INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While
the most advanced technological means to photograph and reproduce this document
have been used, the quality is heavily dependent upon the quality of the original
submitted.

The following explanation of techniques is provided to help you understand
markings or patterns which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document
photographed is “Missing Page(s)”’. If it was possible to obtain the missing
page(s) or section, they are spliced into the film along with adjacent pages.
This may have necessitated cutting thru an image and duplicating adjacent
pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it
is an indication that the photographer suspected that the copy may have
moved during exposure and thus cause a blurred image. You will find a
good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being
photographed the photographer followed a definite method in
“sectioning” the material. It is customary to begin photoing at the upper
left hand corner of a large sheet and to continue photoing from left to
right in equal sections with a small overlap. If necessary, sectioning is
continued again - beginning below the first row and continuing on until
complete.

4. The majority of users indicate that the textual content is of greatest value,
however, a somewhat higher quality reproduction could be made from
“photographs” if essential to the understanding of the dissertation. Silver
prints of “photographs” may be ordered at additional charge by writing
the Order Department, giving the catalog number, title, author and
specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as
received.

Xerox University Microfilms
300 North Zeeb Road
Ann Arbor, Michigan 48106
MYERS, Geraldine, 1939-
A PRIMAL-DUAL ALGORITHM FOR INTEGER PROGRAMMING
OVER A CONE.

Rice University, Ph.D., 1975
Mathematics

Xerox University Microfilms, Ann Arbor, Michigan 48106

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED.
RICE UNIVERSITY

A PRIMAL-DUAL ALGORITHM FOR INTEGER PROGRAMMING
OVER A CONE

by

Geraldine Myers

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

Thesis Director's Signature:

Richard W. Young

Houston, Texas
May, 1975
ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my thesis advisor, Professor Richard D. Young, for his unfailing support and encouragement. His thorough knowledge of integer programming was a vital factor in both motivating and directing the topic.

I am indebted to Professor Robert M. Thrall for his many valuable observations and suggestions, particularly those which helped improve the notation. The advice of the third member of my committee, Professor Angelo Miele, to "always simplify" inspired my approach to the topic.

For his encouragement to pursue this degree, I am especially grateful to Dr. Henry J. Kelley.

I would like to thank my mother for her abounding patience, understanding and encouragement during my many years of pursuing this degree. Her strength and love have always been my inspiration.

For the many hours spent tediously over a Xerox machine, I am grateful to my husband, Dean. Finally, to my sweet little girl, Kristin, who never cried or fussed when mommy was too busy for her, I dedicate this thesis with all my love and thanks.
INDEX

<table>
<thead>
<tr>
<th>CHAPTER 1. INTRODUCTION TO THESIS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Notation and Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Significance of the Problem</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Current Algorithms for Solving the ILPC</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Development of the Thesis</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 2. ALGEBRAIC RESULTS UTILIZED IN THE THESIS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Number Theory</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Group Theory</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Linear Algebra</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 3. A NEW EQUIVALENT REPRESENTATION OF THE ILPC</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Purpose of the Representation</td>
<td>15</td>
</tr>
<tr>
<td>3.2 Summary of the Theorems</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Preliminary Analysis</td>
<td>17</td>
</tr>
<tr>
<td>3.4 Preliminary Definition of the Adjoined Constraint</td>
<td>20</td>
</tr>
<tr>
<td>3.5 Triangularization Process</td>
<td>20</td>
</tr>
<tr>
<td>3.6 Explanation of the Triangularization Process</td>
<td>21</td>
</tr>
<tr>
<td>3.7 Calculating the Solution Set</td>
<td>25</td>
</tr>
<tr>
<td>3.8 Equivalence Algorithm</td>
<td>28</td>
</tr>
<tr>
<td>3.9 Number of Roots of a System of Congruences</td>
<td>31</td>
</tr>
<tr>
<td>3.10 Simplifying Assumption</td>
<td>32</td>
</tr>
<tr>
<td>3.11 Proof of Theorems</td>
<td>33</td>
</tr>
<tr>
<td>3.12 Numerical Examples</td>
<td>44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 4. PROPERTIES OF THE REPRESENTATION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Double Description</td>
<td>50</td>
</tr>
<tr>
<td>4.2 Group Properties</td>
<td>54</td>
</tr>
<tr>
<td>4.3 Proofs of Theorems</td>
<td>55</td>
</tr>
<tr>
<td>4.4 Numerical Examples</td>
<td>63</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION TO THESIS

1.1 Notation and Statement of the Problem

The following notation will be standard throughout the paper.

All vectors are column vectors unless otherwise specified.

Let:

\[ A = (m) \times (n) \text{ integer matrix with rank } r \]
\[ b = \text{ an integer } m \text{-vector} \]
\[ Q = \text{ a positive integer} \]
\[ A_i = \text{ the } i^{\text{th}} \text{ column of } A \]
\[ A_i = \text{ the } i^{\text{th}} \text{ row of } A \text{ (Note: } A_i \text{ is a row vector)} \]
\[ J_n = \{1, 2, \ldots, n\} \]
\[ S = (s_1, s_2, \ldots, s_k) \text{ represent a sequence of } k \]
\[ \text{distinct integers from the set } J_n \text{ (} k \leq n \text{).} \]
\[ T = (t_1, \ldots, t_{n-k}) \text{ represent a sequence of } (n-k) \]
\[ \text{distinct integers from } J_n \text{ not in } S. \]
\[ A^S = \text{ the } (m) \times (k) \text{ matrix whose columns in order} \]
\[ \text{from left to right are } A^{S_1}, A^{S_2}, \ldots, A^{S_k}. \]
\[ x^S = \text{ the } k \text{-vector } (x_{s_1}, x_{s_2}, \ldots, x_{s_k}). \]
\[ \emptyset = \text{ the empty set} \]
\[ \theta = \text{a vector or matrix of zeros.} \]

The integer linear programming problem over a cone (ILPC) may now be stated as follows:

\[
\begin{align*}
\text{Min. } f &= cx \\
\text{subject to } Ax &\equiv b \pmod{Q} \\
0 &\leq x_i < Q, x_i \text{ integer for all } i.
\end{align*}
\]

where \(x\) is the \(n\)-vector of unknown variables and \(c\) is a non-negative row vector. The system \(Ax \equiv b \pmod{Q}\) is a system of \(m\) linear congruences modulo \(Q\) noted simply by \([Ab]\).

### 1.2 Significance of the Problem

The significance of the ILPC is its relation to the general integer linear programming problem (ILP) which may be posed as follows:

\[
\begin{align*}
\text{Max } v_0 &= \overline{c}v \\
\text{subject to } \overline{A}v &= \overline{b} \\
v &\geq 0, \text{ integer}
\end{align*}
\]

where \(\overline{A}\) is the integer constraint matrix, \(\overline{b}\) is the integer right hand side, \(v\) is the vector of unknown variables (including slack variables where necessary) and \(\overline{c}\) is the row vector of integer objective function coefficients. Dropping the integer restriction on \(v\) results in a corresponding linear programming problem (LP) for which an optimal basic sequence \(S\) and corresponding non-basic sequence \(T\) may be determined using standard LP techniques (see Spivey and Thrall [1970]). Denote the optimal basis matrix \(\overline{A}^S\) by \(B\) and the remaining columns \(\overline{A}^T\) by \(N\). In terms of the sequences
S and T, the LP corresponding to 1.2.1 is given by:

\[
\begin{align*}
\text{Max } v_0 &= \frac{c^S}{c^T} v^S + \frac{c^T}{c} v^T \\
B v^S + N v^T &= \bar{b} \\
v^S, v^T &\geq 0
\end{align*}
\]

By expressing \( v^S \) in terms of \( v^T \) we can now rewrite the ILP as:

\[
\begin{align*}
\begin{bmatrix}
\text{Max } v_0 &= \frac{c^S}{c^T} B^{-1} \bar{b} - \left( \frac{c^S}{c^T} B^{-1} N - \frac{c^T}{c} \right) v^T \\
v^S &= B^{-1}(\bar{b} - N v^T) \\
v^S, v^T &\geq 0, \text{ integer.}
\end{bmatrix}
\end{align*}
\]

1.2.2

Denote the row vector \( (\frac{c^S}{c^T} B^{-1} N - \frac{c^T}{c}) \) by the row vector \( c \), and note that since \( B \) is an optimal L.P. basis matrix, \( c \geq 0 \).

If the L.P. optimal solution is integer, it is, of course, the solution to the I.L.P. as well. Otherwise, consider a relaxation of 1.2.2 by dropping the non-negativity constraints on \( v^S \) giving

\[
\begin{align*}
\begin{bmatrix}
\text{Max } v_0 &= \frac{c^S}{c^T} B^{-1} \bar{b} - c v^T \\
v^S &= B^{-1}(\bar{b} - N v^T) \\
v^S \text{ integer, } v^T &\geq 0, \text{ integer}
\end{bmatrix}
\end{align*}
\]

1.2.3

Since the feasible points of the L.P. associated with 1.2.3 correspond to a displaced cone \( C \), this relaxed problem is called an ILP over a cone (ILPC). The convex hull of all integer points within \( C \) is referred to as the corner polyhedron. An optimal solution to 1.2.3 for which the non-negativity constraints \( v^S \geq 0 \) are satisfied will be an optimal solution to the original ILP. Sufficient conditions
for \( v^S \geq 0 \) have been determined (See Gomory [1965] and Hu [1969]), and will be met if each component of \( b \) is suitably large.

To see that the ILPC as given by 1.2.3 is the same as the problem defined in Section 1, we first denote the variables \( v^T \) by \( x \).

Since the constant term \( c^S b^{-1} b \) may be dropped from \( v_0 \), the objective function may be written as \( \min f = cx \). The restriction that \( v^S = b^{-1}(\bar{b} - Nx) \) be integer may be replaced by the condition that \( b^{-1}(\bar{b} - Nx) \equiv 0 \pmod{1} \) where \( (\pmod{1}) \) means that each component of the vector is computed using modulo 1 arithmetic (see 2.1). We can now rewrite 1.2.3 as:

\[
\begin{align*}
\text{Min } f &= cx \\
(B^{-1}N)x &\equiv b^{-1}b \pmod{1} \\
x &\geq 0, \text{ integer}
\end{align*}
\]

Multiplying each constraint row by \( Q \), the lowest common denominator of all the entries in \( (B^{-1}N) \) and \( (b^{-1}b) \) gives an equivalent form (see 2.1)

\[
\begin{align*}
\text{Min } f &= cx \\
Ax &\equiv b \pmod{Q} \\
x &\geq 0, \text{ integer}
\end{align*}
\]

where \( A \) and \( b \) have integer entries, and, as before, \( c \) is a row vector with non-negative components. Since in any optimal solution to 1.2.4, \( x_i < Q \) for all \( i \) (see 2.1), we may rewrite the ILPC in the final form
Min \ f = cx \\
Ax \equiv b \pmod{Q} \\
0 \leq x_i \leq Q, x_i \text{ integer for all } i.

Thus the ILPC is a problem of determining an optimal vector from a finite subset of integer points in the corner polyhedron.

1.2.1 Smith Canonical Form of the ILPC

An alternate derivation of the ILPC as stated in Section I may be obtained (see Gomory [1965]) from 1.2.3 by first determining \( \Delta \), the Smith Normal Form of the optimal LP basis matrix \( B \), where \( \Delta = RBC \) (see 2.3.4). Define a new variable \( y = C^{-1}v^S \). Then the conditions

\[
Bv^S = \bar{\bar{B}} - Nv^T, \quad v^S \text{ integer}
\]

are equivalent to the conditions

\[
\Delta y = R(\bar{\bar{B}} - Nv^T), \quad y \text{ integer}.
\]

Let \( \delta_i \) denote the \( i \)th diagonal element of \( \Delta \). Then

\[
\delta_i y_i = R_i(\bar{\bar{B}} - Nv^T)
\]

and \( y_i \) is integer if and only if

\[
R_i(\bar{\bar{B}} - Nv^T) \equiv 0 \pmod{\delta_i}
\]

1.2.5

For any \( i \) such that \( \delta_i = 1 \), condition 1.2.5 is a trivial constraint since it is satisfied by any integer vector \( v^T \).

Assume \( \delta_i = 1 (i = 1, \ldots, k - 1) \) and \( \delta_i > 1 (i = k, \ldots, m) \) and define constraint rows \( k, \ldots, m \) as the non-trivial constraints.
By defining
\[ x = v^T, \]
\[ A'_{i} = R_{i} N \]
\[ b'_{i} = R_{i} \overrightarrow{b} \]  \( i = k, \ldots, m \)
\[ \delta = \begin{pmatrix} \delta_{k} \\ \vdots \\ \delta_{m} \end{pmatrix} \]
we may rewrite 1.2.3 as
\[
\begin{align*}
\text{Min } f &= cx \\
A'x &\equiv b' \pmod{\delta}
\end{align*}
\]
1.2.6
\[ x \geq 0, \text{ integer} \]
This is called Smith Canonical form of the ILPC. Multiplying \( A'_{i} \) and \( b'_{i} \) by the integer \( \delta_{m}/\delta_{i} \) (for all \( i \)) gives the ILPC in the form as stated in Section 1 with \( Q = \delta_{m} \). The advantage of deriving the ILPC from Smith Canonical form is that it will have a minimal number of non-trivial constraints.

1.3 Current Algorithms for Solving the ILPC

The ILPC was introduced by Gomory (1965) who recognized that in Smith Canonical form it was a knapsack problem over a finite Abelian group of order \( D = \prod_{i=k}^{m} \delta_{i} \). The method of solution proposed is based on a dynamic programming recursion formula and requires computation time proportional to \( nD \) where \( n \) is the dimension of the vector \( x \). Other methods of solution for the group knapsack problem were later proposed by Hu (1968), Shapiro (1968 and 1970) and
Glover (1969) each of which requires computation time proportional to \( D \). Of these only the method of Glover will be expounded upon (see Appendix A) since it is modified and utilized in the algorithm presented in Chapter 6.

1.4 Development of the Thesis

It is the purpose of the thesis to present a primal-dual algorithm for solving the ILPC. Chapter 2 contains a brief summary of the needed results from Number Theory, Linear Algebra, and Algebraic Group Theory. In Chapter 3 an equivalent representation of the ILPC is developed from which it can be immediately determined if any feasible solutions exist. Properties of the new representation are discussed in Chapters 3 and 4. An optimization algorithm is then presented in Chapters 5 and 6 which is primal-dual in the LP sense of generating at each stage a feasible solution and a solution which is optimal, provided it is feasible. The algorithm proceeds toward optimality in a finite number of stages each of which replaces the feasible solution by a new one with lower cost, whenever possible. Numerical examples are presented in Chapter 7. The thesis ends with the concluding remarks appearing in Chapter 8.
CHAPTER 2

ALGEBRAIC RESULTS UTILIZED IN THE THESIS

2.1 Number Theory

Let \( a, b, c, d, r \) be integers and \( Q \) be a positive integer. Let \( a, \beta, \delta \) be integer \( m \)-vectors and \( Z \) be an integer \( m \)-vector with positive components \( z_i \).

Definition 2.1.1

\[ a \equiv b \pmod{Q} \text{ if and only if } a = b + kQ \text{ for some integer } k. \]

Definition 2.1.2

\[ a \equiv \beta \pmod{Q} \text{ if and only if } a_i \equiv \beta_i \pmod{Q} \text{ for } i = 1, \ldots, m. \]

Definition 2.1.3

\[ a \equiv \beta \pmod{Z} \text{ if and only if } a_i \equiv \beta_i \pmod{z_i} \text{ for } i = 1, \ldots, m. \]

Definition 2.1.4

\( \hat{c} = c \text{ reduced modulo } Q \) if and only if

\( \hat{c} \equiv c \pmod{Q} \) and \( 0 \leq \hat{c} < Q. \)

Definition 2.1.5

\( \hat{a} = a \text{ reduced modulo } Q \) if and only if

\( \hat{a} \equiv a \pmod{Q} \) and \( 0 \leq \hat{a}_i < Q. \)
Definition 2.1.6
\[ \hat{\alpha} = \alpha \text{ reduced modulo } Z \text{ if and only if} \]
\[ \hat{\alpha} \equiv \alpha \pmod{Z} \text{ and } 0 \leq \hat{\alpha} < z_i. \]

Definition 2.1.7
\[ r|Q \iff Q = rk \text{ for some integer } k. \]

Definition 2.1.8
The greatest common factor of \( a \) and \( Q \), denoted by \((a,Q)\), is the unique integer \( r \) such that \( a = rk_1 \) for some integer \( k_1 \) and \( Q = rk_2 \) for some integer \( k_2 \) where \( k_1 \) and \( k_2 \) are relatively prime.

Definition 2.1.9
A solution to a system of linear congruences given by \( Ax \equiv \alpha \pmod{Q} \) is an integer vector \( \overline{x} \) such that \( A\overline{x} \equiv \alpha \pmod{Q} \).

Definition 2.1.10
A root of a system of linear congruences given by \( Ax \equiv \alpha \pmod{Q} \) is a solution \( \overline{x} \) such that \( 0 \leq x_i < Q \).

Definition 2.1.11
Two systems of linear congruences are equivalent if and only if either they have exactly the same solutions or both systems are unsolvable.

Proposition 2.1.0
Given a system of linear congruences \( Ax \equiv \alpha \pmod{Q} \); form \( C = [A\alpha] \). Then any element \( c_{ij} \) of \( C \) may be replaced by \( \hat{c}_{ij} \equiv c_{ij} \pmod{Q} \) and the resulting system of congruences is equivalent to the original one.
Proposition 2.1.1

\[ a \equiv b \pmod{Q} \text{ and } b \equiv c \pmod{Q} \rightarrow a \equiv c \pmod{Q}. \]

Proposition 2.1.2

\[ a \equiv b \pmod{Q} \text{ and } c \equiv d \pmod{Q} \rightarrow a + c \equiv b + d \pmod{Q}. \]

Proposition 2.1.3

\[ a \equiv b \pmod{Q} \text{ and } c \equiv d \pmod{Q} \rightarrow ac \equiv bd \pmod{Q}. \]

Proposition 2.1.4

\[ a \equiv b \pmod{Q} \rightarrow ac \equiv bc \pmod{Qc}. \]

Proposition 2.1.5

\[ ar \equiv br \pmod{Q} \text{ and } (r, Q) = d \rightarrow a \equiv b \pmod{Q/d}. \]

Proposition 2.1.6

The congruence \( ax \equiv b \pmod{Q} \) is solvable for \( x \) if and only if \( (a, Q) \mid b \). If the congruence is solvable there are exactly \( (a, Q) \) roots.

