over $\mathcal{C}$. The minimizer of a barrier function is called an analytic center associated with the barrier. The most popular barrier function is the log-barrier. The log-barrier analytic center for (FMILP, FILP) is given by the solution of

$$\max\{\rho(x, \mathcal{P}) := -\sum_{i=1}^{n} \ln x_i \mid Rx = r\}.$$

For (FMICP) under appropriate assumptions the analytic center is a solution of

$$\max\{\rho(x, \mathcal{C}) := -\sum_{i=1}^{n} \ln c_i(x) \mid Rx = r\}.$$

The log-barrier analytic center is well defined if the inequality constraints are given by convex functions and it is bounded with non-empty relative interior. The gradient and Hessian of the log-barrier at a point $x$ are given by:

$$\nabla\rho(x, \mathcal{P}) = -\left(1/x_i\right), \ \nabla^2\rho(x, \mathcal{P}) = \left[diag(1/x_i^2)\right],$$
$$\nabla\rho(x, \mathcal{C}) = -\sum_{i=1}^{l} \tfrac{1}{c_i(x)} \nabla c_i(x), \ \nabla^2\rho(x, \mathcal{P}) = \sum_{i=1}^{l} \left[\tfrac{1}{c_i^2(x)} \nabla c_i(x) \nabla c_i(x)^T + \tfrac{1}{c} \nabla^2 c_i(x)\right].$$

## 3. Branching on Hyperplane Algorithms

We now review Lenstra's algorithm. Our description here follows Schrijver [25]. This description is different from the one given in Lenstra [16] in Step 2. Lenstra performs a transformation of variables to apply LLL algorithm, while Schrijver [25] gives the basis reduction algorithm without performing variable transformation.

Lenstra assumes that the feasible set of integer linear program is full dimensional. He gives a method for achieving this. More recently Aardal and Lenstra [1] and Aardal *et. al.* [2] have given a particular reformulation scheme that also achieves this for knapsack and market split problems. In these schemes (FILP) is transformed to an equivalent feasibility problem:

$$\text{Find } \{y \in \mathbb{Z}^k \mid y \in \mathcal{Y} := Z^T y \geq v\}, \tag{3.1}$$

where $Z \in \mathbb{Z}^{n \times k}$ is any basis for $\mathcal{K}(A^T)$, and $v \in \mathbb{Z}^n$ is any integral solution satisfying $Ax = a$. Aardal and Lenstra [1] and Aardal *et. al.* [2] give a scheme for computing a suitable $Z$ for knapsack and market split problem instances.

Lenstra's algorithm performs four basic steps at each node of the branch-and-bound tree it generates. It suffices to describe the algorithm at any node (say the root node). The steps are:

Step 1 **Ellipsoidal Rounding**
Find a positive definite matrix $Q \in \mathbb{Q}^{k\times k}$ and a point $w \in \mathcal{Y}$ so that the ellipsoidal approximation of $\mathcal{Y}$: $\mathcal{E}(w,Q) := \{\|y-w\|_Q \leq 1\}$ satisfies,

$$\mathcal{E}(w,Q) \subseteq \mathcal{Y} \subseteq \mathcal{E}(w,Q/\gamma).$$

This ellipsoidal approximation is called the $\gamma$-approximation of $\mathcal{Y}$. The parameter $\gamma > 1$ represents the quality of approximation. A larger value means worse approximation quality.

Step 2 **Basis Reduction**
Given a $Q$ describing the ellipsoidal rounding, find a reduced basis $b_1, \ldots, b_k$ of the lattice $\mathbb{Z}^k$ in $\|\cdot\|_Q$ norm. This is done by Lenstra, Lenstra, and Lovász [15] basis reduction algorithm.

Step 3 **Feasibility Check and Branching Hyperplane Computations**
Let $b_1, \ldots, b_k$ be the basis vectors given in Step 2. Write $w = \sum_{i=1}^{k} \beta_i b_i$ and compute $\hat{y} = \sum_{i=1}^{k} \lfloor \beta_i \rceil b_i$. If $\hat{y} \in \mathcal{Y}$ we have found a feasible integer solution at the current node. Lifting $\hat{y}$ to the space of the original problem will give the desired feasible solution. Otherwise, take a primitive vector $p$ satisfying $p^T b_i = 0$ for $i = 1, \ldots, k-1$.

Step 4 **Branching and Dimension Reduction**
Add hyperplanes $p^T y = \alpha$, $\alpha \in \mathbb{Z}$, $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$, $\alpha_{\min} = \lceil \min_{y\in\mathcal{Y}} p^T y \rceil$, and $\alpha_{\max} = \lfloor \max_{y\in\mathcal{Y}} p^T y \rfloor$. The number of variables in the resulting problems is reduced by one or more to maintain full dimensionality. Reconstruct the feasibility problems in the lower dimension and add these nodes to the branch-and-bound tree. Return to Step 1, with an unexplored node of the branch-and-bound tree.

The existence of a $k-$approximation ellipsoid for $\mathcal{Y}$ (in fact a full dimensional convex set) is known since John [11]. John [11] showed that a minimum volume circumscribing ellipsoid gives a $k-$approximation of a convex set. However, to our knowledge no algorithm is known to find the minimum volume circumscribing ellipsoid. It is now known that the maximum volume inscribed ellipsoid can also provide a $k-$approximation of $\mathcal{Y}$. Polynomial time algorithms are known for finding a maximum volume inscribed ellipsoid (Tarasov, et al. [26], Nesterov and Nemirovskii [23], Khachiyan and Todd [13], Anstreicher [5]). We refer the reader to Anstreicher [5] for a discussion and further references on these methods and their computational complexity. Lenstra [16] also gave an algorithm for finding a $k-$approximation of a polyhedral set.

Two alternative notions of centers have also emerged with the development of interior point methods. These are the log-barrier center mentioned in Section 2, and the Vaidya-volumetric center introduced by Vaidya [28]. Suitable approximations of both centers are computable in polynomial time. However, the log-barrier center is more practical and it is a fundamental element of log-barrier interior point methods in optimization [23, 24]. The Vaidya center is more

suitable for non-differentiable convex functions or situations where log-barrier center can not produce a provably good quality approximation of the convex set. In this paper we will emphasize the use of log-barrier center. A companion paper [21] uses the Vaidya center for non-differentiable integer programming convex programming problems.

For the choice of $p$ constructed in Steps 2 and 3, Lenstra showed that $\mathcal{W}(p, \mathcal{Y})$ is bounded by $2n(n+1)2^{n(n-1)/4}$. This bound was improved by Babai [6] (see also Lovász [17]) to $2(n+1)\sqrt{n}(3/\sqrt{2})^n$. This bound shows that the number of hyperplanes added in Step 4 are constant when $n$ is fixed, and they are independent of the size of the problem data. The branching in Lenstra's algorithm is called branching on hyperplanes. This branching is different from standard branching, which branches on one variable ($p$ is $e_i$) at a time.

Hence, using the fact that the dimensionality of (3.1) is reduced by one after a hyperplane is added, and the data of the reduced problem remains polynomially bounded in the size of original data, ellipsoidal approximation is achieved in polynomial time, and LLL algorithm runs in polynomial time, Lenstra showed that (FILP) is solved in polynomial time for a fixed number of variables. Lenstra extended his algorithm for (FMILP) by doing a variable projection that projects out continuous variables from the problem and converts it into (FILP).

The algorithm of Lovász and Scarf [18] combines Steps 1 and 2 in Lenstra's algorithm. A vector from a LS-reduced basis is used to describe the branching hyperplane while full dimensionality of the original set is assumed. Our goal is to work in the space of original variables without performing an explicit dimension reduction or data transformation.

## 4. The Branching Hyperplane Problem in the Original Space

In this section we reformulate the branching hyperplane finding problem in generalized branching algorithms as a shortest lattice vector problem in the original space. For simplicity we first describe our ideas for the linear case with pure integer variables. The extension to general mixed integer problems is addressed in Section 7.

### 4.1 Ellipsoidal Approximation

Let $\mathcal{E}(w, Q) := \{x \mid (x-w)^T Q(x-w) \leq 1, Ax = a\}$. If $w$ is taken as the log-barrier analytic center, then it is well known that $\mathcal{E}(w, Q)$ gives a $n$-approximation of $\mathcal{P}$ [24]. In particular,

$$\mathcal{E}(w, \nabla^2\rho(w, \mathcal{P})) \subseteq \mathcal{P} \subseteq \mathcal{E}(w, \nabla^2\rho(w, \mathcal{P})/n).$$

Note that this is worse than the $k$-approximation possible if the problem dimension is reduced and the methods of Lenstra [16], Vaidya [28], and Anstriecher [3, 4] are used. It is not necessary to compute the log-barrier analytic center exactly to obtain a good rounding. This is discussed in Section 8 in the context of mixed integer convex feasibility problems.

### 4.2 The Branching Hyperplane Problem

Let us consider (FILP) available at a node of the branch-and-bound tree. Without loss of generality assume that $A \in \mathbb{Z}^{m\times n}$ includes all the hyperplanes added thus far in the branching process. The following theorem shows that the width of projected set $\mathcal{P}$ along an integral vector is equal to the width of $\mathcal{P}$ along a vector in an adjoint lattice of $A$. The opposite is also true.