Proposition 2.1.7

Let the linear congruences \( L \) be given by

\[
\sum_{j=1}^{n} a_{i j} x_{j} \equiv b_{i} \pmod{Q}.
\]

Then the system

\[
\begin{align*}
L_{i} & \equiv 0 \pmod{Q} \quad i = 1, \ldots, m \\
L & = 0 \pmod{Q} 
\end{align*}
\]

is equivalent to the system

\[
\begin{align*}
L_{i} & \equiv 0 \pmod{Q} \quad i = 1, \ldots, j-1, j+1, \ldots, m \\
k \cdot L_{j} + \sum_{i=1, i \neq j}^{m} k \cdot L_{i} & \equiv 0 \pmod{Q}
\end{align*}
\]

where \( (k_{i}, Q) = 1 \) and \( k_{i} \) is any integer \( (i = 1, \ldots, m, i \neq j) \).

This proposition implies that ordinary row operations may be performed on a system of linear congruences without changing any solutions.
2.2 Group Theory

Definition 2.2.1

A finite Abelian group is a finite set $S$ with operation $+$ that satisfies the properties of closure, associativity, commutivity, identity and inverse.

Definition 2.2.2

The number of elements of a group $G$ is called the order of the group and is denoted by $|G|$.

Let $Z$ be an integer $m$-vector with positive components, $Q$ be a positive integer and $I_m$ be the $(m) \times (m)$ identity matrix.

Proposition 2.2.1

The set of all integers under the operation of addition modulo $Q$ is a finite Abelian group of order $Q$.

Proposition 2.2.2

The set of all integer $m$-vectors under the operation of addition modulo $Z = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix}$ forms a finite Abelian group of order $\prod_{i=1}^{m} z_i$. Denote this group by $G(I, Z)$.

Proposition 2.2.3

Let $A$ be an $(m) \times (n)$ integer matrix. Then the set of all integer linear combinations of columns of $A$ forms a group under the operation of addition modulo $Z$. Denote this group by $G(A, Z)$. 
Proposition 2.2.4

G(A,Z) is a subgroup of G(I_m,Z).

Proposition 2.2.5

A system of linear congruences, Ax ≡ b (mod Z), where A is an integer matrix and b is an integer vector has a solution if and only if b is an element of G(A,Z).

Proposition 2.2.6

The order of a finite group is a multiple of the order of each of its subgroups.

2.3 Linear Algebra

Definition 2.3.1

A nonsingular integer matrix with integer inverse is called a unimodular matrix.

Proposition 2.3.1

Premultiplying a matrix by a unimodular matrix is equivalent to a series of elementary row operations over the ring of integers, that is, (1) adding an integer multiple of one row to another row, (2) interchanging rows and (3) multiplying a row by minus one.

Definition 2.3.2

An (n) x (n) matrix is upper triangular if and only if all entries below the main diagonal are zero.

Definition 2.3.3

Let Σ(n) denote the set of all (n) x (n) upper triangular, integer matrices with positive diagonal elements.
Theorem 2.3.1 (Hermite)

Given a nonsingular \((n) \times (n)\) integer matrix \(C\), there exists an \((n) \times (n)\) unimodular matrix \(K\) such that \(KC \in \mathbb{T}(n)\). Let \(y\) be any \(m\)-vector and \(k\) be a positive integer such that \(k \leq m\).

Definition 2.3.4

The vector \(y\) is said to be \(k\)-triangularized if and only if \(y_k \geq 0\) and \(y_j = 0\) for \(j > k\).

For the next two corollaries to Theorem 2.3.1 assume \(A = (m) \times (n)\) integer matrix with rank \(r\).

Corollary 2.3.2

\(A^i\) can be transformed to a \(k\)-triangularized vector \(A^i\) by a series of elementary row operations on rows \(k, \ldots, m\). Moreover \(a^i_{k,i}\) equals the greatest common factor of \((a_k,i, a_{k+1,i}, \ldots, a_m,i)\). Therefore \(a^i_{k,i} = 0\) if and only if \(a_k,i = a_{k+1,i} = \ldots = a_m,i = 0\).

Corollary 2.3.3

If \(S = \{s_1, \ldots, s_k\}\) is a sequence of \(k\) distinct integers from the set \(J^n\) and if

\[A^S = [H(k)]^\theta\]

where \(H(k) \in \mathbb{T}(k)\) and \(\theta = (m-k) \times (k)\) matrix of zeros

then either:

1) \(A^i = \theta\) for \(i = k+1, \ldots, m\) and rank of \(A = k\) or

2) There exists \(j \in J^n, j \notin S\) such that \(A^j\) can be transformed by a series of elementary row operations on rows \(k+1, \ldots, m\) to a \((k+1)\)-triangularized vector \(A^j\) with \(a_{k+1,j} > 0\).
Definition 2.3.5

A sequence \( S = (s_1, s_2, \ldots, s_p) \) of distinct integers is in natural order if and only if \( s_1 < s_2 < \ldots < s_p \).

Theorem 2.3.4 (Smith)

Given a nonsingular \((n) \times (n)\) integer matrix \( C \), there exist \((n) \times (n)\) unimodular matrices \( E, F \) such that \( D = ECF \) is a diagonal matrix with positive diagonal elements such that \( d_{11} |d_{22}| \ldots |d_{nn} | \).

Corollary 2.3.5

The determinant of \( C = \prod_{i=1}^{n} d_{ii} \).

Theorem 2.3.6

Given a system of consistent linear equations

\[
Ax = b
\]

\( x \) integer

where \( A \) is an \((m) \times (n)\) integer matrix of rank \( m \) and \( d \) is any solution to the system, then there exists an \((n) \times (n - m)\) integer matrix \( E \) with full column rank such that the set of all solutions is given by

\[
x = d + Ey, \quad y \text{ integer}.
\]
CHAPTER 3

A NEW EQUIVALENT REPRESENTATION OF THE ILPC

3.1 Purpose of the Representation

As stated previously, the ILPC can be represented as a problem of determining which root of the system given by:

\[ Ax \equiv b \pmod{Q} \]  \hspace{1cm} 3.1.1

minimizes a linear function with non-negative coefficients. It is the purpose of this chapter to present an algorithm for deriving a system of linear congruences which is equivalent to the system given by 3.1.1. That is, both systems either have exactly the same solutions or both are unsolvable. From the new representation it can be determined exactly how many, if any, roots to system 3.1.1 exist and further, all roots can be immediately calculated.

3.2 Summary of the Theorems

The main result of the chapter may be summarized by the following theorem:

**Triangularization Theorem:**

Any system of linear congruences given by:

\[ Ax \equiv b \pmod{Q} \]

\[(A \neq 0)\]
is either not solvable or is equivalent to a system given by:
\[ Rx \equiv d \pmod{Q} \]
where \( R = (p) \times (n) \) non-negative integer matrix and 
\( d = \) non-negative integer p-vector \((p \leq n)\).

Moreover there exists a naturally ordered sequence 
\( S = (s_1, \ldots, s_p) \) of \( p \) distinct integers from \( J_n \) with the following properties:

Let \( D = R^S \)

\[ T = \text{the naturally ordered sequence consisting of integers in } J_n \text{ but not in } S \]

\[ K = R^T \]

\[ M = \sum_{i=1}^{p} d \]  
for \( i \) ii

Then 1) \( D \) is a \((p) \times (p)\) upper triangular matrix with positive diagonal elements.

2) \( d_{ii} | Q \) for \( i = 1, \ldots, p \).

3) \( 0 \leq d_{ij} < d_{jj} \) \((1 \leq i < j \leq p)\).

4) For any given integer vector \( x^T \), the system of congruences in \( x^S \) given by
\[ Dx^S + Kx^T \equiv d \pmod{Q} \]
is solvable for \( x^S \) and has exactly \( M \) roots.

The proofs of the above theorem contain procedures for construction of all matrices, vectors and sequences whose existence is asserted. These procedures will also indicate whether the original system of congruences is solvable.
3.3 Preliminary Analysis

Given any system of linear congruences

\[ Ax \equiv b \pmod{Q} \]

we can represent the system by the matrix \( C = [Ab] \) and
the modulus \( Q \). By proposition 2.1.0 we can reduce any
element of \( C \) modulo \( Q \) without changing the solution of the
system. In all of the matrix computations that follow in
this chapter, it will be assumed that such a substitution
is automatic.

**Definition 3.3.1**

Let \( T(k) \) denote the set of all \((k) \times (k)\) upper
triangular, non-negative integer matrices with positive
diagonal elements.

If we assume there exists a sequence \( S = (s_1, \ldots, s_{k-1}) \)
of \((k-1)\) distinct integers from the set \( J_n \) such that

\[ C^S = \begin{bmatrix} V \\ \Theta \end{bmatrix} \text{ where } V \in T(k-1) \]

and

\[ \Theta = (m-k+1) \times (k-1) \text{ matrix of zeros} \]

then by Cor. 2.3.3 either

1) \( A_i = \Theta \) for \( i = k, \ldots, m \) and rank \( A = r = k-1 \)

or

2) There is \( j \in J_n \), \( j \notin S \) such that \( A^j \) can be transformed
by a series of elementary row operations on rows \( k, \ldots, m \)
to a \( k \)-triangularized vector \( \hat{A}^j \) with \( \hat{a}_{k,j} > 0 \).

If condition 2) occurs then adjoining \( s_k = j \) to the sequence
\( S \) makes \( S \) a sequence of \( k \) distinct elements from \( J_n \). The
process of transforming \( A^j \) to a \( k \)-triangularized vector
also transforms \( C \) to a matrix \( \hat{C} \) by a series of elementary row operations. By Prop. 2.1.7, the system of congruences represented by \( \hat{C} \) and \( Q \) is equivalent to the original system. The matrix \( \hat{C} \) also has the property that

\[
\hat{C}^S = \begin{bmatrix} \hat{V} \\ \theta \end{bmatrix}
\]

where \( \hat{V} \in T(k) \)

and

\[
\theta = (m-k) \times (k) \text{ matrix of zeros.}
\]

This procedure may be iteratively continued until eventually condition 1) must hold. When condition 1) occurs we have transformed the original system of congruences to an equivalent system

\[
Px \equiv d \pmod{Q}.
\]

The sequence \( S \) has \( r \) distinct elements.

Define \( T = \) the naturally ordered sequence of all integers in \( J_n \) not in \( S \).

Then \( P^S = \begin{bmatrix} D \\ \theta \end{bmatrix} \) where \( D \in T(r) \)

and

\[
\theta = (m-r) \times (r) \text{ matrix of zeros}
\]

\[
p^T = \begin{bmatrix} K \\ \theta \end{bmatrix} \text{ where } \begin{cases} K = (r) \times (n-r) \text{ integer matrix} \\ \theta = (m-r) \times (n-r) \text{ matrix of zeros} \end{cases}
\]

and

\[
d = \begin{bmatrix} d^1 \\ d^2 \end{bmatrix} \text{ where } \begin{cases} d^1 = \text{ integer } r\text{-vector} \\ d^2 = \text{ integer } (m-r)\text{-vector} \end{cases}
\]

The system of congruences

\[
Px \equiv d \pmod{Q}
\]

is therefore solvable only if \( d^2 = 0 \). Since this system is equivalent to the original system, the condition that \( d^2 = 0 \) is a necessary condition for solvability of the
system \( Ax \equiv b \pmod{Q} \).

However \( d^2 = \theta \) is not a sufficient condition to guarantee that for any choice of the integer vector \( x^T \) the system
\[
Dx^S + Kx^T \equiv d^1 \pmod{Q}
\]
is solvable for \( x^S \).

It is the purpose of the next section to define a new row constraint for each \( s \in S \). Each constraint is adjoined to the system of congruences without changing the number of variables \( n \), the solution set or the triangularization process. It may change the rank of the constraint matrix and thus the number of elements in the sequences \( S \) and \( T \).

The new constraints collectively imply upon termination of the triangularization process with an equivalent system
\[
P x \equiv \begin{bmatrix} \bar{d}^1 \\ \bar{d}^2 \end{bmatrix} \pmod{Q}
\]
and sequences \( S = (s_1, \ldots, s_p) \) and \( T = (t_1, \ldots, t_{n-p}) \) that \( d^2 = \theta \) is both necessary and sufficient for solvability of the system
\[
Ax \equiv b \pmod{Q}.
\]

It will also be true that if \( d^2 = \theta \), then for any integer vector \( x^T \) the system
\[
pS_x^S + p^T x^T \equiv d \pmod{Q}
\]
has at least one solution \( x^S \) and if the columns of \( A \) are appropriately rearranged then \( \begin{bmatrix} x^S \\ x^T \end{bmatrix} \) solves the system
\[
Ax \equiv b \pmod{Q}.
\]

If the iterative procedure described in this section begins with \( S = \emptyset, T = J_n \) and at each step we always choose
the index \( j \) to be the smallest integer in \( T \) such that \( A^j \)
can be transformed to the desired vector, then it is
easily seen that \( S \) will be a sequence in natural order.

3.4 Preliminary Definition of the Adjoined Constraint

**Theorem 3.4.1**

Let \( \sum_{j=k}^{n} a_j x_j \equiv d \pmod{Q} \) \hspace{1cm} 3.4.2

be a congruence such that \( a_j, (j = k, \ldots, n) \) and \( d \) are
integers and \( a_k > 0 \).

Define \( r = (a_k, Q) \) and \( s = Q/r \) and a congruence
\( \sum_{j=k+1}^{n} (sa_j)x_j \equiv (sd) \pmod{Q} \) \hspace{1cm} 3.4.3

then

1) if \((\overline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)\) is any solution to 3.4.2,
\((\overline{x}_{k+1}, \ldots, \overline{x}_n)\) solves 3.4.3

and

2) if \((\overline{x}_{k+1}, \ldots, \overline{x}_n)\) is any solution to 3.4.3, there
exists an integer \( \overline{x}_k \) such that \((\overline{x}_k, \overline{x}_{k+1}, \ldots, \overline{x}_n)\) solves
3.4.2.

Each time a new element \( s_k \) is adjoined to the sequence \( S \), a
constraint such as 3.4.3 is derived from the \( k^{th} \) row of the
current system of congruences and adjoined to that system.

3.5 Triangularization Process

Initialize \( S = \phi, T = J_n, j = 1, k = 1. \)

Form \( C = [A_b] \)
1) Transform C to \( \hat{C} \) by the series of elementary row operations necessary to transform \( C^j \) to a k-triangularized vector. Reduce the elements of \( \hat{C} \) modulo Q. If \( \hat{c}_{k,j} = 0 \) go to 4, else continue.

2) Set \( s_k = j \) and remove j from the sequence T

3) Add a new row to \( \hat{C} \) defined as follows:
   
   Let \( r = Q/(\hat{c}_{k,j}, Q) \)
   
   \( \hat{c}_{m+k,i} = 0 \) for \( i \in S \)
   
   \( \hat{c}_{m+k,i} = r\hat{c}_{k,i} \) for \( i \in T \)
   
   \( \hat{c}_{m+k,n+1} = r\hat{c}_{k,n+1} \)
   
   Reduce \( \hat{c}_{m+k,i} \) modulo Q for \( i \in T \) and \( i = n + 1 \).

4) Denote \( \hat{C} \) by C

5) Set \( j = j + 1 \). If \( j = n + 1 \), go to 8. Otherwise go to 6.

6) If \( c_{ij} = 0 \) for \( i = (k+1), \ldots, (m+k) \) go to 5. Else go to 7.

7) Set \( k = k + 1 \) and go to 1)

8) Set \( p = k \) and stop.

3.6 Explanation of the Triangularization Process

An iteration of the Triangularization Process is defined to be the execution of steps 1-4. During each iteration \( k \) of the process, the following three things are accomplished.

1) Some column \( C^j \) is transformed to a k-triangularized vector \( \hat{C}^j \) such that \( 0 < \hat{c}_{kj} < Q \).

2) The column index \( j \) is adjoined to the sequence \( S \) and removed from \( T \)
3) A constraint is derived from row \( k \) corresponding to the congruence defined in Theorem 3.4.1 and the new constraint is adjoined to the system of congruences.

At the end of the \( k \text{th} \) iteration, the transformed system of congruences, represented by \( \hat{C} \), will consist of \((m+k)\) rows and \((n+1)\) columns. The sequence \( S \) will consist of \( k \) distinct integers from the set \( J_n \). Also

\[
\hat{C}^S = \begin{bmatrix} V \end{bmatrix} \quad \text{where} \quad \begin{cases} V \in T(k) \\ \theta = (m) \times (k) \text{ matrix of zeros} \end{cases}
\]

\[
\hat{C}^T = \begin{bmatrix} W \\ U \end{bmatrix} \quad \text{where} \quad \begin{cases} W = (k) \times (n-k) \text{ integer matrix} \\ U = (m) \times (n-k) \text{ integer matrix} \end{cases}
\]

\[
\hat{C}^{n+1} = \begin{bmatrix} e_1^{1} \\ e_2^{2} \end{bmatrix} \quad \text{where} \quad \begin{cases} e_1^{1} = \text{ integer k-vector} \\ e_2^{2} = \text{ integer m-vector} \end{cases}
\]

The index \( j \) is always chosen as the smallest integer from the naturally ordered set \( T \) such that \( C_j \) can be transformed to the desired \( k \)-triangularized vector. Thus for any \( i \in T \) with \( i < j \), \( c_{h,i} = 0 \) for all \( h > k \) and further row operations will leave column \( i \) unchanged. Therefore the index \( i \) will never be a candidate for the sequence \( S \) and the sequences \( S \) and \( T \) will remain in natural order.

The procedure terminates when at some step \( k \) no column \( C_j(j \in T) \) can be transformed to a \( k \)-triangularized vector with \( \hat{c}_{k,j} > 0 \). This can only occur if \( T = \emptyset \) or (by Cor. 2.3.3) if \( c_{i,j} = 0 \) for \( i = k, \ldots, m + k - 1 \) and for all \( j \in T \). Since \( n \) remains invariant throughout the process, termination must occur after \( p \) steps where \( p \leq n \).
Upon termination we have:

1) $S$ consists of $p$ naturally ordered, distinct integers and $T$ consists of $(n-p)$ naturally ordered, distinct integers. (It is, of course, possible that $p = n$ and $T = \emptyset$).

and

2) $\hat{C}^S = \begin{bmatrix} V \\ \theta \end{bmatrix}$ where $\begin{cases} V_{\theta} = T(p) \\ \theta = (m) \times (p) \text{ matrix of zeros} \end{cases}$

$\hat{C}^T = \begin{bmatrix} W \\ \theta \end{bmatrix}$ where $\begin{cases} W_{\theta} = (p) \times (n-p) \text{ integer matrix} \\ \theta = (m) \times (n-p) \text{ matrix of zeros} \end{cases}$

$\hat{C}^{n+1} = \begin{bmatrix} \hat{e} \\ \hat{e} \end{bmatrix}$ where $\begin{cases} \hat{e} = \text{ integer } p\text{-vector} \\ \hat{e} = \text{ integer } m\text{-vector} \end{cases}$

The following theorem asserts that upon termination of the Triangularization process the solution set remains intact and the condition $\hat{e} = 0$ is a necessary and sufficient one for solvability of the system.

**Lemma 3.6.1**

Let $C = [Ab]$ represent any system of congruences (mod $Q$) to which the Triangularization Process is applied. At the end of the $k^{th}$ iteration let $\hat{C}$ represent the transformed system of congruences with sequences $\hat{S}$ and $\hat{T}$ such that

$\hat{C}^S = \begin{bmatrix} \hat{V} \\ \theta \end{bmatrix}$ where $\begin{cases} \hat{V}_{\theta} = T(k) \\ \theta = (m) \times (k) \text{ matrix of zeros} \end{cases}$

$\hat{C}^T = \begin{bmatrix} \hat{W} \\ \hat{U} \end{bmatrix}$ where $\begin{cases} \hat{W} = (k) \times (n-k) \text{ integer matrix} \\ \hat{U} = (m) \times (n-k) \text{ integer matrix} \end{cases}$

Then

1) the systems represented by $C$ and $\hat{C}$ are equivalent (mod $Q$)
and 2) if there exists any integer vector $\vec{x}^T$ satisfying the system of congruences given by rows $k+1, \ldots, k+m$ of $\hat{C}$, then there exists at least one integer vector $\vec{x}$ satisfying the system represented by $C$.

**Theorem 3.6.2**

Let $C = [Ab]$ represent any system of congruences (mod $Q$) to which the Triangularization Process is applied. Upon termination of the process at iteration $p$, let $\hat{C}$ represent the transformed system of congruences with sequences $S$ and $T$ such that

\[
\hat{C}^S = \begin{bmatrix} V \\ \emptyset \end{bmatrix} \text{ where } \begin{cases} V_T(p) \\ \emptyset = (m) \times (p) \text{ matrix of zeros} \end{cases}
\]

\[
\hat{C}^T = \begin{bmatrix} W \\ \emptyset \end{bmatrix} \text{ where } \begin{cases} W = (p) \times (n-p) \text{ integer matrix} \\ \emptyset = (m) \times (n-p) \text{ matrix of zeros} \end{cases}
\]

\[
\hat{C}^{n+1} = \begin{bmatrix} \vec{e} \\ \vec{\hat{e}} \end{bmatrix} \text{ where } \begin{cases} \vec{e} = \text{ integer p-vector} \\ \vec{\hat{e}} = \text{ integer m-vector} \end{cases}
\]

Then

1) the systems represented by $C$ and $\hat{C}$ are equivalent (mod $Q$)

2) $\vec{\hat{e}} \neq \emptyset \rightarrow$ the system represented by $C$ is not solvable (mod $Q$)

and

3) $\vec{\hat{e}} = 0 \rightarrow$ for any integer vector $\vec{x}^T$, there exists at least one integer vector $\vec{x}$ satisfying the system represented by $C$.

Summarizing these results we state:

1) The system of linear congruences

$Ax \equiv b \pmod{Q}$
is solvable if and only if $\hat{e} = 0$

2) Any vector $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ solves the system

$$Ax \equiv b \pmod{Q}$$

if and only if the corresponding vectors $\overline{x}^S$ and $\overline{x}^T$ solve the system

$$V\overline{x}^S + W\overline{x}^T \equiv e \pmod{Q}$$

3) If $\hat{e} = 0$, then for any integer vector $\overline{x}^T$, there exists an integer vector $\overline{x}^S$ such that the corresponding vector $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ solves the system

$$Ax \equiv b \pmod{Q}.$$
and

2) for each solution $\bar{x}_s^p$ to 3.7.2, there exists at least one integer vector $(\bar{x}_{s_1}^s, \ldots, \bar{x}_{s_{p-1}}^s)$ such that $\bar{x}^S = (\bar{x}_{s_1}^s, \ldots, \bar{x}_{s_{p-1}}^s)$ solves 3.7.1.

In general one may state these properties as follows:

Let $\bar{x}_{s_{k+1}}^s, \ldots, \bar{x}_{s_p}^s$ constitute any solution to congruences $k + 1, \ldots, p \ (1 < k+1 \leq p)$ of 3.7.1.

Then the $k^{th}$ congruence is given by:

$$v_{kk} \bar{x}_s^k + \sum_{j=k+1}^{p} v_{kj} \bar{x}_s^j \equiv g_k \pmod{Q} \quad 3.7.3$$

and the triangularization process guarantees

1) $(v_{kk}, Q) | (g_k - \sum_{j=k+1}^{p} v_{kj} \bar{x}_s^j)$

and

2) for each integer $\bar{x}_{s_k}^s$ satisfying 3.7.3, there exists at least one integer vector $(\bar{x}_{s_1}^s, \ldots, \bar{x}_{s_{k-1}}^s)$ such that $\bar{x}^S = (\bar{x}_{s_1}^s, \ldots, \bar{x}_{s_{k-1}}^s)$ satisfies 3.7.1.

Although 1) guarantees the solvability of the $k^{th}$ congruence for $x_s^s$, the actual calculation of all solutions $x_{s_k}^s$ can be a computational inconvenience.

Consider the following 2 examples:

$$7x \equiv 1 \pmod{11}$$

$$2x \equiv 4 \pmod{6}$$

By Prop. 2.1.6, both examples are solvable for $x$. Some computations are needed to solve the first one, but the solutions to the second one are immediate since the coefficient of $x$ divides the modulus 6 and must therefore
divide the right hand side 4. Similarly if each diagonal element \( v_{kk} \) divides \( Q \), then solving the \( k^{th} \) congruence for \( x_{s_k} \) is a simple process. The significance of the following theorem is that it enables all solutions to the original system to be easily calculated.

**Theorem 3.7.4**

Given the system

\[
Vx^S \equiv e - Wx^t \pmod{Q} \quad 3.7.5
\]

where \( V \in T(p) \) and for any integer vector \( x^t \) the system is solvable for \( x^S \), then there exists an integer vector \( u = (u_1 \dots u_p) \) with \( u_i > 0 \) (i = 1, ..., p) and defining:

\[
D'_i = u_i V_i \quad \text{reduced modulo } Q \]
\[
d'_i = u_i e_i \quad \text{reduced modulo } Q \]
\[
K'_i = u_i W_i \quad \text{reduced modulo } Q
\]

\( i = 1, \ldots, p \)

gives the system

\[
D'x^S \equiv d' - K'x^t \pmod{Q} \quad 3.7.6
\]

with the properties that:

1) \( D' \in T(p) \)

2) \( d'_{jj} \mid Q \) (j = 1, ..., p)

3) the system given by 3.7.5 is equivalent to the system given by 3.7.6.

The following corollary simply asserts that the off-diagonal elements of \( D' \) can be reduced without disturbing the system.
Corollary 3.7.7

Given system 3.7.6 there exists a system

\[ Dx^S \equiv d - Kx^T \pmod{Q} \]  

3.7.8

with the properties that:

1) \( D \varepsilon T(p) \)

2) \( d_{jj} \mid Q \quad (j = 1, \ldots, p) \)

3) \( 0 \leq d_{ij} < d_{jj} \quad (1 \leq i < j \leq p) \)

4) system 3.7.6 is equivalent to system 3.7.8.

3.8 Equivalence Algorithm

We may now state the main algorithm of this chapter which transforms the original system of congruences to an equivalent system with the properties described by the Triangularization Theorem.

Let \( Ax \equiv b \pmod{Q} \) be any nontrivial system of congruences.

1) Form \( C = [Ab] \).

2) Apply the Triangularization Process for \( p \) iterations, determining an equivalent system \( \hat{C} \) with naturally ordered sequences \( S \) and \( T \) such that

\[
\hat{C}^S = \begin{bmatrix} V \end{bmatrix} \text{ where } \begin{cases} V_{\varepsilon T(p)} \\ \theta = (m) \times (p) \text{ matrix of zeros} \end{cases}
\]

\[
\hat{C}^T = \begin{bmatrix} W \end{bmatrix} \text{ where } \begin{cases} W = (p) \times (n-p) \text{ integer matrix} \\ \theta = (m) \times (n-p) \text{ matrix of zeros.} \end{cases}
\]

\[
\hat{C}^{n+1} = \begin{bmatrix} e \end{bmatrix} \text{ where } \begin{cases} e = \text{ integer p-vector} \\ \hat{e} = \text{ integer m-vector} \end{cases}
\]

3) If \( \hat{e} \neq 0 \), stop. Else go to 4.
4) Determine the integer p-vector \( u \) by solving the p congruences
\[
 v_k, k^u_k \equiv (v_{k}, k')Q \pmod{Q}
\]
for any root \( u_k \) \((k = 1, \ldots, p)\).

5) Multiply the \( k \)th row of the system represented by [VWe] by the integer \( u_k \) \((k = 1, \ldots, p)\).

6) Reduce all elements modulo \( Q \) and reduce the off-diagonal elements so the diagonal element is the largest integer in its column.

7) Denote the new system by [DKd]
where \( D \in T(p) \)
\[
K = (p) \times (n-p) \text{ integer matrix}
\]
and \( d = \text{integer p-vector} \)

8) Define a \((p) \times (n)\) matrix \( R \) such that
\[
R^S = D
\]
\[
R^T = K
\]

From the previous theorems we see that the original system is solvable if and only if the algorithm does not terminate at step 3. At step 8 we have the original system equivalent to the system
\[
Rx \equiv d \pmod{Q}
\]
with the properties that:
\[
R^S = D
\]
\[
R^T = K
\]

Moreover the matrix \( D \) is a \((p) \times (p)\) upper triangular matrix with positive diagonal elements \( d_{ii} \) such that
\[
d_{ii} |Q \text{ and } 0 \leq d_{ij} < d_{jj} \ (1 \leq i < j < p).
\]
Finally for any given integer vector $\bar{x}^T$, the system of congruences in $x^S$ given by

$$Dx^S + K\bar{x}^T \equiv d \pmod{Q}$$

has at least one solution.

3.8.1

It is important to note that neither the integer $p$, the sequences $S$ and $T$ nor the matrices $D$ and $K$ are unique. They all depend on the criteria by which column indices of $A$ are chosen to be candidates for the sequence $S$. The criterion employed by the Triangularization Process of Section 3.5 was selected simply to preserve the natural ordering of the sequences $S$ and $T$. For optimization purposes it will be assumed that the columns of $A$ and the variables $x$ are ordered such that $i < j \Leftrightarrow c_i \leq c_j$ where $c$ is the vector of non-negative objective function coefficients. Then since $S$ and $T$ are naturally ordered we have that:

1) $i < j \Leftrightarrow s_i < s_j \Leftrightarrow c_{s_i} \leq c_{s_j}$

and

2) $i < j \Leftrightarrow t_i < t_j \Leftrightarrow c_{t_i} \leq c_{t_j}$

The reason for trying to restrict $x^S$ to the set of "cheapest" variables is to get as good an initial feasible solution as possible by setting $x^T$, the vector of "more expensive" variables, equal to zero. The importance of this criterion will be illustrated in the chapters on optimization.
3.9 Number of Roots of a System of Congruences

In the introduction to Chapter 3, the question was posed how many roots exist to a system given by
\[ Ax \equiv b \pmod{Q}. \]

From the Triangularization Theorem we see that if the system is solvable it is equivalent to the system
\[ Rx \equiv d \pmod{Q} \]
where \( R^S = D \)
and \( R^T = K. \)

The following lemma and theorem complete the proof of the Triangularization Theorem by stating exactly how many roots a given system has.

Lemma 3.9.1

Let:
\[ Dx^S + Kx^T \equiv d \pmod{Q} \]
represent the system of congruences at step 7 of the Equivalence Algorithm.

Let the determinant of \( D = \prod_{i=1}^{n} i, i \) be denoted by \( M. \)

Then for any given integer vector \( x^T \) the system of congruences in \( x^S \) given by
\[ Dx^S + Kx^T \equiv d \pmod{Q} \]
has exactly \( M \) roots.

Theorem 3.9.2

If the system of congruences
\[ Ax \equiv b \pmod{Q} \]
is solvable, then it has exactly
MQ^{n-p} distinct roots.

The proof of the Triangularization Theorem as stated in Section 3.2 is now complete.

3.10 Simplifying Assumption

Although the sequences S and T are naturally ordered, it is not necessarily true that $S = (1, 2, \ldots, p)$ and $T = (p+1, \ldots, n)$. The following lemma asserts that if a column index $j$ is not adjoined to the sequence $S$, but a later column index $i (i > j)$ is adjoined to $S$, then changes in the variable $x_j$ have no effect on the variable $x_i$.

Lemma 3.10.1

If at step 8 of the Equivalence Algorithm (with a matrix $R$ and sequences $S$ and $T$) there exists some $j \in T$ and $i \in S$ such that $i > j$, then $R_{i,j} = 0$.

This lemma is utilized in Chapter 6 to show the finiteness of the Primal-Dual Algorithm.

In most of what follows, however, there is no loss of generality in assuming that $S = (1, 2, \ldots, p)$ and $T = (p+1, \ldots, n)$. Since this ordering greatly simplifies notation, it will be assumed throughout. All cases for which this simplifying assumption jeopardizes generality will be noted.

We may now assume that if a system of congruences:

$$Ax \equiv b \pmod{Q}$$

is solvable, then it is equivalent to a system:

$$Dx^S + Kx^T \equiv d \pmod{Q}$$
where the matrix $D$ and the system of congruences have all the properties as stated in the Triangularization Theorem.

3.1.1 Proof of Theorems

Theorem 3.4.1

1) Let $(\bar{x}_k, \ldots, \bar{x}_n)$ solve 3.4.2 →

$$\sum_{j=k}^{n} a_j \bar{x}_j \equiv d \pmod{Q} \rightarrow \text{(by Prop. 2.1.3)}$$

$$\sum_{j=k}^{n} (s a_j) \bar{x}_j \equiv (sd) \pmod{Q}$$

But

$$s a_k = \frac{Q a_k}{a_k, Q} \equiv 0 \pmod{Q} \rightarrow \text{(by Prop. 2.1.0)}$$

$$\sum_{j=k+1}^{n} (s a_j) \bar{x}_j \equiv (sd) \pmod{Q} \rightarrow$$

$$(\bar{x}_{k+1}, \ldots, \bar{x}_n) \text{ solves 3.4.3}$$

2) Let $(\bar{x}_{k+1}, \ldots, \bar{x}_n)$ solve 3.4.3 →

$$\sum_{j=k+1}^{n} (s a_j) \bar{x}_j \equiv sd \pmod{Q} \rightarrow \text{(by Prop. 2.1.5)}$$

$$\sum_{j=k+1}^{n} a_j \bar{x}_j \equiv d \pmod{r} \rightarrow$$

$$0 \equiv d - \sum_{j=k+1}^{n} a_j \bar{x}_j \pmod{r} \rightarrow \text{(by def. 2.1.1)}$$

$$d - \sum_{j=k+1}^{n} a_j \bar{x}_j = mr \text{ for some integer } m \rightarrow \text{(by def. 2.1.7)}$$

$$r | (d - \sum_{j=k+1}^{n} a_j \bar{x}_j) \rightarrow \text{(by Prop. 2.1.6)}$$

$$a_k \bar{x}_k \equiv d - \sum_{j=k+1}^{n} a_j \bar{x}_j \pmod{Q} \text{ is solvable for } x_k \rightarrow$$
There exists an integer $\bar{x}_k$ such that $(\bar{x}_k, \bar{x}_{k+1}, \ldots, \bar{x}_n)$ solves \ref{3.4.2}

**Lemma 3.6.1**

The proof is by induction. Assume lemma is true for the first $(k-1)$ iterations (Proof for $k = 1$ is analogous).

Let $\bar{C}$ represent the transformed system of congruences at the end of the $(k-1)^{st}$ iteration with sequences $\bar{S}$ and $\bar{T}$ such that

$$\bar{C}^\bar{S} = \begin{bmatrix} V \\ \theta \end{bmatrix} \text{ where } \begin{cases} V \in \mathbb{T}(k-1) \\ \theta = (m) \times (k-1) \text{ matrix of zeros} \end{cases}$$

$$\bar{C}^\bar{T} = \begin{bmatrix} W \\ U \end{bmatrix} \text{ where } \begin{cases} W = (k-1) \times (n-k+1) \text{ integer matrix} \\ U = (m) \times (n-k+1) \text{ integer matrix} \end{cases}$$

Then the system of congruences represented by rows $k, \ldots, (m+k-1)$ of $\bar{C}$ involve only the variables in the vector $x^T$ and may be represented by the matrix

$$\bar{R} = \begin{bmatrix} \bar{C}_k \\ \vdots \\ \bar{C}_{m+k-1} \end{bmatrix}$$

By inductive hypothesis:

1) the systems represented by $C$ and $\bar{C}$ are equivalent (mod $Q$) and

2) if there exists any integer vector $\bar{x}^T$ satisfying the system represented by $\bar{R}$, then there exists at least one integer vector $\bar{x}$ satisfying the system represented by $C$.

**Proof:** If $U = \theta$, then the Triangularization Process is complete, $k - 1 = p$ and the theorem is proved. If $U \neq \theta$, then (by Cor. 2.3.3) there exists $j \in T$ such that $\bar{C}^\bar{j}$ can be transformed by a series of elementary row operations on
rows \( k, \ldots, m + k - 1 \) of \( \hat{C} \) to a \( k \)-triangularized vector \( \hat{C}^j \) with \( \hat{c}_{k,j} > 0 \). (For simplicity of notation we will assume \( j = \xi_1 \)). Let \( \hat{C} \) denote the transformed system of congruences reduced modulo \( Q \). By Prop. 2.1.7 the systems of congruences represented by \( \hat{C} \) and \( \hat{C} \) are equivalent (mod \( Q \)). Let the system of congruences given by rows \( k, \ldots, m + k - 1 \) of \( \hat{C} \) be represented by the matrix

\[
R = \begin{bmatrix}
\hat{c}_{k,k} \\
\vdots \\
\hat{c}_{m+k-1,k}
\end{bmatrix}
\]

By Prop. 2.1.7 the systems represented by \( \hat{R} \) and \( R \) are equivalent (mod \( Q \)) and involve only the variables in the vector \( x^T \).

Denote by \( \hat{S} \) the sequence formed by adjoining \( j \) to the sequence \( \hat{S} \) and by \( \hat{T} \) the sequence formed by removing \( j \) from the sequence \( T \).

Note that

\[
\hat{C}\hat{S} = \begin{bmatrix}
\hat{V} \\
\theta
\end{bmatrix}
\]

\[
\text{where} \begin{cases}
\hat{V}_{\xi T}(k) \\
\theta = (m-1) \times (k)
\end{cases}
\]

matrix of zeros.

Derive a new row constraint \( \hat{c}_{m+k} \) from \( \hat{c}_k \) in accordance with Theorem 3.4.1 (Step 3 of the Triangularization Process).