**Theorem 4.1** *Let $u \in \Lambda^*$, and $u \neq 0$, then there exists a $p \in \mathbb{Z}^k$ ($p \neq 0$) such that*

$$\mathcal{W}(u, \mathcal{P}) = \mathcal{W}(p, \mathcal{Y}). \tag{4.1}$$

*Furthermore, for $p \in \mathbb{Z}^k$ there exists a $u \in \Lambda^*$ such that (4.1) also holds. In particular, $u = Z^*p$, where $Z^*$ is a basis of $\Lambda^*$ satisfying $Z^T Z^* = I$.*

**Proof:** Since $u \in \Lambda^*$, we have a $p$ such that $u = Z^*p$. We show that $p$ is the desired vector. Assume that $\max_{x\in\mathcal{P}} u^T x$ is attained at $x_{\max}$. Similarly, let $\min_{x\in\mathcal{P}} u^T x$ be attained at $x_{\min}$. Clearly we have a $\hat{y} \in \mathcal{Y}$, such that $x_{\max} = Z\hat{y} + v$, since $A(x_{\max} - v) = 0$ for an integral $v$ satisfying $Av = a$, and $Z$ is a basis for $\mathcal{N}(A)$. Now $u^T x_{\max} = p^T Z^{*T}(Z\hat{y} + v) = p^T\hat{y} + p^T Z^{*T} v$. Similarly, we have a $\bar{y} \in \mathcal{Y}$ such that $u^T x_{\min} = p^T\bar{y} + p^T Z^{*T} v$. Hence, $\mathcal{W}(u, \mathcal{P}) \leq \mathcal{W}(p, \mathcal{Y})$. Now assume that $\max_{y\in\mathcal{Y}} p^T y$ ($\min_{y\in\mathcal{Y}} p^T y$) is attained at $y_{\max}$ ($y_{\min}$), and construct $\hat{x} = Zy_{\max} + v$ ($\bar{x} = Zy_{\min} + v$). Clearly, $\hat{x}, \bar{x} \in \mathcal{P}$, and $u^T\hat{x} = (Z^*p)^T\hat{x} = p^T Z^{*T} y_{\max} + p^T Z^{*T} v$, and $u^T\bar{x} = (Z^*p)^T\bar{x} = p^T Z^{*T} y_{\min} + p^T Z^{*T} v$. Hence, we have $\mathcal{W}(u, \mathcal{P}) \geq \mathcal{W}(p, \mathcal{Y})$. This completes the proof. □

This immediately results in the following corollary.

**Corollary 4.1** $\min_{u\in\Lambda^*\setminus 0} \mathcal{W}(u, \mathcal{P}) = \min_{p\in\mathbb{Z}^k\setminus 0} \mathcal{W}(p, \mathcal{Y})$.

Note that Theorem 4.1 and Corollary 4.1 hold for any adjoint lattice, however integral adjoint lattices are of importance because they allow us to add constraints defined using integral vectors to an integral matrix $A$. Corollary 4.1 formulates the branching hyperplane finding problem over $\Lambda^*$. We now formulate the branching hyperplane problem when $\mathcal{P}$ is approximated with an ellipsoid.

The branching hyperplane finding problem for the ellipsoid $\mathcal{E}(w, Q)$ is to solve the minimization problem:

$$\min_{u\in\Lambda^*\setminus 0} \mathcal{W}(u, \mathcal{E}(w, Q)), \quad \text{or equivalently,} \quad \min_{u\in\Lambda^*\setminus 0} \mathcal{W}(u, \mathcal{E}(0, Q)), \tag{4.2}$$

where $\mathcal{E}(0, Q) = \{x \in \mathbb{R}^n \,|\, \|x\|_Q \leq 1, Ax = 0\}$. Since for any $u \in \mathbb{R}^n$, $\min_{x\in\mathcal{E}(0,Q)} u^T x = -\max_{x\in\mathcal{E}(0,Q)} u^T x$, we have the following result.

**Proposition 4.1** *The width of the ellipsoid $\mathcal{E}(0,Q)$ along $u \in \mathbb{Z}^n$ is*

$$\mathcal{W}(u, \mathcal{E}(0,Q)) = 2\|Q^{-1/2}u\|_{P_{AQ^{-1/2}}}, \tag{4.3}$$

where $P_{AQ^{-1/2}} = I - Q^{-1/2}A^T(AQ^{-1}A^T)^{-1}AQ^{-1/2}$, *or* $P_{AQ^{-1/2}} = Q^{1/2}Z(Z^TQZ)^{-1}Z^TQ^{1/2}$

*is an orthogonal projection matrix. In particular, if $u \in \Lambda^*$, then*

$$\frac{1}{2}\min_{u\in\Lambda^*\setminus 0} \mathcal{W}(u, \mathcal{E}(0,Q)) = \min_{p\in\mathbb{Z}^k\setminus 0} \|Q^{-1/2}Z^*p\|_{P_{AQ^{-1/2}}} \tag{4.4}$$

$$= \min_{p\in\mathbb{Z}^k\setminus 0} \sqrt{p^T(Z^TQZ)^{-1}p} \tag{4.5}$$

$$= \min_{p\in\mathbb{Z}^k\setminus 0} \|Q^{1/2}Z(Z^TQZ)^{-1}p\|. \tag{4.6}$$

A proof of Proposition 4.1 is easily constructed under the variable transformation $\hat{x} := Q^{-1/2}x$. The matrix $P_{AQ^{-1/2}}$ is an orthogonal projection matrix, and (4.3) computes the $l_2$ norm of vector $Q^{-1/2}u$ projected orthogonally on $\mathcal{N}(AQ^{-1/2})$. Equations (4.4–4.6) give us three related ways of computing the minimum width direction. The equality (4.4) allows us to formulate the problem of finding the branching hyperplane in the original space using an adjoint lattice basis $Z^*$. Equality (4.5) gives a formulation of the same problem using a basis $Z$ of $\Lambda$. Equality (4.6) gives a formulation of the problem over the dual lattice $\Lambda^\perp$. Equality (4.6) also shows (by taking $p = e_i$) that a basis of $\Lambda^\perp$ is obtained from a basis of an adjoint lattice via an orthogonal projection. Recall that the dual lattice is unique, where as we can have many different integral adjoint lattices.

Adjoint lattice is the most natural lattice to generate branching hyperplanes in the original space. For a given $\Lambda^*$, since $\Lambda^* \oplus \mathcal{L}(A^T)$ is also an adjoint lattice, they allow one to reduce the elements of $\Lambda^*$ prior to performing any computation. This is numerically beneficial particularly when we add branching hyperplanes to existing constraints. Lenstra [16] generates a branching hyperplane as a byproduct of a rounding calculation after finding the ellipsoidal rounding and an LLL-reduced basis. The above discussion has formulated this problem directly. It is possible that one finds a branching hyperplane with 'large' width after solving this problem. In the following discussion we develop the rounding procedure in our setting. We also show that if the width of $\mathcal{P}$ along the first vector of an appropriately reduced $Z^*$ exceeds a threshold, then we should be able to round the currently available center to a feasible integer solution.

### 4.3 Rounding to a Feasible Solution

We now recast the problem of rounding the computed center in Lenstra's algorithm in the space of original variables. Let $w$ be a center of $\mathcal{P}$ (or more generally $\mathcal{C}$). Let $v$ be any integral solution satisfying $Av = a$. Now consider the closest lattice vector (CLV) problem:

$$\min_{q\in\Lambda\setminus 0} \|w - v - q\|_Q \tag{4.7}$$

If the solution of (4.7) is feasible, we are done; otherwise, (as shown below) we must have a $u \in \Lambda^*$ along which the ellipsoid $\mathcal{E}(w, Q)$ (and by extension the set $\mathcal{P}$ or $\mathcal{C}$) is thin. Although Lenstra does not directly consider (4.7) his main result uses information from an approximate solution of (4.7). We show this below in our context, where we also establish the connection between reduced bases of $\Lambda$ and $\Lambda^*$.

Let $Z_1, \ldots, Z_k$ be a LLL-reduced basis of $\Lambda$ under the ellipsoidal norm $\|\cdot\|_Q$, i.e., $Q^{1/2}Z$ is LLL-reduced under $l_2$-norm. Since $w - v \in \mathcal{N}(A)$ and $Z$ is a basis for $\mathcal{N}(A)$, we have $\zeta_i \in \mathbb{R}, i = 1, \ldots, k$, such that $w - v = \sum_{i=1}^{k} \zeta_i Z_i$. Now generate a vector $\tilde{v} = \sum_{i=1}^{k} \lfloor \zeta_i \rceil Z_i$, and take $\bar{v} = v + \tilde{v}$ as a candidate solution. Clearly, $A\bar{v} = a$. If $\bar{v} \geq 0$, then we have a feasible solution; Otherwise, $(w - \bar{v})^T Q (w - \bar{v}) > 1$. Obviously $\tilde{v}$ is an approximate solution to the closest lattice vector problem, however, it is polynomially computable.

The next theorem is an analogue of the result of Babai [6] for our current setting. It shows that if the above discussed rounding procedure does not generate a feasible solution, then the ellipsoid $\mathcal{E}(0, Q)$ is thin along an element of $\Lambda^*$.