It is easily seen from Theorem 3.4.1 that the solution set of \( \hat{C} \) remains unchanged. Therefore the systems of congruences represented by \( C, \bar{C} \) and \( \hat{C} \) are all equivalent (mod \( Q \)) and part 1 of the lemma is proved.
We now have:
\[
\hat{C}^S = \begin{bmatrix} \hat{V} \\ \theta \end{bmatrix}
\]
where \(\hat{V} \in \mathbb{R}^{(k)}\) is a matrix of zeros

\[
\hat{C}^T = \begin{bmatrix} \hat{W} \\ U \end{bmatrix}
\]
where \(\hat{W} \in \mathbb{Z}^{(k)}\) is an integer matrix
\(U \in \mathbb{Z}^{(m)}\) is an integer matrix.

Let the system of congruences given by rows \(k+1, \ldots, m+k\) of \(\hat{C}\) be represented by the matrix
\[
\hat{R} = \begin{bmatrix} \hat{C}_{k+1} \\ \vdots \\ \hat{C}_{m+k} \end{bmatrix}
\]
and note that \(\hat{R}\) involves only the variables in the vector \(\hat{x}\).

By Theorem 3.4.1, for any integer vector \(\bar{x}^\hat{T}\) satisfying the single congruence given by \(\hat{C}_{m+k}\), there exists an integer \(\bar{x}_j\) such that the vector \(\bar{x}^\hat{T}\) with components \((\bar{x}_j, \bar{x}^\hat{T})\) satisfies the congruence represented by \(\hat{C}_k\).

Therefore for any integer vector \(\bar{x}^\hat{T}\) satisfying the system \(\hat{R}\), there exists an integer vector \(\bar{x}^\hat{T}\) satisfying the system \(R\). But \(R\) and \(\bar{R}\) are equivalent \((\text{mod } Q)\), so \(\bar{x}^\hat{T}\) also must satisfy the system \(\bar{R}\).

By inductive hypothesis 2) there exists at least one integer vector \(\bar{x}\) satisfying the system represented by \(C\) and part 2 of the lemma is proved.

**Theorem 3.6.2**

From Lemma 3.6.1 with \(k = p\), we see that \(C\) and \(\hat{C}\) are equivalent systems \((\text{mod } Q)\). Trivially the congruences given by rows \(p+1, \ldots, p+m\) of \(\hat{C}\) are solvable if and only if \(\hat{e} = 0\). Therefore if \(\hat{e} \neq 0\), the system represented by all rows of \(\hat{C}\) is not solvable \((\text{mod } Q)\) and \(C\) is not solvable \((\text{mod } Q)\).
If $\hat{e} = \theta$, then any integer vector $\vec{x}^T$ satisfies the trivial system of congruences given by rows $p + 1, \ldots, p + m$ of $\hat{C}$. Therefore by Lemma 3.6.1, for any integer vector $\vec{x}^T$, there exists at least one integer vector $\vec{y}$ satisfying the system represented by $C$ and the theorem is proved.

**Theorem 3.7.4**

Let $V_k x^S \equiv e_k - W_k x^T \pmod{Q}$

be the $k$th row ($k \leq p$) of the system

$V x^S \equiv e - W x^T \pmod{Q}$

and define

$d'_k, k = (v_{k,k}, Q)$.

Let $\alpha_k = v_{k,k} / d'_k, k$

$\beta_k = Q / d'_k, k$

then $(\alpha_k, \beta_k) = 1$

By Prop. 2.1.6 the congruence

$v_{kk} Z \equiv d'_k, k \pmod{Q}$ is solvable for $Z$.

Let $u_k$ be any positive solution, i.e.

$v_{k,k} u_k \equiv d'_k, k \pmod{Q}$

Before proving the theorem, we will first prove three lemmas.

**Lemma 1:** $(u_k, \beta_k) = 1$

**Proof:** Since $v_{k,k} u_k \equiv d'_k, k \pmod{Q}$

$(d'_k, k \alpha_k) u_k \equiv d'_k, k \pmod{d'_k, k \beta_k}$

(by prop. 2.1.5 and prop. 2.1.4)

$\alpha_k u_k \equiv 1 \pmod{\beta_k}$

the congruence

$u_k y \equiv 1 \pmod{\beta_k}$ is solvable
with solution $y = a_k \phi$ (by prop. 2.1.6)
$$(u_k, \beta_k)|1 \rightarrow (u_k, \beta_k) = 1$$
For the next 2 lemmas let $u, d, a, b, \alpha, \beta, g$ be integers and $u, d, \beta \geq 1$

**Lemma 2:** If $uda \equiv udb \pmod{d\beta}$
and $(u, \beta) = 1$
then $da \equiv db \pmod{d\beta}$

**Proof:** Since $(u, \beta) = 1 \rightarrow (ud, d\beta) = d \rightarrow \text{(by Prop. 2.1.5)}$
$a \equiv b \pmod{\beta} \rightarrow \text{(by prop. 2.1.4)}$
da \equiv db \pmod{d\beta}$.

**Lemma 3:**
\[
\begin{cases}
(d\alpha)x \equiv g \pmod{d\beta} \text{ is solvable for } x \\
(\alpha, \beta) = 1
\end{cases}
\]
If $u\alpha x \equiv ug \pmod{d\beta}$ for $x$ integer
$$(u, \beta) = 1$$
then $d\alpha x \equiv g \pmod{d\beta}$

**Note:** The necessity of all four conditions of the hypothesis can be seen by omitting any one and easily constructing counterexamples.

**Proof:** Since $(\alpha, \beta) = 1 \rightarrow (d\alpha, d\beta) = d$
Since $(d\alpha)x \equiv g \pmod{d\beta}$ is solvable for $x \rightarrow$
(by Prop. 2.1.6) $d|g \rightarrow \text{(by def. 2.1.7)}$
g = db for some integer $b \rightarrow$
u$d\alpha x \equiv udb \pmod{d\beta}$ for $x$ integer
Since $(u, \beta) = 1$ by hypothesis $\rightarrow \text{by Lemma 2}$
da$x \equiv db = g \pmod{d\beta}$. 
Proof of Theorem

Let \( u_k \) be as defined and

\[
D'_k = u_k V_k \quad \text{with} \quad v_{k,k} u_k \quad \text{replaced by} \quad d_{k,k}'
\]

(Since \( v_{k,k} u_k \equiv d_{k,k}' \) (mod Q), by Prop. 2.1.0 we can make this substitution without changing the set of solutions.)

The matrix \( D' \) now has properties (1) and (2).

To show property (3), let \( \overline{x}^T, \overline{x}^S \) be any solution to 3.7.5 → (by prop. 2.1.3)

\[
\overline{x}^S, \overline{x}^T \text{ is a solution to 3.7.6.}
\]

Let \( \overline{x}^S, \overline{x}^T \) be any solution to 3.7.6 and define

\[
g_k = (e_k - W_k \overline{x}^T) \quad (k = 1, \ldots, p).
\]

By hypothesis

\[
V_k \overline{x}^S \equiv g_k \quad \text{ (mod Q)}
\]

\[
k = 1, \ldots, p
\]

is solvable for \( x^S \). We wish to show that \( \overline{x}^S \) is a solution.

By assumption:

\[
u_k V_k \overline{x}^S \equiv u_k g_k \quad \text{(mod Q)}
\]

\[
\text{for } k = 1, \ldots, p
\]

Proof is by backwards induction starting with \( p \):

By hypothesis:

\[
V_{pp} \overline{x}_{sp} = \begin{bmatrix} d'_{pp} & \alpha_{pp} \end{bmatrix} \overline{x}_{sp} \equiv g_p \quad \text{(mod } d'_{pp}, \beta_p) \]

is solvable for \( x_{sp} \)

and \( (\alpha_p, \beta_p) = 1 \)

By assumption:

\[
u_p V_p \overline{x}_{sp} = u_p d'_{pp} \overline{x}_{sp} \equiv u_p g_p \quad \text{(mod } d'_{pp}, \beta_p, \overline{x}_{sp})
\]

for \( \overline{x}_{sp} \) integer.
By lemma 1:

\[(u_p, \beta_p) = 1 \rightarrow\]

(by Lemma 3)

\[v_{pp} \overline{x}_{sp} \equiv g_p \pmod{Q}.\]

Assume for any integer \(k\) such that \(1 < k < p\) that

\[(\overline{x}_{sk+1}, \ldots, \overline{x}_p)\]

solves

\[
\begin{cases}
V_i x^S \equiv g_i \pmod{Q} & \text{if } i = k + 1, \ldots, p
\end{cases}
\]

By hypothesis and Lemma 3.6.1:

\[v_{k,k} \overline{x}_{sk} \equiv g_k - \sum_{j=k+1}^{p} v_{k,j} \overline{x}_{sj} \pmod{Q}\]

is solvable for \(x_{sk}\) (where \(v_{k,k} = d_{k,k}^l\), \(Q = d_{k,k}^r\))

By definition:

\[(\alpha_k, \beta_k) = 1\]

By assumption:

\[u_k v_{k,k} \overline{x}_{sk} \equiv u_k \left[ g_k - \sum_{j=k+1}^{p} v_{k,j} \overline{x}_{sj} \right] \pmod{Q}\]

for \(\overline{x}_{sk}\) integer

By lemma 1:

\[(u_k, \beta_k) = 1\]

and therefore by lemma 3,

\[v_{k,k} \overline{x}_{sk} \equiv g_k - \sum_{j=k+1}^{p} v_{k,j} \overline{x}_{sj} \pmod{Q}\]

\[\rightarrow (\overline{x}_{sk}, \ldots, \overline{x}_p)\]

solves

\[V_i x^S \equiv g_i \pmod{Q} \quad i = k, \ldots, p\]
By induction \( \overline{x}^S \) solves \( Vx^S \equiv g \pmod{Q} \) and therefore systems 3.7.5 and 3.7.6 are equivalent.

**Corollary 3.7.7**

Form \( C = [D'K'd'] \) and let \( C^j \) be the \( j \)th column of \( C \). The first \( p \) columns of \( C \) can be sequentially reduced to the desired form starting with \( j = 2 \) as follows.

If \( c_{ij} \geq c_{jj} \) (\( i < j \)), then subtract as many multiples of row \( j \) from row \( i \) as necessary to make \( 0 \leq c_{ij}^j < c_{jj} \). Since \( c_{jk} = 0 \) for \( 1 \leq k \leq j - 1 \), this process does not alter columns \( 1, \ldots, j - 1 \). Neither the diagonal elements or triangularity of the first \( p \) columns is changed. Thus properties (1), (2) and (3) are proved. Since only elementary row operations are performed, the solution set also is unchanged and the resulting system is equivalent to the original one.

**Lemma 3.9.1**

Let \( \overline{x}^T \) be any integer vector and

\[
g = d - k \overline{x}^T.
\]

To see that the system

\[
Dx^S \equiv g \pmod{Q}
\]

has \( M \) distinct roots, we calculate the roots by back substitution starting with \( x_p^S \).

The congruence

\[
d_p, p \overline{x}_p^S \equiv g_p \pmod{Q}
\]

has exactly \( d_p, p \) distinct roots (by Prop. 2.1.6). Let \( \overline{x}_p^S \) be any of these.
By Lemma 3.6.1 the congruence
\[ d_{p-1, p-1} x_{s_{p-1}} \equiv (g_{p-1} - d_{p-1, p} \overline{x}_s) \pmod{Q} \]
is solvable for \( x_{s_{p-1}} \) and by Prop. 2.1.6 it has exactly \( d_{p-1, p-1} \) distinct roots. Therefore there are exactly \( \prod_{k, k} d_{k, i} \) roots satisfying the last 2 congruences.

If we assume there are exactly \( \prod_{i=k+1}^{p} d_{i, i} \) roots satisfying congruences \( k+1, \ldots, p \), then (by Lemma 3.6.1 and Prop. 2.1.6) there are exactly \( \prod_{i=1}^{p} d_{i, i} \) roots satisfying all the congruences.

**Theorem 3.9.2**

If the system given by
\[ Ax \equiv b \pmod{Q} \]
is solvable, then it must have the same number of roots as the system
\[ Dx^S + Kx^T \equiv d \pmod{Q}. \]

Since \( x^T \) is arbitrary, each component \( x_{t_i} \) can assume any integer value between 0 and \( Q \). Therefore there are \( Q^{n-p} \) integer vectors \( x^T \) such that
\[ 0 \leq x_{t_i} < Q \ (i = 1, \ldots, n-p). \]

By Lemma 3.9.1, for each of these vectors \( x^T \), the system
\[ Dx^S + Kx^T \equiv d \pmod{Q} \]
has exactly \( M \) distinct roots.
Therefore the system of congruences
\[ Ax \equiv b \pmod{Q} \]
has exactly \( MQ^{n-p} \) distinct roots.

**Lemma 3.10.1**

On the \( j \)th iteration of the Triangularization Process \( C_j \) is transformed to a \( k \)-triangularized vector \( \hat{C}_j \) where \( j \geq k \). Since \( j \) is not adjoined to the sequence \( S \), \( \hat{C}_{h,j} = 0 \) for \( h = k, \ldots, m+k-1 \). By checking the steps of the Equivalence Algorithm it is easily verified that from this point on only the following operations are performed:

1) Elementary row operations on rows whose indices are greater than or equal to \( k \).

2) Addition of new constraint rows

3) Reduction of elements modulo \( Q \)

First we note that any new constraint row will be a scalar multiple of row \( h \) (\( h \geq k \)) and therefore any new entry in column \( j \) must be zero.

Since all elementary row operations involve only those rows with indices greater than or equal to \( k \), the zero entries in column \( j \) from rows \( k \) on will remain unchanged. A zero entry reduced modulo \( Q \) remains zero.

Therefore at step 8 of the Equivalence Algorithm
\[ R_{h,j} = 0 \text{ for } h = k, \ldots, p. \]

By hypothesis \( i > j \geq k \rightarrow R_{i,j} = 0. \)
3.12 Numerical Examples

Some simple examples will be used to illustrate the algorithm.

Example 1: A knapsack problem.

Minimize \( f = 3x_1 + 7x_2 + 4x_3 \)
subject to \( 8x_1 + 3x_2 + 5x_3 \equiv 6 \pmod{11} \)

Order the variables such that \( c_i \leq c_j \) for \( i < j \)
\( + x = [x_1, x_3, x_2] \)

Equivalence Algorithm

1) Form \( C = [8 \ 5 \ 3 \ 6] \)

2) Triangularization Process

Iteration 1)

Since \( C^1 \) is already triangularized, set \( s_1 = 1 \),
\( S = \{1\}, T = \{2, 3\} \). Adjoin the new row \( \hat{C}_2 \)
where \( r = 11/(8, 11) = 11 \) giving
\( \hat{C} = \begin{bmatrix} 8 & 5 & 3 & 6 \\ 0 & 55 & 33 & 66 \end{bmatrix} \)

Reduce each new entry modulo 11 giving
\( \hat{C} = \begin{bmatrix} 8 & 5 & 3 & 6 \\ 0 & 0 & 0 & 0 \end{bmatrix} \)

Since \( \hat{C}S = \begin{bmatrix} 8 \\ 0 \end{bmatrix} \), \( \hat{C}^T = \begin{bmatrix} 5 & 3 \\ 0 & 0 \end{bmatrix} \) and \( \hat{C}^4 = \begin{bmatrix} 6 \\ 0 \end{bmatrix} \), the Triangularization Process terminates with \( \hat{e} = 0 \) and the system is solvable.

4) Determine \( u_1 = 7 \)

Then \( u_1 = 7 \)
5) & 6) Multiply row 1 by $u_1$, and reduce the elements modulo 11 giving

\[
\begin{bmatrix}
1 & 2 & 10 & 9 \\
\end{bmatrix}
\]


8) $R = [1 \ 2 \ 10]$

The original problem may be written equivalently:

Min $f = 3x_1 + 4x_3 + 7x_2$

subject to $x_1 \equiv 9 - 2x_3 - 10x_2 \pmod{11}$

Note that the system has $11^2$ roots and for any integer values of $x_3$ and $x_2$ the system is solvable. In particular letting $x_3 = x_2 = 0$ gives the feasible solution $(9, 0, 0)$ and an upper bound of 27 on the objective function.

Example 3: A problem with two constraints

Minimize $f = x_1 + x_2 + 7x_3$

subject to $4x_1 + 5x_2 + 11x_3 \equiv 7 \pmod{12}$

$2x_1 + 3x_2 + 2x_3 \equiv 5 \pmod{24}$

Multiply constraint (1) by 2 to get the same modulus for both constraints.

Equivalence Algorithm

1) Form $C = \begin{bmatrix}
8 & 10 & 22 & 14 \\
2 & 3 & 2 & 5 \\
\end{bmatrix}$

\[ Q = 24 \]

2) Triangularization Process

Iteration 1)

Triangularize $C^1$ giving

\[ \hat{C} = \begin{bmatrix}
2 & 3 & 2 & 5 \\
0 & -2 & 14 & -6 \\
\end{bmatrix} \]
Reduce elements of \( \hat{C} \) (modulo 24) giving
\[
\hat{C} = \begin{bmatrix} 2 & 3 & 2 & 5 \\ 0 & 22 & 14 & 18 \end{bmatrix}
\]

Set \( s_1 = 1 \), \( S = \{1\} \) and \( T = \{2,3\} \)

Add a new row \( \hat{C}_3 \) (where \( r = 24/(2,24) = 12 \))
giving
\[
\hat{C} = \begin{bmatrix} 2 & 3 & 2 & 5 \\ 0 & 22 & 14 & 18 \\ 0 & 12 & 0 & 12 \end{bmatrix}
\]

Iteration 2)

Triangularize \( C^2 \) and reduce elements (mod 24)
giving
\[
\hat{C} = \begin{bmatrix} 2 & 3 & 2 & 5 \\ 0 & 2 & 10 & 6 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]

Set \( S = \{1,2\} \) and \( T = \{3\} \)

Add a new row \( \hat{C}_4 \) (where \( r = 24/(2,24) = 12 \))
giving
\[
\hat{C} = \begin{bmatrix} 2 & 3 & 2 & 5 \\ 0 & 2 & 10 & 6 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]

Iteration 3)

\( C^3 \) is already triangularized.

Set \( S = \{1,2,3\} \) and \( T = \phi \)

Add a new row \( \hat{C}_5 \) (where \( r = 24/(12,24) = 2 \))
giving
\[
\hat{C} = \begin{bmatrix} 2 & 3 & 2 & 5 \\ 0 & 2 & 10 & 6 \\ 0 & 0 & 12 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\]
The Triangularization Process is now complete with
\[
\hat{C}^S = \begin{bmatrix}
2 & 3 & 2 \\
0 & 2 & 10 \\
0 & 0 & 12 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\text{ and } \hat{C}^4 = \begin{bmatrix}
5 \\
6 \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[ T = \phi \text{ and } p = 3. \]

3) Since \( \hat{e} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \), the system is solvable.

4) It is immediately determined that \( u = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \)

5) May be bypassed

6) Reducing the off-diagonal elements of columns 2 and 3 gives
\[
\begin{bmatrix}
2 & 1 & 4 & 23 \\
0 & 2 & 10 & 6 \\
0 & 0 & 12 & 0
\end{bmatrix}
\]

7) \( D = \begin{bmatrix} 2 & 1 & 4 \\ 0 & 2 & 10 \\ 0 & 0 & 12 \end{bmatrix}, \quad d = \begin{bmatrix} 23 \\ 6 \\ 0 \end{bmatrix} \)

8) \( R = D \)

The original problem may now be written equivalently as:

\[
\text{Min } f = x_1 + x_2 + 7x_3 \\
\text{subject to } 2x_1 + x_2 + 4x_3 = 23 \\
2x_2 + 10x_3 \equiv 6 \pmod{24} \\
12x_3 \equiv 0
\]

Note that the system has 48 roots. From inspection it is easily seen that \((10, 3, 0)\) is a feasible solution giving an upper bound of 13 to the objective function.
Example 2:

Minimize \( f = 3x_1 + 4x_2 + 5x_3 + 7x_4 \)

subject to \( 5x_1 + 9x_2 + 3x_3 + 4x_4 \equiv 1 \pmod{10} \)

**Equivalence Algorithm**

1) Form \( C = \begin{bmatrix} 5 & 9 & 3 & 4 & 1 \end{bmatrix} \)

2) Apply Triangularization Process

**Iteration 1)**

\( C^1 \) is already triangularized

Set \( S = \{1\} \) \( T = \{2,3,4\} \)

Adjoin \( \hat{C}_2 \) giving

\( \hat{C} = \begin{bmatrix} 5 & 9 & 3 & 4 & 1 \\ 0 & 8 & 6 & 8 & 2 \end{bmatrix} \)

**Iteration 2)**

\( C^2 \) is already triangularized

Set \( S = \{1,2\} \) \( T = \{3,4\} \)

Adjoin \( \hat{C}_3 \) giving

\( \hat{C} = \begin{bmatrix} 5 & 9 & 3 & 4 & 1 \\ 0 & 8 & 6 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \)

The Triangularization Process is now complete with

\( V = \begin{bmatrix} 5 & 9 \\ 0 & 8 \end{bmatrix} \)

\( W = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix} \)

\( e = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \)

\( \hat{e} = [0] \)

3) Since \( \hat{e} = 0 \), the system is solvable

4) It is immediately seen that \( u_1 = 1 \).

Determine \( u_2 \) = any root of

\( 8Z \equiv 2 \pmod{10} \)

Then \( u_2 = 4 \) and \( u = \begin{bmatrix} 1 \\ 4 \end{bmatrix} \)
5) Multiply row 2 of
\[
\begin{bmatrix}
5 & 9 & 3 & 4 & 1 \\
0 & 8 & 6 & 8 & 2
\end{bmatrix}
\]
by 4 giving
\[
\begin{bmatrix}
5 & 9 & 3 & 4 & 1 \\
0 & 32 & 24 & 32 & 8
\end{bmatrix}
\]

6) Reduce all elements giving
\[
\begin{bmatrix}
5 & 1 & 7 & 6 & 9 \\
0 & 2 & 4 & 2 & 8
\end{bmatrix}
\]

7) \( D = \begin{bmatrix} 5 & 1 \\ 0 & 2 \end{bmatrix} \quad K = \begin{bmatrix} 7 & 6 \\ 4 & 2 \end{bmatrix} \quad d = \begin{bmatrix} 9 \\ 8 \end{bmatrix} \)

8) \( R = [DK] \)

The original problem may now be written equivalently as

\[
\text{Minimize } 3x_1 + 4x_2 + 5x_3 + 7x_4
\]
subject to \( 5x_1 + x_2 \equiv 9 - 7x_3 - 6x_4 \pmod{10} \)

\[
2x_2 \equiv 8 - 4x_3 - 2x_4
\]

The system has \( 10^3 \) roots. From inspection it is easily seen that \((1,4,0,0)\) is a feasible solution giving an upper bound of 19 to the objective function.
4.1 Double Description

It is well known (see Theorem 2.3.6) that the integer solutions to a consistent system of linear equations

\[ Ax = b \]

can be represented by

\[ x = d + Ey, \quad y \text{ integer.} \]

An analogous theorem for a consistent system of linear congruences may now be stated.

**Double Description Theorem**

If a system of linear congruences given by:

\[ Ax \equiv b \pmod{Q} \]

is solvable with equivalent representation:

\[ Dx^S + Kx^T \equiv d \pmod{Q} \]

as derived by the equivalence algorithm, then all integer solutions are given by:

\[ x^S = W^0 + Wx^T + Ey, \]

\[ x^T \text{ integer} \]

\[ y \text{ integer} \]

where:

\[ W^0 = \text{a nonnegative, integer p-vector} \]

\[ W = \text{a} \ (p)x(n-p) \text{ nonnegative, integer matrix} \]
and

\[ E = a \text{ (p)}x(p) \text{ upper triangular, nonnegative integer matrix with positive diagonal elements.} \]

A constructive proof of the theorem utilizes the following lemmas:

**Lemma 4.1.1**

Let \( U = \text{any (m)}x(p) \text{ integer matrix} \)

\[ V = \text{any (m)}x(r) \text{ integer matrix} \]

\[ g = \text{an m-vector of integers} \]

\[ Z = \text{an m-vector of positive integers}. \]

If for each integer choice of the vector \( y \), the system of congruences given by:

\[ Uw + Vy \equiv g \pmod{Z} \]

is satisfied by at least one integer vector \( w \) then \( G(V,Z) \) is a subgroup of \( G(U,Z) \) where \( G(V,Z) \) is the group formed by the columns of \( V \) under the operation of addition modulo \( Z \) and \( G(U,Z) \) is defined similarly (see Prop. 2.2.3).

**Lemma 4.1.2**

Let the system of linear congruences

\[ S \]

\[ Dx^T + Kx \equiv d \pmod{Q} \]

be as derived by the equivalence algorithm, then \( G(K,Q) \) is a subgroup of \( G(D,Q) \) and the systems

\[ Dx^S \equiv d \]

\[ Dx^S \equiv -K^j \quad (j=1,\ldots,(n-p)) \]

each have at least one root.
Let $W^0$ be any root of $Dx^S \equiv d \pmod{Q}$

$W^j$ be any root of $Dx^S \equiv -K^j \pmod{Q}$ \quad (j=1,\ldots,(n-p))

$W$ be the $(p)\times(n-p)$ matrix whose columns are

$W^j \quad (j=1,\ldots,(n-p)).$

**Lemma 4.1.3**

Let the system of congruences

$$Dx^S + Kx^T \equiv d \pmod{Q}$$

be as derived by the equivalence algorithm, then for any integer vector $\bar{x}^T$, the vector

$$\bar{x}^S = (W^0 + W\bar{x}^T)$$

satisfies the system

$$D\bar{x}^S + K\bar{x}^T \equiv d \pmod{Q}$$

where $W^0$ and $W$ are as defined above.

**Lemma 4.1.4**

Let the matrix $D$ be as derived by the equivalence algorithm, then the set of solutions to the homogeneous system of congruences:

$$Dx^S \equiv 0 \pmod{Q}$$

is given by

$$x^S = Ey, \; y \text{ integer}$$

where $E \in T(p)$ and $0 \leq e_{ij} < e_{ii} = Q/d_{ii}$ for $1 \leq i < j \leq p$.

The proof of Lemma 4.1.4 illustrates the construction of such a matrix $E$ and a numerical example is included in 4.4. The columns of the matrix $E$ represent a set of generators of the group of solutions to the homogenous system

$$Dx^S \equiv 0 \pmod{Q}.$$
Lemma 4.1.5

Let the matrix $D$ be as derived by the equivalence algorithm and let $g$ be any integer $p$-vector such that the system:

$$Dx^S \equiv g \pmod{Q} \quad 4.1.2$$

is solvable with solution

$$x^S = g^0.$$ 

Then the set of all solutions to $4.1.2$ is given by:

$$x^S = g^0 + Ey, \ y \text{ integer}$$

where $E$ is as specified in Lemma 4.1.4.

We may now state the Double Description Theorem more precisely.

Let

$$Dx^S + Kx^T \equiv d \pmod{Q}$$

be the system of congruences derived by the equivalence algorithm, then all integer solutions are given by:

$$x^S = W^0 + Wx^T + Ey, \ x^T \text{ and } y \text{ integer}$$

where

$W^0$ is a root of $Dx^S \equiv d \pmod{Q}$

$W^j$ is a root of $Dx^S \equiv -K^j \pmod{Q}$ for $j = 1, \ldots, (n-p)$

and

$E \in T(p)$ and its columns generate the group of solutions to the homogeneous system $Dx^S \equiv 0 \pmod{Q}$.

Although the Double Description Theorem is not utilized by the algorithm described in this paper, it illustrates an important property of the representation which could be exploited by subsequent algorithms.
4.2 Group Properties

As stated in Chapter 1, there are a number of group algorithms for solving the ILPC in Smith Canonical form. In this form the constraints are written as

\[ A'x \equiv b' \pmod{\delta} \]  \hspace{1cm} 4.2.1

where \( \delta = \begin{pmatrix} \delta_k \\ \delta_m \end{pmatrix} \) (see 1.2.6). The amount of computation required by these algorithms is generally proportional to the order of \( G(A', \delta) \), the group generated by the columns of \( A' \) under the operation of addition modulo \( \delta \).

If the ILPC is solvable, then we have seen in Chapter 3 that the constraints may be equivalently written:

\[ Dx^S + Kx^T \equiv d \pmod{Q} \] \hspace{1cm} 4.2.2

where \( Q = \delta_m \).

If we denote:

\( A'^S \) by \( U \)

and

\( A'^T \) by \( V \),

then 4.2.1 may be written:

\[ Ux^S + Vx^T \equiv b' \pmod{\delta} \] \hspace{1cm} 4.2.3

and for any integer vector \( x^T \) the system of congruences is solvable.

It follows immediately from Lemma 4.1.1 that \( G(V, \delta) \) is a subgroup of \( G(U, \delta) \). Let \( |G(\_)| \) denote the order of the group \( G(\_ \_ \) . The following lemmas indicate the order of \( G(K, Q) \) relative to \( G(A', \delta) \).
Lemma 4.2.1

\[ |G(K, Q)| = |G(V, \delta)| \]

Lemma 4.2.2

\[ |G(K, Q)| \text{ is a factor of } |G(A', \delta)|. \]

When the Primal-Dual Algorithm is presented in Chapter 6, it will be seen that computation time is dependent upon the number of distinct integer vectors \( x^T \) that are generated using Glover's Method. If a group type algorithm were to replace Glover's Method in the Primal-Dual Algorithm, then computation time would be proportional to \( |G(K, Q)| \) which by the preceding lemma is a factor of \( |G(A', \delta)| \). This suggests that variations of the Primal-Dual Algorithm incorporating group methods could be developed with computation times competitive with, or even significantly better than the same group methods as applied to Smith Canonical form of the ILPC.

4.3 Proofs of Theorems

Lemma 4.1.1

Let \( v \in G(V, Z) \), then \( v \equiv \bar{v}y \pmod{Z} \) for some integer \( \bar{v} \).

By hypothesis the system

\[ Uw \equiv g - \bar{v}y \equiv (g-v) \pmod{Z} \]

is solvable \( \rightarrow \) (by Prop. 2.2.5)

\[ (g - v) \in G(U, Z) \]

By hypothesis the system

\[ Uw \equiv g \pmod{Z} \]

is also solvable \( \rightarrow \) (by Prop. 2.2.5)
\[ g \in G(U,Z) \rightarrow (\text{by def. 2.2.1}) \]
\[ -g \in G(U,Z). \]
By group closure we have
\[ (-g) + (g-v) \equiv -v \in G(U,Z) \rightarrow \]
\[ v \in G(U,Z) \text{ and} \]
\[ G(V,Z) \text{ is a subgroup of } G(U,Z). \]

**Lemma 4.1.2**

By Lemma 4.1.1, \( G(K,Q) \) is a subgroup of \( G(D,Q) \).
Since the system is solvable we know that \( d \in G(D,Q) \).
Since \( -k_j \in G(K,Q) \rightarrow -k_j \in G(D,Q) \) for \( j = 1, \ldots, (n-p) \).
By Prop. 2.2.5 the systems
\[
\begin{align*}
&Dx^S \equiv d \\
&Dx^S \equiv -k_j \quad (j = 1, \ldots, (n-p))
\end{align*}
\]
are solvable.

**Lemma 4.1.3**

By definition
\[ DW^0 \equiv d \pmod{Q} \]
and \( DW^j \equiv -k_j \pmod{Q} \) \( (j = 1, \ldots, (n-p)) \).
By Prop. 2.1.3
\[ DW^j \bar{x}_t \equiv -k_j \bar{x}_t \pmod{Q} \quad (j = 1, \ldots, (n-p)) \]
(by Prop. 2.1.2)
\[
\sum_{j=1}^{n-p} D(W^j \bar{x}_t) \equiv -\sum_{j=1}^{n-p} k_j \bar{x}_t \pmod{Q} \quad \text{or} \]
\[ DW \bar{x}^T \equiv -k \bar{x}^T \pmod{Q}. \]
If \( \bar{x}^S = (W^0 + W \bar{x}^T) \) then
\[ D\bar{x}^S + K \bar{x}^T = DW^0 + DW \bar{x}^T + K \bar{x}^T \equiv d - K \bar{x}^T + K \bar{x}^T \pmod{Q} \]
and therefore \( \bar{x}^S \) satisfies the system
\[ D\bar{x}^S + K \bar{x}^T \equiv d \pmod{Q}. \]
Lemma 4.1.4

By the upper triangularity of the matrix $D$ we see that for any $i$ ($1 \leq i \leq p$) the solution set of the system of congruences given by

$$D_j x^S \equiv 0 \pmod{Q} \quad (j=i, \ldots, p) \quad 4.3.1$$

depends only on the variables $x_{S_i}, x_{S_{i+1}}, \ldots, x_{S_p}$.

Let $Z = \begin{pmatrix} x_{S_i} \\ x_{S_{i+1}} \\ \vdots \\ x_{S_p} \end{pmatrix}$

It will be shown by backwards induction that for any $i$ ($1 \leq i \leq p$) the solution set $Z$ corresponding to 4.3.1 is given by

$$Z = F y$$

where $y = \begin{pmatrix} y_i \\ \vdots \\ y_p \end{pmatrix}$

is an arbitrary integer vector and $F$ is an upper triangular matrix with the property that

$$0 \leq f_{jk} < f_{jj} = Q/d_{jj} \quad \text{for } i \leq j < k \leq p.$$ 

First we define

$$f_{jj} = Q/d_{jj} \quad \text{for } j = 1, \ldots, p.$$

By the Triangularization Theorem,

$$d_{jj} > 0 \quad \text{and} \quad d_{jj} \mid Q \
\Rightarrow \quad f_{jj} > 0 \quad \text{and} \quad f_{jj} \text{ integer for } j = 1, \ldots, p.$$
Inductive proof for the case $i = p$.

The single congruence

$$D_p x^S \equiv 0 \pmod{Q}$$

is given by

$$d_{pp} x^S_p \equiv 0 \pmod{Q}$$

By Prop. 2.1.4 and Prop. 2.1.5, the solution to 4.3.2 is

$$x^S_p \equiv 0 \pmod{f_{pp}}$$

or by Def. 2.1.1

$$x^S_p = f_{pp} y_p$$ for $y_p$ integer.

Thus the lemma is proved for the case $i = p$.

Assume the inductive hypothesis is true for $i+1$, that is

$$Z = \begin{pmatrix} x^S_{i+1} \\ \vdots \\ x^S_p \end{pmatrix}$$

and the solution set corresponding to congruences $i+1, \ldots, p$

is given by

$$Z = F y$$

where $y = ( y_{i+1} \ldots y_p )$ is an arbitrary integer vector

$$\begin{pmatrix} y_{i+1} \\ \vdots \\ y_p \end{pmatrix}$$

and $F$ has the required properties.

Then for the $i$th case the system of congruences is given by:

$$\begin{cases} d_{ii} x^S_i \equiv \prod_{j=i+1}^p \gamma_j x^S_j \pmod{Q} \\ Z = F y, y \text{ integer} \end{cases}$$

Define a row vector $v = (v_{i+1}, \ldots, v_p)$ as follows:

$$v_j = Q - d_{ij} \quad \text{for } j = i+1, \ldots, p.$$
By definition
\[ v_j \equiv -d_{ij} \pmod{Q} \]
and \( v_j \geq 0 \).

By Prop. 2.1.0 the system given by 4.3.3 is equivalent to:
\[
\begin{align*}
\text{d}_{ii} x_{S_i} & \equiv v Z \pmod{Q} \\
Z & = F y, \text{ y integer}
\end{align*}
\] 4.3.4

Define a row vector \( w = v F \). Then we may rewrite 4.3.4 equivalently as
\[
\begin{align*}
\text{d}_{ii} x_{S_i} & \equiv w y \pmod{Q} \\
Z & = F y, \text{ y integer}
\end{align*}
\] 4.3.5

By Prop. 2.1.4 and Prop. 2.1.5, the system given by 4.3.5 is equivalent to
\[
\begin{align*}
x_{S_i} & \equiv \text{P} \sum_{j=i+1}^{p} (w_j / d_{ii}) y_j \pmod{f_{ii}} \\
Z & = F y, \text{ y integer}
\end{align*}
\] 4.3.6

To see that \( d_{ii} \mid w_j \) for all \( j = i+1, \ldots, p \), we note that by Lemma 3.6.1 \( x_{S_i} \) is integer for any integer choice of the vector \( y \). Choosing \( y \) to be the vector with 1 in the \( j \)th position and zeros elsewhere gives
\[
\begin{align*}
x_{S_i} & \equiv w_j / d_{ii} \pmod{f_{ii}} \\
w_j / d_{ii} \text{ must be integer for all } j.
\end{align*}
\]

Define
\[
f_{ij} = w_j / d_{ii} \text{ reduced modulo } f_{ii}
\]
for \( j = i+1, \ldots, p \).