**Theorem 4.2** *Let $Q^{1/2}Z$ be LLL-reduced under $l_2$-norm. Let $\bar{v}$ constructed as above satisfy $(w - \bar{v})^T Q (w - \bar{v}) > 1$. Then there exists a branching direction $u \in \Lambda^*$ such that*

$$\mathcal{W}(u, \mathcal{P}) \leq 2\gamma (3/\sqrt{2})^k, \tag{4.8}$$

*where $\gamma$ satisfies $\mathcal{E}(w, Q) \subseteq \mathcal{P} \subseteq \mathcal{E}(w, Q/\gamma)$. In particular, one such $u$ is given by $Z^* e_j$, where $Z^T Z^* = I$ is a basis of $\Lambda^*$, and $j = argmin\{\|Q^{-1/2} Z_j^*\|_{P_{AQ^{-1/2}}}\}$.*

The motivation for the particular choice of $u$ can be seen from the following Lemma. This Lemma establishes a connection between a 2-reduced basis of $\Lambda$ and a 2-reduced basis of $\Lambda^*$. This Lemma shows that a 2-reduced 'projected' $Z^*$ is obtained from a 2−reduced $Z$, and vice versa. If a LLL-reduced $Z^*$ (or $Z$) is needed, then a 'size reduction step' is performed to convert a 2−reduced basis to a LLL-reduced basis. We can perform this size reduction inductively as discussed in Section 2.

**Lemma 4.1** *Let $Z$,$Z^*$ be bases of $\Lambda$ and $\Lambda^*$ respectively satisfying $Z^T Z^* = I$. Let $\bar{Z} := Q^{1/2} Z$ be 2-reduced (or $Z$ is LLL-reduced under $\|.\|_Q$ norm). Then, the column vectors of $\bar{Z}^* := P_{AQ^{-1/2}} Q^{-1/2} Z^*$ are 2-reduced in the reverse order. Also, if $P_{AQ^{-1/2}} Q^{-1/2} Z^*$ are 2-reduced, and $Z$ is constructed such that $Z^T Z^* = I$, then $Q^{1/2} Z$ is 2-reduced in the reverse order.*

**Proof**: Let $\bar{Z} = \hat{Z}\Gamma$, where $\hat{Z}$ is an orthogonal matrix obtained from the Gram-schmidt orthogonalization of $Q^{1/2}Z$, and $\Gamma$ be an upper triangular matrix, whose diagonal elements are 1.

Then,

$$\bar{Z}^{*T}\bar{Z}^* = Z^{*T}Q^{-1/2}[Q^{1/2}Z(Z^TQZ)^{-1}Z^TQ^{1/2}]Q^{-1/2}Z^* = (Z^TQZ)^{-1} = (\Gamma^T D\Gamma)^{-1},$$

where $D = diag\{\|\hat{Z}_1\|^2, \ldots, \|\hat{Z}_k\|^2\}$ is a diagonal matrix. Hence, $\bar{Z}^*\Gamma^T = \hat{Z}^*$ is the Gram-Schmidt orthogonal basis of $\bar{Z}^*$ calculated in the reverse order. Clearly,

$$\hat{Z}^T\hat{Z}^* = \Gamma^{-T}Z^TQ^{1/2}[Q^{1/2}Z(Z^TQZ)^{-1}Z^TQ^{1/2}]Q^{-1/2}Z^*\Gamma^T = I,$$

and $\|\hat{Z}_i\|\,\|\hat{Z}_i^*\| = 1$ for $i = 1, \ldots, k$. Since $\bar{Z}$ is LLL-reduced, by the definition of LLL-reduced basis, we have $\|\hat{Z}_{i+1}\| \geq (\delta - \Gamma_{i,i+1}^2)\|\hat{Z}_i\|^2$. Hence, $\|\hat{Z}_i^*\|^2 \geq (\delta - \Gamma_{i,i+1}^2)\|\hat{Z}_{i+1}^*\|^2$. Also $|(\Gamma^{-T})_{i+1,i}| \leq 1/2$, $i = 1, \ldots k-1$ because $(\Gamma^{-T})_{i+1,i} = -\Gamma_{i,i+1}$. The proof for the second part of the statement is similar. □

**Proof of Theorem 4.2**. We follow the notations in the proof of Lemma 4.1. Let $\zeta_i - \lfloor \zeta_i \rceil = \nu_i$ and $\nu$ be a vector whose $i$-th component is $\nu_i$, then $w - \bar{v} = Z\nu$. Now,

$$1 \leq \|Z\nu\|_Q^2 = \|\bar{Z}\nu\|^2 \leq \sum_{i=1}^{k} |\nu_i|\|\bar{Z}_i\| \leq \frac{1}{2\min_i \|\bar{Z}_i^*\|} \sum_{i=1}^{k} \|\bar{Z}_i^*\|\,\|\bar{Z}_i\|.$$

Hence, in order to prove the theorem it is sufficient to bound $\sum_{i=1}^{k} \|\bar{Z}_i^*\|\,\|\bar{Z}_i\|$. From $\bar{Z}^* = \hat{Z}^*\Gamma^{-T}$, we have $\bar{Z}_i^* = \sum_{j=i}^{k} \Gamma_{ij}^{-1}\hat{Z}_j^*$, hence

$$\|\bar{Z}_i^*\|^2 \leq \sum_{j=i}^{k} (\Gamma_{ij}^{-1})^2 \|\hat{Z}_j^*\|^2 = \sum_{j=i}^{k} (\Gamma_{ij}^{-1})^2 \frac{1}{\|\hat{Z}_j\|^2}$$

where the last equality follows from using $\|\hat{Z}_i\|\,\|\hat{Z}_i^*\| = 1$. Now for a LLL reduced basis $barZ$ we have (see Lenstra, Lenstra, and Lováz [15])

$$\frac{\|\hat{Z}_i\|}{\|\hat{Z}_j\|} \leq 2^{(j-i)/2}, \text{ and } \|\bar{Z}_i\| \leq 2^{(i-1)/2}\|\hat{Z}_i\|, \tag{4.9}$$

hence

$$\|\bar{Z}_i^*\|^2\,\|\hat{Z}_i\|^2 \leq \sum_{j=i}^{k} (\Gamma^{-T})_{ij}^2 \|\hat{Z}_i\|^2/\|\hat{Z}_j\|^2 \leq \sum_{j=i}^{k} 2^{(j-i)}(\Gamma_{ij}^{-1})^2 \leq 2^{k-i}\sum_{j=i}^{k}(\Gamma_{ij}^{-1})^2.$$

It follows from (4.9) that

$$\sum_{i=1}^{k} \|\bar{Z}_i^*\|\,\|\bar{Z}_i\| \leq \sum_{i=1}^{k} 2^{\frac{i-1}{2}} \|\bar{Z}_i^*\|\,\|\hat{Z}_i\| \leq \sum_{i=1}^{k} 2^{\frac{i-1}{2}} \left(\sum_{j=i}^{k} 2^{(j-i)}(\Gamma_{ij}^{-1})^2\right)^{1/2} \leq 2^{\frac{k-1}{2}}(\frac{3}{2})^k \leq (\frac{3}{\sqrt{2}})^k,$$

where we have used that $\sum_{i=1}^{k}\sum_{j=i}^{k}(\Gamma_{ij}^{-1})^2 \leq (3/2)^{2k}$. This completes the proof. □

The following corollary is an immediate consequence of Lemma 4.1 and Theorem 4.2.

**Corollary 4.2** *If the column vectors of $P_{AQ^{-1/2}}Q^{-1/2}Z^*$ are 2-reduced (an LLL-reduced basis is obviously 2-reduced) and a vector of $Z^*$ does not satisfy (4.8), then we can use the kernel lattice basis $Z$ corresponding to LLL-reduced $Z^*$ to generate a feasible integer solution of (FILP).*

Using (4.6) Lemma 4.1 also shows that a LLL-reduced $Z$ gives a 2-reduced $Z^\perp$. The branching direction $u \in \mathbb{Z}^n$ can also be constructed from the dual lattice $\Lambda^\perp$ with the width properties similar to those for the branching direction for the projected problem. In particular, it can be shown that

$$\min_{u \in \Lambda^\perp \setminus 0} \|Q^{-1/2}u\|_{P_{AQ^{-1/2}}} = \min_{u \in \Lambda^* \setminus 0} \|Q^{-1/2}u\|_{P_{AQ^{-1/2}}},$$

and $u^T x = \mathbb{Z} + \alpha$ for all $x \in \mathbb{Z}^n$, and $Ax = a$. Here $\alpha = u^T(Z^TQZ)^{-1}Z^Tv$ is a constant, and $v$ is any solution satisfying $Av = a$.

The worst case value of $\gamma$ in Theorem 4.2 depends on the choice of method for rounding $\mathcal{P}$ and computing the center. As discussed earlier the value of $\gamma = n$ when analytic center is used. Hence, for this more practical choice of center and working in the original space we pay a slight penalty by achieving an inferior bound on the worst case number of branching hyperplanes. We note that although LLL-reduced basis (either for $Z$ or $Z^*$) provides a convenient tool for establishing the theoretical bound on the number of branching hyperplanes, they are not essential to the implementation of the algorithm. Both the branching hyperplane finding problem and the CLV problem can be considered independently, and alternative algorithms can be used to solve them exactly or approximately.

## 5. Aardal and Lenstra's Reformulation

Recently Aardal and Lenstra [1] proposed the following method to solve some difficult instances of integer knapsack problems. First they compute a reduced kernel basis $Z$ of $\Lambda$ by using a variant of Lenstra, Lenstra, and Lovász basis reduction algorithm with $l_2$ norm and reduce the problem (2.1) to the full dimensional problem (3.1). Subsequently they branch on the coordinate $e_k, \ldots, e_1$. In the following we show that such a problem reformulation is not required. We now show that the Aardal-Lenstra branching directions are available in the original space by considering the adjoint lattice $\Lambda^*$.