By Prop. 2.1.0 the system given by 4.3.6 is equivalent to
\[
\begin{align*}
x_{S_i} & \equiv \text{P} f_{ij} y_j \pmod{f_{ii}} \\
Z & = F y, \text{ y integer}
\end{align*}
\] 4.3.7
or equivalently

\[
x_{s_i} = f_{ii}y_i + \sum_{j=i+1}^{p} f_{ij}y_j, \quad y_i \text{ integer}
\]

\[
Z = Fy, \quad y \text{ integer}
\]

If we define

\[
F' = \begin{bmatrix}
    f_{ii} & f_{i,i+1} & \cdots & f_{ip} \\
    0 & \ddots & \ddots & \ddots \\
    \vdots & \ddots & \ddots & 0 \\
    0 & \cdots & \ddots & 0
\end{bmatrix}
\]

\[
Z' = \begin{bmatrix}
x_{s_i} \\
Z
\end{bmatrix}
\]

and

\[
y' = \begin{bmatrix}
y_i \\
y
\end{bmatrix}
\]

then the solution set of 4.3.3 is given by the set of all \( Z' \) such that

\[
Z' = F'y', \quad y' \text{ integer}
\]

and \( F' \) has all the required properties.

**Lemma 4.1.5**

It will be shown that \( g' \) is a solution to 4.1.2 which implies and is implied by the fact that

\[
g' = g^0 + E\bar{y}, \quad \bar{y} \text{ integer}.
\]

Part a]

Let \( g' \) be any solution to

\[
Dx^S \equiv g \pmod{Q}
\]
Since $g^0$ is also a solution
\[ D(g' - g^0) \equiv \theta \pmod{Q} \rightarrow \text{(by Lemma 4.1.4)} \]
\[ (g' - g^0) = E\tilde{y} \text{ for some integer vector } \tilde{y} \rightarrow \]
\[ g' = g^0 + E\tilde{y}. \]

Part b)
Let $g' = g^0 + E\tilde{y}$ for some integer vector $\tilde{y} \rightarrow$
\[ Dg' = Dg^0 + D(E\tilde{y}) \equiv g + \theta \equiv g \pmod{Q} \rightarrow \]
$g'$ is a solution to $Dx^S \equiv g \pmod{Q}$.

**Double Description Theorem**

By Lemma 4.1.3 for any integer vector $\overline{x}^T$, the vector
\[ \overline{x}^S = (W^0 + W\overline{x}^T) \] solves the system
\[ Dx^S \equiv d - K\overline{x}^T \pmod{Q} \rightarrow \text{(by Lemma 4.1.5)} \]
the set of all solutions to the system
\[ Dx^S \equiv d - K\overline{x}^T \pmod{Q} \]
is given by
\[ x^S = (W^0 + W\overline{x}^T) + Ey, \text{ } y \text{ integer} \]
where $E$ is as specified in Lemma 4.1.4.

Since $\overline{x}^T$ is arbitrary the theorem follows.

**Lemma 4.2.1**

Let $k \in G(K,Q)$, then $k \equiv K\overline{x}^T \pmod{Q}$ for some integer vector $\overline{x}^T$ and $0 \leq k_i < Q \quad (i = 1, \ldots, p)$
Define $v = V\overline{x}^T$ reduced modulo $\delta$ and $g(k) = v$.
Note that $v \in G(V,\delta)$.

It will be shown that $g$ is a 1-1, onto mapping from $G(K,Q)$ to $G(V,\delta)$.

**Property 1**] For each $k$, $g(k)$ is uniquely defined.
Proof:

Suppose \( k \equiv K\bar{x}^T \equiv K\hat{x}^T \pmod{Q} \).

Let \((\bar{x}^S, \bar{x}^T)\) be any solution to 4.2.2
\[
K\bar{x}^T \equiv d - D\bar{x}^S \pmod{Q} \\
K\hat{x}^T \equiv d - D\hat{x}^S \pmod{Q}
\]

\((\bar{x}^S, \hat{x}^T)\) also solves 4.2.2.

Since 4.2.2 and 4.2.3 are equivalent, then both \((\bar{x}^S, \bar{x}^T)\)
and \((\bar{x}^S, \hat{x}^T)\) solve 4.2.3
\[
V\bar{x}^T \equiv b' - U\bar{x}^S \pmod{\delta} \bigg\} + \bigg\}
V\bar{x}^T \equiv b' - U\hat{x}^S \pmod{\delta} \\
V\hat{x}^T \equiv V\hat{x}^T \pmod{\delta}.
\]

If \( \hat{v} = V\hat{x}^T \) reduced modulo \( \delta \) and \( v = V\bar{x}^T \) reduced modulo \( \delta \),
then since \( \hat{v} \equiv v \pmod{\delta} \) we have that \( \hat{v} = v \) and \( g \) is a
mapping from \( G(K,Q) \) into \( G(V,\delta) \).

Property 2] \( g \) is 1-1 and onto.

Proof:

Let \( k \in G(K,Q) \) and \( h \in G(K,Q) \) such that \( k \neq h \).

Then \( k \equiv K\bar{x}^T \pmod{Q} \) and \( h \equiv K\hat{x}^T \pmod{Q} \) where
\[
K\bar{x}^T \neq K\hat{x}^T \pmod{Q}.
\]

Let \((\bar{x}^S, \bar{x}^T)\) be any solution to 4.2.2
\[
K\bar{x}^T \equiv d - D\bar{x}^S \pmod{Q} \\
\text{and} \bigg\}
K\hat{x}^T \neq d - D\hat{x}^S \pmod{Q}
\]

\((\bar{x}^S, \hat{x}^T)\) does not solve 4.2.2.

Since 4.2.2 and 4.2.3 are equivalent, \((\bar{x}^S, \bar{x}^T)\) solves 4.2.3
but \((\bar{x}^S, \hat{x}^T)\) does not solve 4.2.3 +
\[
\begin{align*}
V\bar{x}^T &\equiv b' - UX^S \pmod{\delta} \\
V\hat{x}^T &\not\equiv b' - UX^S \pmod{\delta} \\
V\bar{x}^T &\not\equiv V\hat{x}^T \pmod{\delta}
\end{align*}
\]

But \(g(k) = \bar{v} = V\bar{x}^T\) reduced modulo \(\delta\)
and \(g(h) = \hat{v} = V\hat{x}^T\) reduced modulo \(\delta\) \}
\(g(k) \neq g(h)\) and
\(g\) is a 1-1 map from \(G(K,Q)\) into \(G(V,\delta)\) +
\[
|G(K,Q)| \leq |G(V,\delta)|
\]
Similarly a 1-1 map from \(G(V,\delta)\) into \(G(K,Q)\) may be defined +
\[
|G(V,\delta)| \leq |G(K,Q)|
\]
and therefore
\[
|G(K,Q)| = |G(V,\delta)|
\]
and Property 2 is proved.

**Lemma 4.2.2**

Since \(G(V,\delta)\) is a subgroup of \(G(A',\delta)\) + (by Prop.2.2.6)
\[
|G(V,\delta)| \text{ is a factor of } |G(A',\delta)|
\]
(by Lemma 4.2.1)
\[
|G(K,Q)| \text{ is a factor of } |G(A',\delta)|
\]

**4.4 Numerical Examples**

Consider example 3 of Chapter 3 where
\[
D = \begin{bmatrix}
2 & 1 & 4 \\
0 & 2 & 10 \\
0 & 0 & 12
\end{bmatrix}
\]
\(Q = 24\)
and \[
\begin{pmatrix}
10 \\
3 \\
0 \\
\end{pmatrix}
\] is a particular solution to
\[
\begin{pmatrix}
Dx \\
\end{pmatrix} \equiv
\begin{pmatrix}
23 \\
6 \\
0 \\
\end{pmatrix} \pmod{24}
\]

Construct E from
\[
Dx \equiv 0 \pmod{24}
\]

1) Initialize \( k = 3 \)

\[
12 \ x_3 \equiv 0 \pmod{24} \rightarrow
\]

\[
x_3 = 2y_3
\]

and \( E_3 = (002) \)

2) \( k = 2 \)

\[
2x_2 + 10x_3 \equiv 0 \pmod{24}
\]

and \( x_3 = 2y_3 \)

\[
2x_2 \equiv 14x_3 \pmod{24}
\]

\[
x_3 = 2y_3
\]

\[
2x_2 \equiv 4y_3 \pmod{24}
\]

\[
x_3 = 2y_3
\]

\[
x_2 \equiv 2y_3 \pmod{12}
\]

\[
x_3 = 2y_3
\]

\[
x_2 = 12y_2 + 2y_3
\]

\[
x_3 = 0y_2 + 2y_3
\]

and \( E_2 = (0 12 2) \)

3) \( k = 1 \)
\[ 2x_1 + x_2 + 4x_3 \equiv 0 \pmod{24} \]
\[ \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 & 2 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} y_2 \\ y_3 \end{pmatrix} \]

\[ 2x_1 \equiv (23 \ 20) \begin{pmatrix} 12 & 2 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} y_2 \\ y_3 \end{pmatrix} \pmod{24} \]
\[ x_1 \equiv 6y_2 + 7y_3 \pmod{12} \]
\[ \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 & 2 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} y_2 \\ y_3 \end{pmatrix} \]

\[ x_1 = 12y_1 + 6y_2 + 7y_3 \]
\[ x_2 = 0 + 12y_2 + 2y_3 \]
\[ x_3 = 0 + 0 + 2y_3 \]

and
\[ E = \begin{bmatrix} 12 & 6 & 7 \\ 0 & 12 & 2 \\ 0 & 0 & 2 \end{bmatrix} \]

the set of all solutions is:
\[ x = \begin{pmatrix} 10 \\ 3 \\ 0 \end{pmatrix} + Ey, \ y \text{ integer} \]
CHAPTER 5

AN AUXILIARY PROBLEM

5.1 Statement of the Problem

In order to simplify the presentation of the Primal-Dual Algorithm in Chapter 6, an auxiliary problem will be defined and its method of solution outlined in this chapter. The problem may be stated as follows:

\[
\begin{align*}
\text{Min } g(x) &= \sum_{i=1}^{p} c_i x_i \\
Dx &\equiv v \pmod{Q} \\
g(x) &\leq G \\
x &\geq 0, \text{ integer}
\end{align*}
\]

where

\[
\begin{align*}
c_i &\text{ integer} \\
0 &\leq c_{i-1} \leq c_i \quad \text{for } i = 1, \ldots, p \\
d_{ii} &\mid Q \\
G &\text{ a positive integer or } G = \infty. \\
D &\in T(p)
\end{align*}
\]

\[v = \text{ an integer p-vector with nonnegative components not all of which are zero.}\]

Moreover, if \( M = \prod_{i=1}^{p} d_{ii} \) then the system of congruences has exactly \( M \) roots.

By the nonnegativity of the objective function coefficients, only the \( M \) roots of the system of congruences are candidates to be an optimal solution. Any root is...
feasible for 5.1.1 if it satisfies \( g(x) \leq G \). If none of the roots satisfies the condition \( g(x) \leq G \), then the problem has no feasible solution. Otherwise let \( x^* \) represent an optimal solution to 5.1.1 with \( g^* = \) the optimal objective function value.

Because \( D \) is upper triangular with positive diagonal elements that divide \( Q \), all the roots may be easily calculated by back substitution (see 3.7).

5.2 Elimination of Roots

Let the \( p \)-vectors \( b_1, \ldots, b_M \) denote the roots of the system of congruences ordered according to the following convention:

If \( b_i = \begin{pmatrix} x_i^1 \\ \vdots \\ x_i^p \end{pmatrix} \) and \( b_j = \begin{pmatrix} x_j^1 \\ \vdots \\ x_j^p \end{pmatrix} \)

where

\[
x_i^k < x_j^k \\
x_i^k = x_j^{k+1} \\
x_i^k = x_j^{k+2} \\
\vdots \\
x_i^p = x_j^p
\]

for some \( k \) between 1 and \( p \), then \( i < j \).

If an optimal solution to 5.1.1 exists but is not unique, we arbitrarily define the optimal representative \( x^* \) as follows:

If \( g^* = g(b_i) = g(b_j) \) where \( i < j \), then \( x^* = b_i \).
The following lemma asserts that some of the roots may always be deleted from consideration.

**Lemma 5.2.1**

Denote \( \overline{M} = M / d_{11} \)

Define a sequence \( V = (v_1, \ldots, v_{\overline{M}}) \) by

\[
\begin{align*}
  v_1 &= 1 \\
  \vdots \\
  v_j &= v_{j-1} + d_{11}. \\
\end{align*}
\]

Then if \( j \not\in V \), \( b_j \neq x^* \).

From this lemma we see that to determine \( x^* \) we need to examine at most \( \overline{M} \) of the \( M \) roots to the system of congruences.

The following obvious properties will also be used to eliminate roots from consideration.

**Property 1**] If a root \( b_i \) is feasible with \( \overline{g} = g(b_i) \), then for any \( j > i \), \( b_j \) can be optimal only if \( g(b_j) \leq \overline{g} - 1 \).

**Property 2**] If \( b_i = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_j \\ \hat{x}_k \\ \hat{x}_p \end{pmatrix} \) is infeasible with \( \sum_{k=j}^p c_k \hat{x}_k \geq G \),

then all roots are infeasible having \( x_j \geq \hat{x}_j \) and

\[
x_k = \hat{x}_k \quad (k = j+1, \ldots, p).
\]

**Property 3**] If \( D^j \) is all zeros except for \( d_{jj} \) \((1 \leq j \leq p)\), then any change in \( x_j \) has no effect on any of the other variables.

5.3 The initial solution

Let the roots of the system of congruences given
by 5.1.1 be ordered by the convention defined in 5.2 and define the first root \( b_1 = \begin{pmatrix} x_1^1 \\ \vdots \\ x_1^p \end{pmatrix} \) as the initial solution.

Define \( e_i = Q/d_{ii} \) (i = 1, ..., p).

Then by the properties of D, \( 1 \leq e_i \leq Q \), \( e_i \) integer for \( i = 1, ..., p \).

The following lemmas assert important properties of the initial solution.

**Lemma 5.3.1**

If \( b_1 = \begin{pmatrix} x_1^1 \\ \vdots \\ x_1^p \end{pmatrix} \), then \( 0 \leq x_i^1 < e_i \) (i = 1, ..., p).

**Lemma 5.3.2**

Let \( b_k = \begin{pmatrix} x_k^1 \\ \vdots \\ x_k^p \end{pmatrix} \) be any other root of the system of congruences. Then for some i (i = 1, ..., p)

\[
x_k^i = x_i^1 + y_i e_i \quad y_i > 0, \text{ integer}
\]

and \( x_{k+1}^i = x_{i+1}^1 \).

**Lemma 5.3.3**

If \( \sum_{j=1}^{i-1} c_{j} e_j \leq c_i e_i \) for all i = 2, ..., p, then for any choice of the right hand side v for which the system
of congruences 5.1.1 is solvable, the initial solution is the only candidate to be the optimal solution \( x^* \).

Note that the conditions of this lemma involve only the objective function coefficients \( c_j \), the diagonal elements \( d_{jj} \) and the modulus \( Q \). If these conditions are satisfied, then for any choice of the right hand side \( v \) for which the system of congruences is solvable either the initial solution \( b_1 \) is optimal or \( g(b_1) > G \) and no feasible solution exists.

**Lemma 5.3.4**

If \( \sum_{j=1}^{i-1} c_j x_j^1 < c_i e_i \) for \( i = 2, \ldots, p \), then the initial solution is the only candidate to be the optimal solution \( x^* \).

The conditions of this lemma involve components of the initial solution and therefore involve a particular choice of the right hand side. If these conditions are satisfied then again either \( b_1 \) is optimal or \( g(b_1) > G \) and no feasible solution exists.

### 5.4 Enumerative Algorithm

The algorithm begins with a dummy \( p \)-vector \( z \) set equal to 0 and an upper bound \( \bar{g} \) set equal to \( G \). The roots \( b_j \) (\( j \in V \)) are calculated by back substitution combined with the formula of Lemma 5.3.2. If a root \( b_i \) is found satisfying the constraint \( g(b_i) \leq \bar{g} \), then \( z \) is set equal to \( b_i \) and \( \bar{g} \) is set equal to \( g(b_i) - 1 \). The procedure is iteratively repeated until all roots have either been
calculated or eliminated using the above properties.
Upon termination, if \( z = 0 \), no feasible solutions exist.
Else \( x^* = z \) and \( g^* = g(x^*) \).

Step 1) Initialization:
Set \( z = 0 \)
Set \( j = p \)
Set \( \bar{g} = G \)
Compute \( e_i = Q/d_{ii} \quad (i = 1, \ldots, p) \).

Step 2]
If \( j = p \), set \( r = 0 \) and go to 3.
Else compute \( r = \prod_{k=j}^{p} d_{jk} \hat{x}_k \) and continue.

Step 3]
Compute \( \hat{x}_j = (v_j - r)/d_{jj} \) reduced modulo \( e_j \).

Step 4]
Compute \( c = \prod_{k=j}^{p} c_k \hat{x}_k \).

Step 5]
If \( c > \bar{g} \), go to 8. Else continue.

Step 6]
If \( j > 1 \), set \( j = j-1 \) and go to 2.
Else continue.

Step 7]
Set \( z = \hat{x} \), \( \bar{g} = g(\hat{x}) - 1 \) and go to 9.

Step 8]
If \( j = p \), go to 13.
Else continue.
Step 9]
   Set \( j = j+1 \)

Step 10]
   If \( d_{kj} = 0 \) (\( k = 1, \ldots, j-1 \)) go to 9.
   Else continue.

Step 11]
   \( \hat{x}_j = \hat{x}_j + e_j \).

Step 12]
   If \( \hat{x}_j < Q \), go to 4.
   Else go to 8.

Step 13]
   If \( z = \emptyset \), there is no solution.
   Else \( x^* = z \), \( g^* = g(x^*) \).

Stop.

If steps 4, 5 and 10 of the algorithm are deleted then all \( \mathbb{W} \) roots \( b_j \) (\( j \in V \)) would be computed. Moreover they would be computed in natural order \( b_{v_1}, \ldots, b_{v_{\mathbb{W}}} \). This property is verified by working through the steps of the algorithm.

At step 5, the following test is made. If for some root \( b_i \), the \( j \)th component has been computed so that \( (x^i, x^i, \ldots, x^i) \) is known and if \( \sum_{k=1}^{p} c x^i > \bar{g} \), then \( b_i \) is deleted. Also by property 2 we can delete all roots having \( x_j > x^i \) and \( x_k = x^i \) (\( k = j+1, \ldots, p \)).

At step 10 if \( D_j^j \) is all zeros except for \( d_{jj} \), then (by property 3) for any \( i \) we can delete all roots with the
property that \( x_j > x_j \) and \( x_k = x_k \) (\( k = j+1, \ldots, p \)).

If step 7 is reached then a feasible root \( b_i = \hat{x} \) has been found. By property 1 we may reset \( \bar{g} = g(b_i) - 1 \) and continue until, ultimately, terminal step 13 must be reached.

5.5 Computational Considerations

The number of computations required by the Enumerative Algorithm is proportional to the number of roots which must be fully or partially evaluated.

The property that \( i < j < c_i \leq c_j \) is of significance for problems where the objective function coefficients are not all equal. For problems such as these, the rate at which roots are deleted from the search is related to the rate of increase of the coefficients. Therefore, the objective function coefficients play an important role in determining the computational cost of the algorithm.

We know that \( M = \) the maximum number of roots to be evaluated. In general, one would conclude that the larger the value of \( M \), the more calculations required by the search. That this is not necessarily so can be justified by the following observation. The larger the value of \( d_{ii} \) (that is the more values \( x_i \) can take on) the smaller the value of \( e_i \) and therefore the smaller the value of \( x_i \), the initial solution value of \( x_i \). If many or all the diagonal elements are large (resulting in a large value for \( M \)), then \( g(b_1) \), the objective function value of the initial solution, will be small. Obviously the smaller
the value of \( g(b_1) \), the faster the rate at which roots are deleted from the search. Similarly the smaller the value of the bound \( G \), the faster roots are deleted.

If problem 5.1.1 has a feasible solution, then the above lemmas, combined with the fact that the objective function coefficients are monotonically increasing, suggest that the initial solution, if not optimal, is at least a reasonably good approximation of the optimal root. For ILPC's which do not require an exact solution, it may be sufficient to bypass the Enumeration Algorithm and accept the initial solution \( b_1 \) as the optimal solution to 5.1.1 provided \( g(b_1) \leq G \). This variation of the algorithm will be discussed further in Chapter 8.

5.6 Proof of Lemmas

Lemma 5.2.1

Choose any \( j = 1, \ldots, M \) such that \( j \notin V \).

Denote \( b_j \) by

\[
\begin{pmatrix}
\bar{x}_1 \\
\vdots \\
\bar{x}_p
\end{pmatrix}
\]

Then the congruence given by:

\[
d_{11} x_1 \equiv v_1 - \sum_{j=2}^{P} d_{1j} \bar{x}_j \pmod{Q}
\]

has exactly \( d_{11} \) roots (Prop. 2.1.6). Thus the system of congruences given by 5.1.1 has exactly \( d_{11} \) roots with components

\[
x_j = \bar{x}_j \quad (j = 2, \ldots, p).
\]

Denote these by:
\[ b_i = \begin{pmatrix} x_1^i \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_p \end{pmatrix} \quad b_{i+1} = \begin{pmatrix} x_1^{i+1} \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_p \end{pmatrix} \quad b_{i+d_{11}-1} = \begin{pmatrix} x_1^{i+d_{11}-1} \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_p \end{pmatrix} \]

where

\[ x_1^i < x_1^{i+1} \ldots < x_1^{i+d_{11}-1} \]

Note that \( j = i + k \) for some \( k \geq 0 \).

It is easily seen from the definition of \( V \), that \( i \in V \Rightarrow k > 0 \) since \( j \not\in V \).

But \( i < j \) and \( g(b_i) \leq g(b_j) \Rightarrow b_j \neq x^* \).

**Lemma 5.3.1**

Proof is by backwards induction starting with \( i = p \). We know the \( p \)th congruence of the system is solvable so by Prop. 2.1.6 the congruence

\[ x_p \equiv v_p / dp \mod e_p \]

has a unique root \( \tilde{x}_p \) with \( 0 \leq \tilde{x}_p < e_p \).

By definition of the ordering of the sequence \( \{ b_i \} \)

\[ x_{p1} = \tilde{x}_p. \]

Assume \( 0 \leq x_j < e_j \) for \( j = i+1, \ldots, p \).

The \( i \)th congruence of the system may be written:

\[ x_i \equiv (v_i - \sum_{j=i+1}^{p} d_{ij} x_j^1) / d_{ii} \mod e_i. \]

Since this congruence is solvable it has a unique solution \( \tilde{x}_i \) such that \( 0 \leq \tilde{x}_i < e_i \). Again by definition of \( b_{1i} \), \( x_i^1 \) must equal \( \tilde{x}_i \) and the inductive hypothesis is proved.
Lemma 5.3.2

By Lemma 4.1.5

\[
b_k = \begin{pmatrix} x_1^k \\ \vdots \\ x_p^k \end{pmatrix} = \begin{pmatrix} x_1^1 \\ \vdots \\ x_p^1 \end{pmatrix} + Ey, \ y \text{ integer}
\]

where \( E \in T(p) \) and \( e_{ii} = e_i = Q/d_{ii} + \)

\[
x_k^p = x_1^p + e_p y_p, \ y_p \text{ integer.}
\]

But \( x_k^p \geq 0 \) and \( 0 \leq x_1^p < e_p + y_p \geq 0 \).

If \( y_p > 0 \) we are done. Else \( y_p = 0 \) and \( x_k^p = x_1^p \).

Assume \( x_j^k = x_j^1 \) for \( j = i+1, \ldots, p \) \(
\)

\[
y_j = 0 \text{ for } j = i+1, \ldots, p.
\]

Then

\[
x_i^k = x_i^1 + e_i y_i, \ y_i \text{ integer.}
\]

Again \( x_i^k \geq 0 \) and \( 0 \leq x_i^1 < e_i + y_i \geq 0 \).

If \( y_i > 0 \), we are done. Else \( y_i = 0 \) and \( x_i^k = x_i^1 \).

Since \( b_k \neq \begin{pmatrix} x_1^1 \\ \vdots \\ x_p^1 \end{pmatrix} \), this process cannot continue for all \( i \).

Therefore there must exist some \( i \) between 1 and \( p \) such that

\[
x_i^k = x_i^1 + e_i y_i, \ y_i > 0, \text{ integer and}
\]

\[
x_j^k = x_j^1 \text{ for } j = i+1, \ldots, p.
\]

Lemmas 5.3.3 and 5.3.4

Let \( b_k \) be any root of the system of congruences. Then by Lemma 5.3.2, there exists some \( i \) between 1 and \( p \) such that
\[ g(b_k) = \frac{i-1}{2} \sum_{j=1}^{i-1} c_{x_j} + c_i(x_i^1 + y_1 e_i) + \frac{p}{2} \sum_{j=i+1}^{p} c_{j} \frac{x_j^1}{j} \]

\[ \geq y_i(c_i e_i) + \sum_{j=i}^{p} c_j x_j^1 \quad \text{where } y_i > 0, \text{ integer.} \]

Assume \( b_k \) is optimal \( \implies g(b_k) < g(b_i) \)

\[ y_i(c_i e_i) < g(b_i) - \sum_{j=i}^{p} c_j x_j^1 \]

\[ = \sum_{j=1}^{i-1} c_j x_j^1. \]

If \( c_i = 0 \implies c_j = 0 \) for \( j = 1, \ldots, i-1 \) \( \implies 0 < 0. \)

This contradiction implies \( c_i > 0 \implies c_i e_i > 0 \implies \]

\[ y_i < \sum_{j=1}^{i-1} c_j x_j^1 / c_i e_i. \quad 5.6.1 \]

By the assumption of lemma 5.3.4

\[ \left( \sum_{j=1}^{i-1} c_j x_j^1 \right) / c_i e_i \leq 1 \implies \]

\[ y_i > 0, \text{ integer and } y_i < 1. \]

This contradiction implies that \( b_k \) cannot be optimal.

By the assumption of lemma 5.3.3

\[ \left( \sum_{j=1}^{i-1} c_j e_j \right) / c_i e_i \leq 1 \]

But by lemma 5.3.1 \( x_j^1 < e_j \) for all \( j \implies \]

\[ \left( \sum_{j=1}^{i-1} c_j x_j^1 \right) / c_i e_i \leq 1 \implies (\text{by } 5.6.1) \]

\[ y_i > 0, \text{ integer and } y_i < 1. \]

This contradiction implies that \( b_k \) cannot be optimal.

5.7 Numerical Examples

Example 1: Consider example 2 of Chapter 3.
Min \( g(x) = 3x_1 + 4x_2 \)

\[
\begin{pmatrix}
5 & 1 \\
0 & 2
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
\equiv
\begin{pmatrix}
9 \\
8
\end{pmatrix} \pmod{10}
\]

\( g(x) \leq \infty \)

\( x \geq 0, \text{ integer} \)

Then

\[
\begin{pmatrix}
e_1 \\
5
\end{pmatrix}
= \begin{pmatrix}
2 \\
5
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
c_i e_1 \\
20
\end{pmatrix}
= \begin{pmatrix}
6 \\
20
\end{pmatrix}
\]

and

\( c_1 e_1 < c_2 e_2 \).

Lemma 5.3.3 is satisfied and for any value of \( v \), the initial solution will be optimal. In particular for

\[
v = \begin{pmatrix}
9 \\
8
\end{pmatrix}, \quad b_1 = x^* = \begin{pmatrix}
1 \\
4
\end{pmatrix} \quad \text{and} \quad g^* = 19.
\]

Example 2: Consider example 3 of Chapter 3.

Min \( g(x) = x_1 + x_2 + 7x_3 \)

\[
\begin{pmatrix}
2 & 1 & 4 \\
0 & 2 & 10 \\
0 & 0 & 12
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
\equiv
\begin{pmatrix}
23 \\
6 \\
0
\end{pmatrix} \pmod{24}
\]

\( g(x) \leq \infty \)

\( x \geq 0, \text{ integer} \)

Then

\[
\begin{pmatrix}
e_1 \\
12
\end{pmatrix}
= \begin{pmatrix}
12 \\
12 \\
2
\end{pmatrix}
\quad \text{and} \quad
\begin{pmatrix}
c_i e_1 \\
14
\end{pmatrix}
= \begin{pmatrix}
12 \\
14
\end{pmatrix}
\]
Then \( c_1 e_1 < c_2 e_2 \), but \( c_1 e_1 + c_2 e_2 > c_3 e_3 \) so Lemma 5.3.3 is not satisfied.

However the initial solution is given by
\[
b_1 = (x_1^1) = \begin{pmatrix} 10 \\ 3 \\ 0 \end{pmatrix}
\]
and so \((c_1 x_1^1) = \begin{pmatrix} 10 \\ 3 \\ 0 \end{pmatrix}\)

Therefore
\[
c_1 x_1^1 < c_2 e_2
\]
and \( c_1 x_1^1 + c_2 x_1^1 < c_3 e_3 \)
so Lemma 5.3.4 is satisfied.

Therefore for \( v = \begin{pmatrix} 23 \\ 6 \\ 0 \end{pmatrix} \), the initial solution is
\[
x^* = b_1 = \begin{pmatrix} 10 \\ 3 \\ 0 \end{pmatrix}
\]
optimal and
\[
g^* = g(b_1) = 13.
\]

Example 3: The following example was chosen to illustrate the algorithm because it is representative of a worst case in the sense of having no variation at all in the objective function coefficients. Also \( G = \infty \).

\[
\text{Min } g(x) = \sum_{i=1}^{6} x_i
\]
\[
\begin{pmatrix}
1 & 0 & 1 & 1 & 1 & 0 \\
0 & 3 & 1 & 1 & 1 & 0 \\
0 & 0 & 2 & 2 & 0 & 1 \\
0 & 0 & 0 & 3 & 1 & 1 \\
0 & 0 & 0 & 0 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 2 \\
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
\end{pmatrix}
\equiv
\begin{pmatrix}
5 \\
5 \\
1 \\
9 \\
9 \\
6 \\
\end{pmatrix}
\pmod{12}
\]

\[g(x) \leq \infty\]
\[x \geq 0, \text{ integer}\]

Note that \(M = \overline{M} = 72\) and there are 72 roots each with 6 components.

\[
\begin{pmatrix}
c_i & e_i \\
\end{pmatrix}
= \begin{pmatrix}
12 \\
4 \\
6 \\
4 \\
6 \\
\end{pmatrix}
\quad \text{and} \quad
b_1 = \begin{pmatrix}
9 \\
3 \\
4 \\
1 \\
3 \\
\end{pmatrix}
\]

It is easily verified that neither lemma 5.3.3 nor lemma 5.3.4 is satisfied and so the Enumerative Algorithm is applied. The following table summarizes the computations of the algorithm. A blank entry indicates no change from the previous column. A dashed entry indicates the component is not computed because that root can be deleted. The optimal solution is determined after 15 roots are partially or fully evaluated and 41 components computed.
<table>
<thead>
<tr>
<th>Iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>x6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>x5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.7.1
CHAPTER 6

PRIMAL-DUAL OPTIMIZATION

6.1 Statement of the Problem

As shown in Chapter 1, the ILPC can be stated as:

\[
\text{Min } f = \sum_{i=1}^{n} c_i x_i \\
Ax \equiv b \pmod Q \\
x \geq 0, \text{ integer.}
\]

From Chapter 3, we know that if the problem is solvable, the variables may be partitioned \( x = (x^S, x^T) \) with a corresponding partition of the objective function coefficients \( c = (c^S, c^T) \) and the ILPC may be equivalently written as:

\[
\text{Min } f = \sum_{i=1}^{P} c^S_{s_i} x^S_{s_i} + \sum_{i=1}^{n-P} c^T_{t_i} x^T_{t_i} \\
Dx^S \equiv d - Kx^T \pmod Q \\
x \geq 0, \text{ integer}
\]

where \( D, K, \) and \( d \) are as given by the Triangularization Theorem and for any integer vector \( x^T \), the system is solvable for \( x^S \).

Without loss of generality we may assume that \( c \) is an integer vector and the sequence \( S = (1, 2, \ldots, p) \).
Define

\[ g(x^S) = \sum_{i=1}^{p} c_i x_i \]

and

\[ h(x^T) = \sum_{i=p+1}^{n} c_i x_i \]

Note that by 3.8.1 we have that

\[ 1 \leq i < j \leq p \text{ if } 0 \leq c_i \leq c_j \]

and

\[ p+1 \leq i < j \leq n \text{ if } 0 \leq c_i \leq c_j \]

The ILPC may now be stated as follows:

\[
\begin{align*}
\text{Min } f &= g(x^S) + h(x^T) \\
Dx^S &= d - Kx^T \pmod{Q} \\
x &\geq 0, \text{ integer.}
\end{align*}
\]

It is the purpose of this chapter to describe an algorithm for finding an optimal solution to 6.1.1. If the optimal solution is not unique, then the algorithm will determine an optimal representative \( x^* = (x^S^*, x^T^*) \) with the following property:

**Property 6.1.2**

If \( \hat{x} = (\hat{x}^S, \hat{x}^T) \) is also optimal, then

either:

\[ h(x^{T^*}) < h(\hat{x}^T) \]

or

\[ h(x^{T^*}) = h(\hat{x}^T) \text{ and } x^{T^*} > L \hat{x}^T \]

where

\[ y > L_z \text{ iffi the first nonzero component of } y - z \text{ is positive.} \]
6.2 Preliminary Analysis of the Algorithm

The algorithm consists of two main phases and two termination criteria.

Phase 1:

a) Define a problem

\[
\begin{align*}
\text{Min } h(x^T) \\
Kx^T &\equiv d \pmod{Q} \\
x^T &\geq 0, \text{ integer}
\end{align*}
\]

and using Glover's Method of Integer Programming Over a Finite Additive Group (see Appendix A) generate a sequence \( W \) of nonnegative integer vectors:

\[ x^T(i) = (x^{i}_{p+1}, \ldots, x^{i}_{n}). \]

b) For each vector \( x^T(i) \) compute:

1) a cost \( h(i) = \sum_{j=p+1}^{n} c_{j}x^{i}_{j} \)

and

2) a group element

\[ K(i) = Kx^T(i) \text{ reduced modulo } Q. \]

As stated in Appendix A, the sequence \( W \) has the following properties:

i) \( p \neq q \Rightarrow x^T(p) \neq x^T(q) \)

ii) \( p < q \Rightarrow h(p) < h(q) \) or

\[ h(p) = h(q) \text{ and } x^T(p) \leq x^T(q). \]

iii) \( K(p) = d \) for some \( x^T(p) \) iffi 6.2.1 has a feasible solution.

iv) \( x^T(p) \) is optimal for 6.2.1 iffi \( K(p) = d \) and
p < q for any q such that K(q) = d.

v] Unless the sequence is terminated, it will consist of all the distinct nonnegative integer vectors having (n-p) components.

Phase 1 begins with

\[ x^T(1) = K(1) = \theta, \text{ the (n-p) vector of zeros and } h(1) = 0. \]

Each time an element of the sequence W is generated, the algorithm will either terminate or execute Phase 2.

**Phase 2:**

a] For each \( x^T(i) \) compute:

1] \( v(i) = d - K(i) \) reduced modulo Q and

2] \( G(i) = F(i-1) - h(i) - 1 \)

where \( F(0) = \infty \).

b] Using the Enumerative Algorithm of Chapter 5, compute \( x^S(i) \), the solution to the problem:

\[
\begin{align*}
\text{Min } g(x^S) \\
Dx^S & \equiv v(i) \pmod{Q} \\
x^S & \geq 0, \text{ integer} \\
g(x^S) & \leq G(i)
\end{align*}
\]

6.2.2

c] Update \( F(i) \) as follows:

If 6.2.2 has no solution, set \( F(i) = F(i-1) \).

If 6.2.2 is solvable, let \( x(i) = (x^S(i), x^T(i)) \) and

\[ F(i) = g(x^S(i)) + h(x^T(i)) = f(x(i)). \]

To understand Phase 2 we note that by the Triangularization
Theorem of Chapter 3, for any vector \( x^T(i) \) the system:
\[
Dx^S \equiv v(i) \pmod{Q}
\]
\( x^S \geq 0 \), integer
has at least one solution \( \hat{x}^S \) such that \((\hat{x}^S, x^T(i))\) is a feasible solution to the ILPC given by 6.1.1. Clearly if \( \hat{x}^S \) is any other solution such that
\[
g(\hat{x}^S) < g(\hat{x}^S),
\]
\[
f(\hat{x}^S, x^T(i)) < f(\hat{x}^S, x^T(i)).
\]
Therefore for any given vector \( x^T(i) \), the only candidate for \( x^{S*} \) can be a solution to the problem:
\[
\text{Min } g(x^S)
\]
\[
Dx^S \equiv v(i) \pmod{Q}
\]
\( x^S \geq 0 \), integer.

The bound \( G(i) \) is justified by the following analysis.
Initially \( i = 1 \) with \( x^T(1) = \emptyset \) and \( G(1) = \infty \). Problem 6.2.2 is therefore solvable with solution \( x^{S}(1) \). The vector \( x(1) = (x^{S}(1), \emptyset) \) is a feasible solution to the ILPC and \( f(x(1)) = F(1) \) is an upper bound on the objective function \( f(x) \). When \( i = 2 \), \( x^T(2) \) can be a candidate for \( x^{T*} \) only if
\[
g(x^S) + h(2) \leq f(x(1)) = F(1).
\]
By the definition of the optimal representative \( x^* \), the equality cannot hold since \( h(2) > h(1) = 0 \Rightarrow
\]
\[
g(x^S) \leq F(1) - h(2) - 1 = G(2).
\]
If no solution to 6.2.2 exists then \( x^T(2) \neq x^{T*} ; F(2) = F(1) \) and \( x(1) \) is the best feasible solution computed so far. If a solution to 6.2.2 exists then \( x(2) = (x^{S}(2), x^T(2)) \) is a
better feasible solution to the ILPC than \( x(1) \) and \( F(2) = f(x(2)) \) is a new upper bound on the objective function \( f(x) \).