**Corollary 5.1** *Consider the polyhedra $\mathcal{Y} = \{y \mid Z^Ty \geq v\}$, where $Z$ is a basis for $\Lambda$, and $v \in \mathbb{Z}^n$ satisfies $Av = a$. Then,*

$$\mathcal{W}(e_k, \mathcal{Y}) = \mathcal{W}(Z_k^*, \mathcal{P}),$$

*where $Z_k^* = Z^*e_k$ is the $k-$th column of $Z^*$.*

**Proof**: The proof follows immediately from Theorem 4.1 by taking $p = e_k$. □

If $Z$ is LLL-reduced in $l_2$ norm, then from Proposition 4.1 the last vector of $P_A Z^*$ for the corresponding adjoint lattice is the first vector of a 2-reduced basis, where $P_A = I - A^T(AA^T)^{-1}A$. Since the size reduction step does not change the order of a 2-reduced basis, the last vector of $Z^*$ corresponding to a LLL-reduced basis $Z$ gives a good approximation for the problem

$$\min_{u \in \Lambda^* \setminus 0} \|u\|_{P_A}.$$

We reach the following conclusion.

**Lemma 5.1** *At the root node the method in Aardal and Lenstra [1] finds a thin direction for an ellipsoid defined by taking $Q = I$ in the ellipsoidal approximation step of Lenstra's algorithm.*

Note that Aardal and Lenstra [1] also include a solution $v$ in there LLL-reduction procedure. Consequently, their final basis may not be the LLL-reduced basis mentioned above, however, one would get the same basis if the solution is removed from their LLL procedure. We have ignored this while writing Lemma 5.1. A comparison of our computational results in Mehrotra *et. al.* [20] with those reported by Aardal and Lenstra [1] shows similar number of nodes while solving the knapsack or market split problems [2]. In view of our discussion in the previous sections, we can work with the problem defined using $Z^*$ instead of working with the reformulated problem. The next proposition shows that the kernel lattice basis and its adjoint are trivially available after a branching hyperplane is added.

**Proposition 5.1** *Let $Z, Z^*$ be bases of $\Lambda$ and $\Lambda^*$ satisfying $Z^T Z^* = I$. Let us consider the problem obtained by adding a constraint $u^T x = \alpha$ to the set $Ax = b$, where $u = Z_k^*$. Then $\tilde{Z} = [Z_1, \ldots, Z_{k-1}]$ is an integral basis for $\mathcal{K}(\tilde{A})$ where $\tilde{A} := \begin{bmatrix} A \\ u^T \end{bmatrix}$. Furthermore, $\tilde{Z}^* = \left[Z_1^*, \ldots, Z_{k-1}^*\right]$ satisfies $\tilde{Z}^T \tilde{Z}^* = I$.*

**Proof**: Consider the Hermite normal form of $A$, i.e., $A\hat{U} = [H : 0]$. Let $\hat{U} = [U_A : \hat{Z}]$, where $AU_A = H$, and $A\hat{Z} = 0$. Since $\hat{Z}$ is basis of $\Lambda$, $Z = \hat{Z}\hat{U}$. Let $U = [U_A : Z]$. Clearly, $A[U_A : Z] = [H : 0]$. Now, $\begin{bmatrix} A \\ u^T \end{bmatrix} U = \begin{bmatrix} H : 0 : 0 \\ u^T U_A : 1 : 0 \end{bmatrix}$, where in $U$ we have permuted the columns of $Z$, to make $Z_k$ its first column. Hence, the matrix $U$ also puts $\tilde{A}$ into its Hermite normal form, therefore $\tilde{Z}$ is a basis for $\mathcal{K}(\tilde{A})$. Clearly, $\tilde{Z}^T \tilde{Z}^* = I$. □

From Proposition 5.1 and Corollary 5.1 we conclude the following result.

**Theorem 5.1** *Branching on the coordinates $e_k, \ldots, e_1$ of the problem $Zy \geq v$, is equivalent to branching on the vectors $Z_k^*, \ldots, Z_1^*$ of the adjoint lattice basis satisfying $Z^T Z^* = I$.* □

The discussion of this section suggests the possibility of using the LLL-reduced basis for an ellipsoidal approximation (use of $l_2$-norm is a further approximation) or the GBR-reduced basis available at the root node to generate branching hyperplanes at all nodes of the generalized branch-and-bound tree. We formalize the properties of using the basis from the root node for the entire branch-and-bound tree in the next section.

## 6. Reduced Basis at the Root Node

In this section we study the GBR and LLL basis reduction algorithms at the root node. In particular, we show that the basis available at the root node approximately minimizes the total number of nodes generated in the generalized branch-and-bound algorithm suggesting that the root node basis is a good quality basis. This discussion also opens the possibility of general implementations where the reduced basis is computed at the root node and only selectively at subsequent nodes of the branch-and-bound tree. This may significantly reduce the burden of running these algorithms in a practical implementation.

**Proposition 6.1** *Let $Z^* = [b_1, \ldots, b_k]$ be an adjoint lattice basis, and assume that the branch-and-bound tree is generated by branching in the order $b_1, \ldots, b_k$. The total number of nodes in the generalized branch-and-bound tree is bounded by*

$$\tilde{\Delta} := \lfloor (F_1(b_1) + 1) \rfloor + \lfloor (F_1(b_1) + 1) \rfloor \lfloor (F_2(b_2) + 1) \rfloor + \ldots + \prod_{i=1}^{k} \lfloor (F_i(b_i) + 1) \rfloor. \tag{6.1}$$

**Proof:** It is sufficient to show that the number of nodes at $i$-th level of the generalized branch-and-bound tree is bounded by $\prod_{j=1}^{i} (\lfloor F_j(b_j) \rfloor + 1)$ for $i = 1, \ldots, k$. First, it is easy to see that the number of nodes at root node is bounded by $\lfloor F_1(b_1) \rfloor + 1$. We prove that the number of nodes at its each children node is bounded by $(\lfloor F_1(b_1) \rfloor + 1)(\lfloor F_2(b_2) \rfloor + 1)$. The children nodes are generated by adding constraints $b_1^T x = \mu$ for all feasible values of $\mu$. For example, one subproblem is written as $\{\min c^T x \mid x \in \mathcal{C}, b_1^T x = \mu\}$ for some value of $\mu$. The number of nodes for this subproblem if branching on $b_2$ is bounded by $\lfloor G(b_2) \rfloor + 1$, where

$$G(b_2) = \{\max b_2^T (x - y) \mid x \in \mathcal{C}, y \in \mathcal{C}, b_1^T x = \mu, b_1^T y = \mu\} \tag{6.2}$$

is the difference of maximum and minimum over the set obtained after adding constraint $b_i^T x = \mu$ to $\mathcal{C}$. Now $G(b_2) \leq F_2(b_2)$ because the feasible solution in (2.9) for $F_2(b_2)$ are contained in the feasible set of (6.2). Therefore, the number of nodes at second level of the tree is bounded by $(\lfloor F_1(b_1) \rfloor + 1)(\lfloor F_2(b_2) \rfloor + 1)$. A similar argument is true for any node at any level of the generalized branch-and-bound tree, hence the result follows. □

The progress of GBR-algorithm is measured by the function $\Delta := \prod_{i=1}^{k} F_i(b_i)^{k+1-i}$ (or $\prod_{i=1}^{k} \|b_i\|_E^{k+1-i}$ in LLL algorithm with ellipsoidal approximation).

**Proposition 6.2** *Let $F_j(b_j) \geq 1$ for $j = 1, \ldots, k$. $\Delta$ and $\tilde{\Delta}$ satisfy the following inequalities*

$$\Delta^{\frac{1}{k}} \leq \tilde{\Delta} \leq k2^k\Delta.$$

**Proof:** The first inequality follows from the arithmetic mean and geometric mean for positive numbers. Since $F_j(b_j) \geq 1$, $2F_j(b_j) \geq \lfloor F_j(b_j) \rfloor + 1 \geq F_j(b_j)$. $\square$

The quantity $\Delta$ is decreased by a factor of $1 - \epsilon$ (or $\delta$ in LLL-algorithm), each time GBR-algorithm swap two successive vectors. This is the only time $\Delta$ (or a bound on $\Delta$ in LLL) improves in these algorithms. In view of Propositions 6.1 and 6.2 the swap step has an impact on improving the worst case bound on the total size of generalized branch-and-bound tree. In this sense each swap step in these algorithms is attempting to do a "global" reduction in the size of the branch-and-bound tree. Our computational experience in Mehrotra, *et. al.* [20] suggests that overall improvement in solvability of integer linear programs is directly correlated with the number of swaps performed by the LLL algorithm. One may monitor $\Delta$ or $\tilde{\Delta}$ directly in the implementation of GBR or LLL algorithm to decide an early termination of these algorithms.

## 7. Mixed Integer Linear Programming Problems

For (FMILP) let

$$\bar{\mathcal{P}} := \{x \in \mathbb{R}^{n+\bar{n}} \,|\, x \geq 0, Rx = r\},$$

$$\hat{\mathcal{P}} := \{x \in \mathbb{R}^{n+\bar{n}} \,|\, x \geq 0, Bx_z + Cx_c = d\}.$$

Clearly, $\bar{\mathcal{P}} = \hat{\mathcal{P}} \cap \{x_z \in \mathbb{R}^n | Ax_z = a\}$. For a set $\mathcal{C}$ in $\mathbb{R}^{n+\bar{n}}$, let

$$Proj_{x_z}(\mathcal{C}) := \{x_z \in \mathbb{R}^n \,|\, \text{ there exists a } x_c \in \mathbb{R}^{\bar{n}} \text{ such that } (x_z, x_c) \in \mathcal{X}\}.$$

The problem of finding a feasible solution of $\bar{\mathcal{P}}$ is equivalent to finding a feasible solution in $Proj_{x_z}(\bar{\mathcal{P}})$. The following proposition shows that the equality constraints $Ax_z = a$ are retained in the computation of the projection set.

**Proposition 7.1**

$$Proj_{x_z}(\bar{\mathcal{P}}) = Proj_{x_z}(\hat{\mathcal{P}}) \cap \{x_z \in \mathbb{R}^n | Ax_z = a\}.$$

An explicit computation of $Proj_{x_z}(\bar{\mathcal{P}})$ as proposed by Lenstra [16] is not practical, and it may generate exponential number of constraints [7, Section 2.8]. In this case, the standard log-barrier methods may no longer produce a polynomial $\gamma-$approximation of $Proj_{x_z}(\bar{\mathcal{P}})$. In this section we develop a methodology for computing branching hyperplanes and rounding the center without requiring a projection of the original set.

### 7.1 Finding Branching Hyperplane for Mixed Integer Case

The following lemma gives a way to compute the width of a polyhedral set along a direction of interest without computing the projection of $\bar{\mathcal{P}}$.

**Lemma 7.1** *Let $Z, Z^*$ satisfying $Z^T Z^* = I$ be bases for $\Lambda$ and $\Lambda^*$ respectively, where $\Lambda$ is the kernel lattice of $A$ and $\Lambda^*$ is an adjoint lattice of $A$. For any $u \in \mathbb{Z}^n$,*

$$\max_{x \in \bar{\mathcal{P}}} u^T x_z - \min_{x \in \bar{\mathcal{P}}} u^T x_z = \mathcal{W}(u, Proj_{x_z}(\bar{\mathcal{P}})).$$

*Furthermore, for every $u \in \mathbb{Z}^n$, there exists a $u^* \in \Lambda^*$ such that*

$$\max_{x \in \bar{\mathcal{P}}} u^{*T} x_z - \min_{x \in \bar{\mathcal{P}}} u^{*T} x_z = \mathcal{W}(u, Proj_{x_z}(\bar{\mathcal{P}})).$$

**Proof**: The first part of this Lemma follows immediately by recalling $(\hat{x}_z, \hat{x}_c) \in \bar{\mathcal{P}}$, if and only if $\hat{x}_z \in Proj_{x_z}(\bar{\mathcal{P}})$, and the objective values in $\bar{\mathcal{P}}$ and $Proj_{x_z}(\bar{\mathcal{P}})$ are the same.

Let $U = [U_A : Z]$ be a unimodular matrix that satisfies $AU = [H : 0]$, where $H$ is the Hermite normal form of $A$. Let $U^{-T} = [U_A^* : Z^*]$. Obviously, $U^T U^{-T} = I$, hence $Z^T U_A^* = 0$. Since $U$ is a unimodular matrix, $U^{-T}$ is also a unimodular matrix, hence it is a basis for $\mathbb{Z}^n$. Hence, for any $u \in \mathbb{Z}^n$, we have integer vectors $p^* \in \mathbb{Z}^k$ and $q^* \in \mathbb{Z}^{n-k}$ such that $u = Z^* p^* + U_A^* q^*$. Now for any $x \in \bar{\mathcal{P}}$, we write $x_z = v + Zy$ for some $y \in \mathbb{R}^k$, and $v$ satisfying $Av = a$. Hence, $x_z^T u = x_z^T (Z^* p^* + U_A^* q^*) = (v + Zy)^T (Z^* p^* + U_A^* q^*) = x_z^T Z^* p^* + v^T U_A^* q^*$. If $x_{\max}$ ($x_{\min}$) is the maximizer (minimizer) of $\max_{x \in \bar{\mathcal{P}}} u^T x_z$ ($\min_{x \in \bar{\mathcal{P}}} u^T x_z$), then $u^T((x_{\max})_z - (x_{\min})_z) = (Z^* p^*)^T((x_{\max})_z - (x_{\min})_z)$, hence $u^* = Z^* p^*$ is the desired vector. □

As a consequence of Lemma 7.1 we can formulate the problem of finding a thin direction over the lattice $\Lambda^*$. Let $\mathcal{E}(w, Q) = \{x \in \mathbb{R}^{n+\bar{n}} \mid \|x - w\|_Q \leq 1, Rx = r\}$ be an ellipsoid inscribed in $\bar{\mathcal{P}}$ and $\mathcal{E}(w, Q) \subseteq \bar{\mathcal{P}} \subseteq \mathcal{E}(w, Q/\gamma)$, where $\gamma$ is the approximation parameter. The branching hyperplane finding problem for the ellipsoid $\mathcal{E}(w, Q)$ is

$$\min_{u \in \Lambda^* \setminus 0} \left\| Q^{-1/2} \begin{bmatrix} u \\ 0 \end{bmatrix} \right\|_{P_{RQ^{-1/2}}}. \tag{7.1}$$

### 7.2 Rounding the Center

The center of the ellipsoid $w$ is rounded to a solution as follows. Let $w_z$ be the components for the integer variables, and $w_c$ be the components of the continuous variables. Let $v$ be a solution such that $Rv = r$, where $v = \begin{bmatrix} v_z \\ v_c \end{bmatrix}$, $v_z \in \mathbb{Z}^n$ and $v_c \in \mathbb{R}^{\bar{n}}$. Clearly $v_z$ statisfies $Av_z = a$. Since $w_z - v_z \in \mathcal{N}(A)$, $w_z - v_z = Z\zeta_z$ for some $\zeta_z \in \mathbb{R}^k$. Then $\bar{v}_z = v_z + Z\lfloor \zeta_z \rceil$ is a rounded integer solution satisfying $Ax_z = a$. We construct a solution $\bar{v} = (\bar{v}_z, \bar{v}_c)$ of $Rv = r$ by letting $\bar{v}_c$ be a solution to the problem:

$$\begin{aligned} &\min_{x_c} \left( \begin{bmatrix} \bar{v}_z \\ x_c \end{bmatrix} - w \right)^T Q \left( \begin{bmatrix} \bar{v}_z \\ x_c \end{bmatrix} - w \right) \\ &s.t.\ B\bar{v}_z + Cx_c = b. \end{aligned} \tag{7.2}$$

If $\bar{v}$ satisfies $\|w-\bar{v}\|_Q \leq 1$, then a feasible solution is found; Otherwise, we show that there exists a thin direction in an adjoint lattice $\Lambda^*$. The following proposition shows that an ellipsoid in the original space is projected to an ellipsoid in the pure integer space.

**Proposition 7.2** *Let* $Q = \begin{bmatrix} Q_z & Q_{zc} \\ Q_{zc}^T & Q_c \end{bmatrix}$. *Then* $Proj_{x_z}(\mathcal{E}(w,Q)) = \tilde{\mathcal{E}}(w_z,\tilde{Q})$, *where* $\tilde{\mathcal{E}}(w_z,\tilde{Q}) := \{x_z \in \mathbb{R}^n | \|x_z - w_z\|_{\tilde{Q}} \leq 1, Ax_z = a\}$ *and* $\tilde{Q} = Q_z - Q_{zc}Q_c^{-1}Q_{zc}^T + (B - CQ_c^{-1}Q_{zc}^T)^T(CQ_c^{-1}C^T)^{-1}(B - CQ_c^{-1}Q_{zc}^T)$.

**Proof:** Without loss of generality let $w = 0$, hence $w_z = 0$. Since a line in the original set projects to a line, it suffices to show that the boundary of $Proj_{x_z}(\mathcal{E}(0,Q))$ is an ellipse defined by $\tilde{Q}$. For any given $x_z$ (not necessarily integral), satisfying $Ax_z = 0$, consider the following optimization problem:

$$\max_{\rho \geq 0,\, x_c} \rho \tag{7.3}$$

$$s.t. \quad \begin{bmatrix} \rho x_z \\ x_c \end{bmatrix}^T \begin{bmatrix} Q_z & Q_{zc} \\ Q_{zc}^T & Q_c \end{bmatrix} \begin{bmatrix} \rho x_z \\ x_c \end{bmatrix} \leq 1, \tag{7.4}$$

$$\rho B x_z + C x_c = 0. \tag{7.5}$$

For a solution $\rho^*, x_c^*$ of (7.3-7.4), the point $(\rho^* x_z, x_c^*)$ is on the boundary of $\mathcal{E}(0,Q)$. Below we show that $\rho^* x_z$ is on the boundary of $\tilde{\mathcal{E}}(0,\tilde{Q})$. An optimal solution $(x_c^*, \rho^*, \lambda^*, \mu^*)$ of (7.3-7.4) satisfies the KKT optimality conditions:

$$\begin{aligned} \lambda^*(2\rho^* Q_{zc}^T x_z + 2Q_c x_c^*) + C^T\mu^* &= 0, \\ -1 + \lambda^*(2\rho^*(x_z^T Q_z x_z) + 2x_z^T Q_{zc} x_c^*) + \mu^{*T} B x_z &= 0, \\ (\rho^* x_z)^T Q_z(\rho^* x_z) + 2\rho^* x_z^T Q_{zc}^T x_c^* + x_c^{*T} Q_c x_c^* &= 1, \\ \rho^* B x_z + C x_c^* &= 0, \end{aligned}$$

where $\lambda^*$ and $\mu^*$ are the Lagrangian multipliers corresponding to the constraints (7.4) and (7.5), respectively. It is an exercise in algebra to show that $\rho^* x_z$ satisfying the KKT conditions satisfies

$$(\rho^* x_z)^T \tilde{Q}(\rho^* x_z) = 1,\ Ax_z = 0.$$

This completes the proof. □

Since $w_z$ is the projection of $w$, and since a line in the original space projects to a line, $\tilde{\mathcal{E}}(w_z,\tilde{Q})$ and $\tilde{\mathcal{E}}(w_z,\tilde{Q}/\gamma)$ are inscribed and circumscribing ellipsoids of $Proj_{x_z}(\bar{\mathcal{P}})$, i.e.,

$$\tilde{\mathcal{E}}(w_z,\tilde{Q}) \subseteq Proj_{x_z}(\bar{\mathcal{P}}) \subseteq \tilde{\mathcal{E}}(w_z,\tilde{Q}/\gamma).$$

Hence, a $\gamma$-approximation of $Proj_{x_z}(\bar{\mathcal{P}})$ is available. Note that for the log-barrier analytic center and associated Hessian, in the worst case $\gamma = n + \bar{n}$. This is worse than $\gamma = n$ possible when

$Proj_{x_z}(\bar{\mathcal{P}})$ is computed explicitly, and $\gamma = n - m$ possible if the full dimensional set is considered by eliminating the equality constraints.

From Proposition 7.2 we can easily show the following corollary, which states that checking the feasibility of $\bar{v}$ in $\mathcal{E}(w, Q)$ can be done by checking if $v_z$ is in $\tilde{\mathcal{E}}(w_z, \tilde{Q})$ and vice versa.

**Corollary 7.1** *If $\|w_z - v_z\|_{\tilde{Q}} \leq 1$, then $\bar{v}$ obtained by solving (7.2) statisfies $\|w - \bar{v}\|_Q \leq 1$. Futhermore, if $\|w - \bar{v}\|_Q > 1$, then $\|w_z - \bar{v}_z\|_{\tilde{Q}} > 1$.* □

Using Proposition 7.2 and following the arguments in Lemma 7.1 we can show that

$$\mathcal{W}(u, \tilde{\mathcal{E}}(w_z, \tilde{Q})) = \mathcal{W}\left(\begin{bmatrix} u \\ 0 \end{bmatrix}, \mathcal{E}(w, Q)\right)$$

for any $u \in \mathbb{Z}^n$. Hence, the branching hyperplane finding problem (7.1) is equivalent to solving

$$\min_{u \in \Lambda^* \setminus 0} \|\tilde{Q}^{-1/2} u\|_{P_{A\tilde{Q}^{-1/2}}}. \tag{7.6}$$

Now we are ready to show that if $\bar{v}$ constructed in (7.2) is not a feasible solution then a good branching hyperplane can be obtained by solving (7.1).

**Theorem 7.1** *Let $\tilde{Q}^{1/2} Z$ be LLL-reduced. If $\bar{v}$ satisfies $\|w - \bar{v}\|_Q > 1$. Then there exists a branching direction $u \in \mathcal{K}^*(A^T)$ such that*

$$\mathcal{W}\left(\begin{bmatrix} u \\ 0 \end{bmatrix}, \bar{\mathcal{P}}\right) \leq \gamma (3/\sqrt{2})^k$$

*In particular, one such $u$ is given by $Z^* e_j$, where $j = argmin\{\|\bar{Z}_j^*\|\}$ and $\bar{Z}^* := P_{A\tilde{Q}^{-1/2}} \tilde{Q}^{-1/2} Z^*$.*

**Proof:** From Corollary 7.1 and $\|w - \bar{v}\|_Q > 1$, we have $\|w_z - v_z\|_{\tilde{Q}} > 1$. Theorem 4.2 shows that $\mathcal{W}(u, Proj_{x_z}(\bar{\mathcal{P}})) \leq \gamma (3/\sqrt{2})^k$ for $u = Z^* e_j$, where $j = argmin\{\|\bar{Z}_j^*\|\}$. Using Proposition 7.1 we have $\mathcal{W}\left(\begin{bmatrix} u \\ 0 \end{bmatrix}, \bar{\mathcal{P}}\right) = \mathcal{W}(u, Proj_{x_z}(\bar{\mathcal{P}}))$. This completes the proof. □

Theorem 7.1 and the construction of the rounding procedure may give the impression that we need to compute $\tilde{Q}$, and a separate LLL-reduced $\tilde{Q}^{1/2} Z$ needs to be computed. In the following proposition we show that such a basis is available from $Z^*$ generated for the minimum width problem (7.1).

**Proposition 7.3** *Let $Z$ be a basis of $\Lambda$, and $Z^*$ be a basis of an adjoint lattice of $A$ satisfying $Z^T Z^* = I$. Assume that the vectors of $P_{RQ^{-1/2}} Q^{-1/2} \begin{bmatrix} Z^* \\ 0 \end{bmatrix}$ are 2-reduced. Then, the vectors $\tilde{Q}^{1/2} Z$ are 2-reduced in the reverse order.*

**Proof:** Because of Lemma 7.1 part (2) we have

$$\mathcal{W}(u, \tilde{\mathcal{E}}(w_z, \tilde{Q})) = \mathcal{W}\left(\begin{bmatrix} u \\ 0 \end{bmatrix}, \mathcal{E}(w, Q)\right),$$

hence the norm $\left\| Q^{-1/2} \begin{bmatrix} Z_i^* \\ 0 \end{bmatrix} \right\|_{P_{RQ^{-1/2}}} = \|\tilde{Q}^{-1/2} Z_i^*\|_{P_{A\tilde{Q}^{-1/2}}}$. Hence, 2-reduced $P_{RQ^{-1/2}} Q^{-1/2} \begin{bmatrix} Z^* \\ 0 \end{bmatrix}$ gives a 2-reduced $P_{A\tilde{Q}^{-1/2}} \tilde{Q}^{-1/2} Z^*$. The proof is complete by using Proposition 4.1. □

Once a 2-reduced $Z$ is computed, in order to compute a size-reduced $\tilde{Q}^{1/2} Z$ we can form $Z^T \tilde{Q} Z$, and work with its Cholesky factors. Computation of $Z^T \tilde{Q} Z$ requires working with inverses of $Q_c$ or $C Q_c^{-1} C$. In the linear case $Q_c$ is a diagonal matrix, and computations with $C Q_c^{-1} C$ are standard interior point computations.

## 8. Mixed Integer Convex Feasibility Problems with Self-Concordant Barrier

The method for finding a feasible solution of a mixed integer convex set with self-concordant barrier is similar to the method for the linear case. First note that all the results of the previous sections hold for general convex sets. While implementing LS-algorithms (which do not necessarily involve rounding of a center solution) we simply need to replace $\mathcal{P}$ with $\mathcal{C}$ in the development of previous sections. Also, our development of Lenstra's algorithm remains unchanged provided that an ellipsoidal approximation of the convex set is available at each node. We discuss this below. Let us consider (FMICP) and assume that the set $\hat{\mathcal{C}} := \{x \mid c_i(x) \leq 0, i = 1, \ldots, l\}$ is bounded with a non-empty interior. An approach to handle non-empty interior assumption is described later in the section. Let $f(x)$ be a self-concordant barrier associated with $\hat{\mathcal{C}}$ with complexity value $\theta$. Then, according to Renegar [24, Corollary 2.3.5] we have

$$\mathcal{E}(w, \nabla^2 f(w)) \subseteq \mathcal{C} \subseteq \mathcal{E}(w, \frac{1}{4\theta + 1} \nabla^2 f(w)). \tag{8.1}$$

In fact, we don't need exact computations of $w$. It is sufficient to find an approximate $w$. Let $\tilde{w} \in \mathcal{C}$ be such that $\|p\|_{\nabla^2 f(x)} \leq 1/4$, where

$$p := -\nabla^2 f(\tilde{w})^{-1} f(\tilde{w})$$

is the Newton direction of $f$ at $\tilde{w}$. Then, using Renegar [24, Theorem 2.2.5] we have $\|w - \tilde{w}\|_{\nabla^2 f(\tilde{w})} \leq 1/2$. Now using the definition of self-concordance function we can see that

$$\mathcal{E}(w, 2\nabla^2 f(w)) \subseteq \mathcal{C} \subseteq \mathcal{E}(w, \frac{1}{4(2\theta + 1)} \nabla^2 f(w)).$$

The log-barrier function $\rho(x) := -\sum_{i=1}^{l} \ln c_i(x)$ is a self concordant barrier in many important situations. In particular, for the $n$ dimensional non-negative orthant, the second-order cone $\left\| \begin{bmatrix} x_2 \\ \vdots \\ x_n \end{bmatrix} \right\|_2^2 \leq x_1^2$ defined using $n$ variables, and semidefinite cones over $n \times n$ semidefinite matrices

(using determinants), this barrier has complexity values $n, 2$, and $n$ respectively.

**Computation of Analytic Center**
We may compute the log-barrier analytic center with a two phase approach. In the first phase we compute a feasible interior solution using an interior point algorithm of choice, and in the second phase we use this interior solution as a starting point to get a suitable approximation of the analytic center. The feasibility problem is given by

$$\min\{x_a \mid Rx = r, c_i(x) - (c_i(x^0) + \kappa)x_a \leq 0, i = 1, \ldots, l\}, \tag{8.2}$$

where $x_a$ is an artificial variable, $x^0$ is any point satisfying $Rx^0 = r$, and $\kappa$ is a suitably large constant that ensures appropriate initial condition ($(x^0, 1)$ being close to the central path for the path following primal interior point methods). A point $(x^*, x_a^*)$ is found via an interior point algorithm (for example, the path following algorithms in Nesterov and Nemirovskii [23] or Renegar [24]) applied to (8.2) satisfying one of the three criteria: (i) a point $(x^*, 0)$ satisfying $Rx^* = r, c_i(x^*) < 0, i = 1, \ldots, l$, (ii) a proof that the optimum objective value of (8.2), $x_a^*$, is greater than $\hat{\epsilon} > 0$, or (iii) $x_a^* \leq \hat{\epsilon}$. In case (i) we use $x^*$ as a starting point and apply damped Newton iterations to obtain $\tilde{w}$. In case (ii) we have an infeasible node. In case (iii) within tolerance at the current node we are unable to verify if the problem is infeasible or it has a non-empty interior. In this case we work with the approximate set

$$\tilde{\mathcal{C}} := \{x \mid Rx = r, c_i(x) - 2(c_i(x^0) + \kappa)\hat{\epsilon} \leq 0, i = 1, \ldots, l\}$$

for which $x^*$ is a feasible interior solution, and find an approximate analytic center $\tilde{w}$ of the set $\tilde{\mathcal{C}}$. If $\epsilon \leq 2(c_i(x^0) + K)\hat{\epsilon}$, we are now working with $\epsilon-$FMICP. We use this analytic center and the associated barrier Hessian to generate the branching hyperplanes and a nearby rounded solution. Note that if a feasible solution to $\tilde{\mathcal{C}}$ is found, then it is within a small perturbation (given with the choice of $\epsilon$) of set $\mathcal{C}$; otherwise, the number of branching hyperplanes is bounded by Theorem 4.2. Using the fact that the analytic center for problems with self-concordant log-barrier can be found in arithmetic operations polynomial in $n + n_c, l, \theta$, $\ln(1/\epsilon)$, and the cost of computing gradient and Hessian matrices (see Renegar [24, Theorem 2.4.1]), we have the following result.

**Theorem 8.1** *There exists an algorithm for finding a solution $(x_z, x_c)$ in $\epsilon$-FMICP in time that is polynomial in fixed $n+n_c$, ($l$, $\theta$, $\ln(1/\epsilon)$, and cost of computing gradient and Hessian) or proving that no such solution exists.*

## 9. Conclusion

In this paper, we have fundamentally restructured Lenstra's algorithm for solving mixed integer programming problems. Special attention is paid to the branching hyperplane finding problem, the crucial step of Lenstra's algorithm. We have introduced the concept of adjoint lattice and

showed that this lattice can be used to generate branching hyperplanes in the original space. Alternative methods for finding general branching directions are developed. This work has opened a host of possibilities for how branching on hyperplane algorithms can be implemented.

In this paper we also developed a method for solving mixed integer convex programs admitting self-concordant barriers. The main algorithmic steps of this method can also be used to develop a method for solving mixed integer convex programs where the convex functions are not differentiable. However, this development requires use of Vaidya center [28]. In particular, we need to extend the method of Anstreicher [4] for a more general setting involving equality constraints. The details of such a method are presented in [21].

This paper focused on the feasibility problem because the essential features of Lenstra's algorithm can be described easily in this setting. In a forth coming computational work [20] we will describe an implementation for optimization problems based on the work presented here.

## Bibliography


[1] K. Aardal and A. K. Lenstra (2004), "Hard Equality constrained Integer Knapsacks," *Mathematics of Operations Research*, 29(3), 724-738.

[2] K. Aardal, R. Bixby, C. Hurkens, A. Lenstra, and J. Smeltink (2000), "Market Split and Basis Reduction: Towards a Solution of the Cornuéjols-Dawande Instances," *INFORMS Journal on Computing*, 12, 192-202.

[3] K. M. Anstreicher (1998), "Towards A Practical Volumetric Cutting Plane Method For Convex Programming," *SIAM Journal of Optimization*, **9**(1), 190-206.

[4] K.M. Anstreicher (1999), "Ellipsoidal approximations of convex sets based on the volumetric barrier," *Mathematics of Operations Research*, 24(1), 193–203.

[5] K.M. Anstreicher (2001), "Improved complexity for maximum volume inscribed ellipsoids," technical report, Departemnt of Management Science, University of Iowa, Iowa City.

[6] L. Babai (1986), "On Lovász' Lattice Reduction and The Nearest Lattice Point Problem," *Combinatorica 6(1)*, 1-13.

[7] D. Bertsimas and J.J. Tsitsiklis (1997), "Introduction to Linear Optimization," Athena Scientific.

[8] J.W.S. Cassels (1971), "An Introduction to the Geometry of Numbers," Springer-Verlag.

[9] W. Cook, T. Rutherford, H.E. Scarf, and D. Shallcross (1993), "An implementation of the generalized basis reduction algorithm for integer programming," *ORSA Journal of Computing*, 5, 206-215.

[10] L. Gao and Y. Zhang (2002), "Computational Experience with Lenstra's Algorithm," *TR02-12*, Department of Computational and Applied Mathematics, Rice University.

[11] F. John (1948), "Extremum problems with inequalities as subsidiary conditions," in *Studies and Essays, Presented to R. Courant on his 60th Birthday*, Wiley Interscience, 187–204.

[12] L.G. Khachiyan (1996), "Rounding of polytopes in the real number model of computation," *Mathematics Of Operations Research*, 21(2), 307–320.

[13] L.G. Khachiyan and M.J. Todd (1993), "On the complexity of approximating the maximal volume inscribed ellipsoid for a polytope," *Mathematical Programming*, 61, 137-159.

[14] H. Koy and C.P. Schnorr (2001), "Segment LLL-Reduction of Lattice Bases," *Lecture Notes in Computer Science 2146*, 67-80.

[15] A.K. Lenstra, H.W. Lenstra, and L. Lovász (1982), "Factoring polynomials with rational coefficients," *Math. Ann.*, 261, 515–534.

[16] H.W. Lenstra (1983), "Integer programming with a fixed number of variables," *Mathematics of Operations Research*, 8(4), 538–548.

[17] L. Lovász (1986), "An algorithmic theory of numbers, Graphs and Convexity," *SIAM.*

[18] L. Lovász and H.E. Scarf (1992), "The generalized basis reduction algorithm," *Mathematics of Operations Research*, 17(3), 751–764.

[19] S. Mehrotra and Z. Li (2001), "Segment LLL Reduction of Latice Bases Using Modular Arithmetic," IE/MS technical report, Northwestern University, Evanston, IL 60208.

[20] S. Mehrotra, Z. Li, and H. Sheng (2005), "On an implementation of generalized branching methods for pure integer linear programs," (in preparation).

[21] S. Mehrotra (2005), "A Method for Solving Mixed Integer Nondifferentiable Convex Programs," (in preparation).

[22] S. Mehrotra and Y. Ye (1993), "Finding an interior point in the optimal face of linear programs," *Mathematical Programming*, 62, 497-515.

[23] Y. E. Nesterov and A. S. Nemirovski (1994), "Interior Point Polynomial Algorithms in Convex Programming," SIAM Publications. SIAM, Philadelphia, USA.

[24] J. Renegar (2001), "A Mathematical View of Interior-Point Methods in Convex Optimization," MPS-SIAM series on Optimization, 40, 59–93.

[25] A. Schrijver (1986), "Theory of linear and integer programming," John-Wiley.

[26] S.P. Tarasov, L.G. Khachiyan, and I.I. Erlich (1988), "The method of inscribed ellipsoids," *Soviet Math. Doklady*, 27, 226–230.

[27] J. Owen and S. Mehrotra (2001), "Experimental Results on Using General Disjunctions in Branch-and-Bound For General-Integer Linear Programs," *Computational Optimization and Applications*, 20(1), 159-170.

[28] P.M. Vaidya (1996), "A new algorithm for minimizing convex functions over convex sets," *Mathematical Programming*, 73, 291–341.

[29] X. Wang (1997), "An Implementation of the Generalized Basis Reduction Algorithm for Convex Integer Programming," Ph.D. dissertation, Department of Economics, Yale University.

[30] Y. Ye (1992), "On the convergence of interior-point algorithms for linear programming," *Mathematical Programming*, 57, 325-335.

## 10. Appendix

### 10.1 LLL Basis Reduction Algorithm For Ellipsoidal Norms

Once the problem of finding the reduced basis of a given lattice is appropriately formulated, the generalization of Lenstra, Lenstra, and Lovász [15] basis reduction algorithm and its relatives is straight forward. We describe this below. We assume that the lattice is given by a basis $B = [b_1, \ldots, b_n]$ (either $Z$ or $Z^*$) and norms and inner products of the vectors are computed under the matrix operator $E$. The matrix $E$ can be a symmetric positive definite matrix, the orthogonal projection matrix $P_A$, or scaled orthogonal projection matrix $P = Q^{-1/2}P_{AQ^{-1/2}}Q^{-1/2}$ in the pure integer case, or $P = Q^{-1/2}P_{RQ^{-1/2}}Q^{-1/2}$ in the mixed integer case.

The LLL algorithm with respect to the ellipsoidal norm is presented in Figure 2. The quantities $\hat{b}_i$ and $\Gamma_{ij}$ are defined in (2.6 - 2.7). The LLL basis reduction algorithm performs two types of computations: Basis vector size reduction (Size Reduction for short) and basis vector swap (Swap for short).

**Size Reduction of a Basis Vector**

The size reduction of a full basis was discussed it Section 2. Starting from a size reduced basis in LLL algorithm at each iteration only the size reduction of a basis vector is necessary. While considering the $i$-th basis vector, the algorithm first ensures that $|\Gamma_{i-1,i}| < 1/2$. If needed, an integer multiple $\lfloor \Gamma_{i-1,i} \rceil$ of $b_{i-1}$ is subtracted from $b_i$ to ensure this. This is called size reduction of $b_i$ using $b_{i-1}$. We can ensure $|\Gamma_{j,i}| < 1/2$ for $j = 1, \ldots, i-2$ by inductively size reducing $b_i$ against $b_{i-1}, \ldots, b_1$. This is called size reduction of basis vector $b_i$.

**Swap of Two Successive Basis Vectors**

The work in LLL algorithm results from the swap of two successive basis vectors. Basis vectors $b_{i-1}$ and $b_i$ are swapped if $\|\hat{b}_i\|_E^2 < (\delta - \Gamma_{i-1,i}^2)\|\hat{b}_{i-1}\|_E^2$. The index counter $i$ is reduced to $i-1$ in the algorithm; otherwise the counter is increased to $i+1$. When two successive vectors are swapped $\|\hat{b}_{i-1}\|_E^2$, $\|\hat{b}_i\|_E^2$ and the coefficients of column/row $i-1$ and $i$ of $\Gamma$ are updated to restore the size reduced property. This is done by the following recurrence using $\mu := \Gamma_{i-1,i}, b := \|\hat{b}_i\|_E^2 + \mu^2\|\hat{b}_{i-1}\|_E^2$:

$$\Gamma_{i-1,i} = \mu\|\hat{b}_{i-1}\|_E^2/b,\ \|\hat{b}_i\|_E^2 = \|\hat{b}_{i-1}\|_E^2\|\hat{b}_i\|_E^2/b,\ \|\hat{b}_{i-1}\|_E^2 = b, \tag{10.1}$$

$$\begin{bmatrix} \Gamma_{j,i-1} \\ \Gamma_{j,i} \end{bmatrix} = \begin{bmatrix} \Gamma_{j,i} \\ \Gamma_{j,i-1} \end{bmatrix} \text{ for } j = 1, \ldots, i-2, \tag{10.2}$$

$$\begin{bmatrix} \Gamma_{i-1,j} \\ \Gamma_{i,j} \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_{i-1,i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -\mu \end{bmatrix} \begin{bmatrix} \Gamma_{i-1,j} \\ \Gamma_{i,j} \end{bmatrix}, \text{ for } j = i+1, \ldots, n. \tag{10.3}$$

The absolute value of the coefficients $\Gamma_{j,i}$ of $\Gamma$ obtained after the swap can become larger than $1/2$, so a size reduction step is performed to ensure that these coefficients are less than $1/2$.

**Algorithm: LLL Basis Reduction Algorithm w.r.t. Ellipsoidal Norm**

{

INPUT: A basis $b_1, b_2, \ldots, b_k$;

OUTPUT: A LLL reduced basis;

Initialize $\delta$ and compute Gram-Schmidt orthogonal basis $\hat{b}_1, \ldots, \hat{b}_k$;

WHILE $i \le k$ DO

IF $\|\hat{b}_i\|_E^2 < (\delta - \Gamma_{i-1,i}^2)\|\hat{b}_{i-1}\|_E^2$ THEN

**SWAP** $b_i$ and $b_{i-1}$, and update $\Gamma$;

[**Size Reduction**] Size reduce vectors $b_k, \ldots, b_i$;

IF $i > 2$ THEN $i \leftarrow \max\{i-1, 1\}$;

ELSE

$i \leftarrow i+1$;

}

Figure 2: LLL Basis Reduction Algorithm

We refer the reader to Mehrotra and Li [19] for a more detailed review of the developments on LLL algorithm since it was originally proposed.

### 10.2 Generalized Basis Reduction Algorithm

The GBR algorithm of Lovász and Scarf [18] is described in Figure 3. At each iteration it requires several computations of $F_i(\cdot)$, which require solutions of corresponding optimization

problems. In the worst case, the number of (convex) minimization problems solved by GBR-algorithm can grow exponentially in $n$ (although 'polynomial' for a fixed $n$) [18, Equation (6), Theorem 6].

**Algorithm: Generalized Basis Reduction Algorithm**

{

INPUT: A basis $b_1, \ldots, b_k$;

OUTPUT: A LS-reduced basis;

WHILE $i \leq k$ DO

Replace $b_{i+1}$ by $b_{i+1} + \mu b_i$ with $\mu$, the integer that minimizes $F_i(b_{i+1} + \mu b_i, \mathcal{C})$.

IF $F_i(b_{i+1}, \mathcal{C}) < (1-\epsilon)F_i(b_i, \mathcal{C})$, THEN

swap $b_i$ and $b_{i-1}$;

IF $i > 2$ THEN $i \leftarrow \max\{i-1, 1\}$;

ELSE

$i \leftarrow i+1$;

}

Figure 3: Generalized Basis Reduction Algorithm

The quantity $\mu$ in the GBR-algorithm is calculated as follows [18]. The function $F_i(b_{i+1} + \mu b_i, \mathcal{C})$ is convex in $\mu$, and after relaxing the integrality requirement on $\mu$, we can solve it via the minimization problem (2.8). Denote the optimal solution by $\mu^*$ and then evaluate $F_i(b_{i+1} + \lfloor \mu^* \rfloor b_i, \mathcal{C})$ and $F_i(b_{i+1} + \lceil \mu^* \rceil b_i, \mathcal{C})$. If $\mathcal{C}$ is a polyhedral set, then the problems solved in (2.8) are linear programs.

Let us now consider the minimum width problem in the context of convex integer problems. For any $u \in \mathbb{R}^n$ it can be shown that [29, Theorem 4.10, Corollary 4.14]

$$\mathcal{W}(u, \mathcal{P}) = \max\{u^T z | z \in \mathcal{C} - \mathcal{C}\} = F_1(u, (\mathcal{C} - \mathcal{C})^*),$$

and

$$F_i(u, (\mathcal{C} - \mathcal{C})_i^*) = \max\{u^T(x-y) | x \in \mathcal{C}, y \in \mathcal{C}, B_i^T(x-y) = 0\},$$

where $B_i = [b_1, \ldots, b_i]$. The GBR-algorithm is now run by taking $\mathcal{C} := (\mathcal{C} - \mathcal{C})^*$. It can be shown [29] that the value of $\mu$ that solves $\min_{\mu \in \mathbb{R}} F_i(b_{i+1} + \mu b_i, (\mathcal{C} - \mathcal{C})^*)$ equals the Lagrange multiplier corresponding to the constraint $b_i^T(x-y) = 0$ in the optimal solution of:

$$\max\{b_{i+1}^T(x-y) | x \in \mathcal{C}, y \in \mathcal{C}, b_j^T(x-y) = 0, j = 1, \ldots, i\}, \tag{10.4}$$

i.e., it is available during the computation of $F_i(b_{i+1}, (\mathcal{C} - \mathcal{C})^*)$. Note also that if $\mathcal{C}$ is approximated with an ellipsoid $\mathcal{E}(w, Q)$, then the GBR-algorithm is run with the set $\mathcal{C} = (\mathcal{E}(0, Q/2))^*$, because

in this case $\mathcal{E}(w,Q) - \mathcal{E}(w,Q) = \mathcal{E}(0,Q/2)$. The computation for $F_i(b_{i+1}, \mathcal{E}(0,Q/2)^*)$ becomes

$$\max\{b_{i+1}^T x | x \in \mathcal{E}(0,Q/2), Ax = 0, B_i^T x = 0\}. \tag{10.5}$$

We state the following proposition without proof.

**Proposition 10.1** *The optimal objective value of (10.5) equals* $2\|\hat{b}_{i+1}\|_E$*, where* $E = P_A$ *or* $P$ *depending on the choice of* $Q$*.* $\square$

Proposition 10.1 provides an alternative way of performing computations in the LLL-algorithm. In such an implementation instead of maintaining $\Gamma_{j,i}$ and updating $\|\hat{b}_i\|$, one would compute these quantities from the solution of the optimization problem (10.5). Such an implementation approach may be more useful in some special situations.

The major difference in our use of GBR-algorithm and the one presented in Cook *et. al.* [9] and Wang [29] is that in our case we can work with the adjoint lattice given by $Z^*$, while Cook *et. al.* and Wang [29] consider the problem in full dimensional space and work with the lattice $\mathbb{Z}^n$.