Clearly \( F(j) \) is a monotonically decreasing function and for any given value of \( j \), \( F(j) = f(x(m)) \) simply indicates that \( x(m) \) is the best feasible solution to the ILPC computed so far, that is for the first \( j \) iterations of the algorithm.

The following lemma formally justifies the bound \( G(i) \).

**Lemma 6.2.1**

For any \( i \geq 2 \) let

\[
F(i-1) = f(x(m)) \quad (1 \leq m \leq i-1)
\]

and \( x^T(i) \) and \( h(i) \) be given.

Then if 6.2.2 has no solution, \( x^T(i) \neq x^{T*} \).

By property (v) of the sequence \( W \), there exists some positive integer \( k \) such that \( x^T(k) = x^{T*} \). By Lemma 6.2.1 when \( i = k \), problem 6.2.2 has a solution \( x^S(k) \). Clearly \( x^S(k) = x^{S*} \) and \( x^* = (x^S(k), x^T(k)) \).

**Termination Criteria:**

At the end of Phase 1 the vectors \( x^T(i) \) and \( K(i) \), and a cost \( h(i) \) have been generated. Then if either:

a] \( h(i) \geq F(i-1) \) or

b] \( h(i) < F(i-1) \) and \( x^T(i) \) is a feasible solution to problem 6.2.1

the algorithm terminates.

The following two theorems justify these criteria.
Theorem 6.2.2

If at some stage \( i \), \( h(i) \geq F(i-1) = f(x(m)) \), then
\[ x^* = x(m) \quad \text{and} \quad f(x^*) = F(i-1). \]

Theorem 6.2.3

If at some stage \( i \), \( h(i) < F(i-1) \) and \( x^T(i) \) is a feasible solution to 6.2.1, then:
\[ x^{S*} = \emptyset \]
\[ x^{T*} = x^T(i) \]
and \( f(x^*) = h(i) \).

We note it is trivially true that \( x^T(i) \) is feasible for 6.2.1 if and only if \( (\emptyset, x^T(i)) \) is feasible for the ILPC.

The term Primal-Dual Algorithm is justified by the following observations.

At each iteration \( i (i \geq 1) \) we have:
\[ F(i) = f(x(m)) \quad (1 \leq m \leq i) \]
where \( x(m) \) is a feasible solution to the ILPC and \( F(i) \) is a monotonically decreasing function of \( i \) representing the current upper bound on the objective function \( f(x) \).

By Theorem 6.2.3 we see that at each iteration \( i \) prior to termination, we have a vector \( (\emptyset, x^T(i)) \) which would be optimal for the ILPC if it were feasible.

Moreover,
\[ h(i) = f(\emptyset, x^T(i)) \]
is a monotonically increasing function of \( i \) representing the current lower bound on the objective function \( f(x) \).

We may now state the Primal-Dual Algorithm.
6.3 Primal-Dual Algorithm

1] Initialize $F(0) = \infty$, $i = 1$.

2] Generate $x^T(i)$, $h(i)$ and $K(i)$ using Glover's Algorithm on problem 6.2.1.

3] If $h(i) \geq F(i-1) = f(x(m))$, set $x^* = x(m)$ and go to 11. Else go to 4.

4] If $x^T(i)$ is feasible for 6.2.1, set $x^* = (\theta, x^T(i))$ and go to 11. Else go to 5.

5] Compute $v(i) = d - K(i)$ reduced modulo $Q$.

6] Compute $G(i) = F(i-1) - h(i) - 1$.

7] Using the Enumerative Algorithm of Chapter 5 solve the problem:

$$\begin{align*}
\text{Min } & g(x^S) \\
& Dx^S \equiv v(i) \pmod Q \\
& x^S \geq 0, \text{ integer} \\
& g(x^S) \leq G(i).
\end{align*}$$

8] If $x^S(i)$ is the solution, then:

set $x(i) = (x^S(i), x^T(i))$

set $F(i) = f(x(i))$

store $x(i)$ and go to 10.

Else go to 9.

9] Set $F(i) = F(i-1)$ and go to 10.

10] Set $i = i + 1$ and go to 2.

11] $f^* = f(x^*)$.

Stop.
6.4 A Special Case of the ILPC

In the preceding sections it was assumed that:

\[ x^S = (x_1, x_2, \ldots, x_p) \]

and \( x^T = (x_{p+1}, \ldots, x_n). \)

However it is possible that

\[ x^S = x = (x_1, x_2, \ldots, x_n) \]

and the ILPC given by 6.1.1 reduces to:

\[
\begin{align*}
\text{Min } f(x) &= g(x) \\
Dx &\equiv d \pmod{Q} \\
x &\geq 0, \text{ integer.}
\end{align*}
\]

In such a case the Primal-Dual Algorithm simply reduces to a single pass through the Enumerative Algorithm of Chapter 5.

6.5 A Screening Procedure for the Algorithm

Let the ILPC be as given by 6.1.1. Before executing the Primal-Dual Algorithm, it may be possible to determine if for some \( i = 1, \ldots, n-p, \) \( x_{t_i}^* = 0. \) The following theorem gives a sufficient condition for \( x_{t_i}^* = 0. \)

Theorem 6.5.1

If the system given by:

\[
\begin{align*}
Dx^S &\equiv K^i \pmod{Q} \\
x^S &\geq 0, \text{ integer} \\
g(x^S) &\leq c_{t_i}^* \\
is solvable, \text{ then } x_{t_i}^* &= 0.
\end{align*}
\]

The Enumerative Algorithm of Chapter 5 may be used to solve the problem given by:
Min \ g(x^S)
Dx^S \equiv K_i \quad (\text{mod } Q)
\begin{align*}
x^S & \geq 0, \text{ integer} \\
g(x^S) & \leq c_{t_i}.\end{align*}

Then if the problem has a solution we may set \( x_{t_i} = 0 \) and remove \( t_i \) from the sequence \( T \). While this screening procedure is not an essential part of the Primal-Dual Algorithm, it may be a useful tool especially for problems with wide variations in the objective function coefficients.

### 6.6 Finiteness of the Primal-Dual Algorithm

To show finiteness we must first prove the following lemma.

**Lemma 6.6.1**

If \( c_{t_j} = 0 \) for any \( t_j \in T \), then \( x^*_j = 0 \) and \( t_j \) may be removed from the sequence \( T \).

We may now assert that for all \( t_j \in T \), \( c_{t_j} > 0 \).

Finally we may state the finiteness of the Primal-Dual Algorithm.

**Theorem 6.6.2**

If problem 6.2.1 has no feasible solution, then at some stage \( i \) of the Primal-Dual Algorithm:

\[ h(i) \geq F(i-1) \]

and the algorithm terminates.

### 6.7 Proof of Theorems

**Lemma 6.2.1**

By the Triangularization Theorem, 6.2.2 has no solution
if and only if \( g(x^S) > G(i) \) for all \( x^S \geq 0 \), integer such that \((x^S, x^T(i))\) is feasible for the ILPC:

\[
g(x^S) > F(i-1) - h(i) - 1 \Rightarrow \\
g(x^S) + h(i) \geq F(i-1).
\]

Case a] If \( g(x^S) + h(i) > F(i-1) = f(m) \Rightarrow x(m) \) is a better feasible solution than \((x^S, x^T(i))\).

Case b] If \( g(x^S) + h(i) = F(i-1) = f(m) \), then since \( m < i \Rightarrow \) (by Prop. ii of the sequence \( W \))

either \( h(m) < h(i) \)
or \( h(m) = h(i) \) and \( x^T(m) \leq_L x^T(i) \)

(by definition of the optimal representative \( x^* \))

if \( f(x(m)) = f(x^S, x^T(i)) = f(x^*) \) then \( x(m) = x^* \).

In both cases it is shown that \( x^T(i) \neq x^T* \).

**Theorem 6.2.2**

If \( h(i) \geq F(i-1) \), then

\[
G(i) = F(i-1) - h(i) - 1 < 0.
\]

But \( F(j) \) is a monotonically decreasing function of \( j \), and (by Prop. ii of \( W \)) \( h(j) \) is a monotonically increasing function of \( j \)

\[
G(j) = F(j-1) - h(j) - 1
\]

is a monotonically decreasing function of \( j \)

\[
G(j) \leq G(i) < 0 \quad \text{for all} \quad j > i.
\]

But for all \( x^S \geq 0 \), \( g(x^S) \geq 0 \Rightarrow \) for all \( j \geq i \) problem 6.2.2 has no solution (by Lemma 6.2.1)

\[
x^T(j) \neq x^T* \quad \text{for any} \quad j \geq i \quad \Rightarrow \\
x^T* = x^T(j) \quad \text{for some} \quad j \leq i-1.
\]
But \( F(i-1) = f(x(m)) \Rightarrow x(m) \) is the best feasible solution for the first \((i-1)\) iterations →

\[
\begin{align*}
    x^{T*} &= x^T(m) \\
    x^{S*} &= x^S(m) \\
    x^* &= x(m)
\end{align*}
\]

and \( f(x^*) = f(x(m)) = F(i-1) \).

**Theorem 6.2.3**

If \( h(i) < F(i-1) \) then

\[
G(i) = F(i-1) - h(i) - 1 \geq 0.
\]

If \( x^T(i) \) is a feasible solution to 6.2.1 →

\[
Kx^T(i) \equiv d \pmod{Q} \Rightarrow
\]

\[
v(i) = 0 \Rightarrow
\]

problem 6.2.2 is solved by \( x^S(i) = \emptyset \) →

\[
x(i) = (\emptyset, x^T(i)) \Rightarrow
\]

\[
F(i) = h(i).
\]

By Prop. ii of the sequence \( W, h(i+1) \geq h(i) = F(i) \) →

(by Theorem 6.2.2)

\[
x^* = x(i) \text{ and}
\]

\[
f(x^*) = F(i) = h(i).
\]

**Theorem 6.5.1**

Let \( x^S \) be any solution to the system and let

\[
x^* = (x^{S*}, x^{T*}) \text{ be the optimal representative.}
\]

Assume \( x^t_i > 0 \). Since \( x^* \) feasible →

\[
Dx^{S*} + \sum_{j=1}^{r} K_j x^*_j + K_i x^*_i \equiv d \pmod{Q}
\]

By hypothesis \( D\overline{x}^S \equiv K^1 \pmod{Q} \) → (by Prop. 2.1.3)
\[ D^{x^*}_x t_i \equiv K^i x^* (\mod Q) \rightarrow (\text{by Props. } 2.1.2 \text{ and } 2.1.1) \]

\[ D x^* + \sum_{j=1, j \neq i}^r K^j x^* t_j + D(\overline{x}^* x^* t_i) \equiv d (\mod Q) \rightarrow \]

\[ D(x^* + \overline{x}^* x^* t_i) + \sum_{j=1, j \neq i}^r K^j x^* t_j \equiv d (\mod Q). \]

Define \( \hat{x}^S = (x^* + \overline{x}^* x^* t_i) \) and note \( \hat{x}_{\overline{s}_i} \geq 0 \) (i=1,...,p).

\[ \hat{x}^T = \begin{cases} x^* & (j=1,...,r, j \neq i) \\ t_i & (j=i) \end{cases} \]

Then \((\hat{x}^S, \hat{x}^T)\) is feasible and

\[ g(\hat{x}^S) = g(x^* + x^* t_i, \overline{x}^*) \]

\[ h(\hat{x}^T) = h(x^* t_i) - c^t_i x^* \]

\[ h(\hat{x}^T) \leq h(x^* t_i) \quad \text{and} \quad h(\hat{x}^S, \hat{x}^T) = f(x^* t_i, \overline{x}^*) + [x^* t_i, c^t_i x^*]. \]

But by hypothesis

\[ 0 \leq g(\overline{x}^S) \leq c^t_i \quad (\text{since } x^* \geq 0) \]
\[ [x^* t_i, \overline{x}^* - c^t_i x^* \leq 0 \quad \implies \]
\[ f(x^* t_i, \hat{x}^T) \leq f(x^*, x^* t_i). \]

But this contradicts the definition of the optimal representative \( x^* \) (see 6.1.2). Therefore \( x^* t_i = 0. \)

**Lemma 6.6.1**

Assume there exists some \( t_j \in T \) such that \( c_{t_j} = 0. \)

Then either:

a) \( c_{s_j} = 0 \) for all \( s_j \in S \Rightarrow x^* t_j = 0, \) \( f^* = 0 \) and \( x^* t_j = 0. \)

or

b) there exists a positive integer \( i \) such that

\[ 0 = c_{s_1} = c_{s_2} = \cdots = c_{s_{i-1}} = c_{t_j} < c_{s_i} \leq \cdots \leq c_{s_p}. \]
But by definition (see 3.8.1) this implies that
\[ t_j < s_k \text{ for } k = i, \ldots, p. \rightarrow (\text{by Lemma 3.10.1}) \]
\[ K_{k,j} = 0 \text{ for } k = i, \ldots, p. \]

By Lemma 4.1.2 the system given by:
\[
Dx^S \equiv K^j \pmod{Q} \\
x^S \geq 0, \text{ integer}
\]
is solvable.

By back substitution it can be seen that this system has a solution denoted by:
\[
\bar{x}^S = (\bar{x}_{s_1}, \bar{x}_{s_2}, \ldots, \bar{x}_{s_{i-1}}, 0, \ldots, 0) \rightarrow \\
g(\bar{x}^S) = \sum_{i=1}^{p} c_i \bar{x}_{s_i} = 0 = c_{t_j} \rightarrow \\
(by \text{ Theorem 6.5.1}) \\
x^*_{t_j} = 0.
\]

**Theorem 6.6.2**

It has already been shown that

a) \( h(1) = 0 \)
b) \( h(j) \) is a monotonically increasing function of \( j \)
c) \( F(1) = \) a finite positive integer
d) \( F(j) \) is a monotonically decreasing function of \( j \).

If 6.2.1 has no feasible solution, the algorithm will not terminate until for some \( i \), \( h(i) \geq F(i-1) \). It is necessary only to show that for any \( j \) there exists a finite integer \( k > j \) such that \( h(k) > h(j) \).

By Prop. ii of the sequence \( W \) either:
\[
h(j+1) > h(j) \quad \text{or} \]
\[
h(j+1) = h(j) \text{ and } x^T(j) > L x^T(j+1).\]
By Lemma 6.6.1, $c_i > 0$ for $i \geq p+1$.

But

$$\sum_{i=p+1}^{n} c_i x_i = h(j)$$

can therefore have only a finite number of distinct solutions with $x_i \geq 0$, integer ($i = p+1, \ldots, n$). This is obvious since each $x_i$ is bounded above by $h(j)/c_i$. At some stage $k > j$, $h(k) > h(j)$.

Since $h(j)$ increases by integer amounts there exists a finite integer $i$ such that

$$h(i) \geq F(i) \geq F(i-1).$$
CHAPTER 7

EXAMPLES OF THE ALGORITHM

7.1 Problems With One Constraint (Knapsack Problems)

Consider the following special case of the ILPC:

\[
\begin{align*}
\text{Min } f(x) &= \sum_{i=1}^{n} c_i x_i \\
\sum_{i=1}^{n} a_i^t x_i &= a_0 \pmod{Q} \\
0 &\leq x_i < Q, \ x_i \text{ integer}
\end{align*}
\]

where \( i < j \) if and only if \( c_i \leq c_j \).

An ILPC in Smith Canonical Form will always be of this type whenever \( B \), an optimal LP basis matrix associated with the ILPC, has a determinant which is either a prime number or is the product of relatively prime factors (see 1.2.1).

A knapsack problem, especially one with a large number of variables, can be the most difficult type problem for the Primal-Dual Algorithm to solve because of the large number of feasible integer solutions.

Type 1: Problems having \( (a_1^t, Q) = 1 \).

From step 3 of the Triangularization Process (see 3.5), we see that if \( (a_1^t, Q) = 1 \), the new constraint row will be all zeros and the Triangularization Process terminates.
Therefore if \((a'_1, Q) = 1\), the triangularized system equivalent to 7.1.1 is given by:

\[
\begin{align*}
\text{Min } f(x) &= (c_1x_1) + \sum_{i=2}^{n} c_ix_i \\
x_1 &\equiv a_0 - \sum_{i=2}^{n} a_ix_i \pmod{Q} \\
0 &\leq x_i < Q, x_i \text{ integer}
\end{align*}
\]

7.1.2

where:

\[D = (1)\]

\[K = (a_2, \ldots, a_n)\]

\[S = (1)\]

and \(T = (2, \ldots, n)\).

For any choice of the variables \((x_2, \ldots, x_n)\), the Enumerative Algorithm of Chapter 5 is trivially solved by:

\[x_1 = a_0 - \sum_{i=2}^{n} a_ix_i \text{ reduced modulo } Q.\]

Note that \(\vec{x} = (a_0, 0)\) is a feasible solution with \(f(\vec{x}) = c_1a_0\). If \(c_1 = 0\), then \(\vec{x} = (a_0, 0)\) and \(\vec{f} = 0\). Else we can assume \(c_i > 0\) for all \(i\) and \(0 \leq x_i \leq c_1a_0/c_i\). Therefore the amount of computation needed to solve the problem depends on the variation in the coefficients \(c_i\) and the value of \(a_0\).

The screening procedure for the Primal-Dual Algorithm (see Theorem 6.5.1) is reduced to the simple test:

- if \(c_1a_i \leq c_j\), then \(x_i^* = 0\) \((i = 2, \ldots, n)\).

Note that problems for which \(Q\) is a prime number will
always be of this type.

**Example 1:**

Consider example 1 of Chapter 3.

\[
\text{Min } f(x) = 3x_1 + 4x_3 + 7x_2 \\
8x_1 + 5x_3 + 3x_2 \equiv 6 \pmod{11} \\
0 \leq x_i < 11, \ x_i \text{ integer for all } i.
\]

Since \((a_i^1, Q) = (8, 11) = 1\), the problem in triangularized form will be given by 7.1.2. More specifically it has been shown in Chapter 3, that the triangularized form is:

\[
\text{Min } f(x) = (3x_1) + (4x_3 + 7x_2) \\
x_1 \equiv 9 - (10x_2 + 2x_3) \pmod{11} \\
0 \leq x_i < 11, \ x_i \text{ integer for all } i.
\]

The computations of the Primal-Dual Algorithm are summarized by the following table.

<table>
<thead>
<tr>
<th>i</th>
<th>(x^T(i))</th>
<th>(h(i))</th>
<th>(K(i))</th>
<th>(v(i))</th>
<th>(x^S(i))</th>
<th>(x(i))</th>
<th>(F(i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0,0)</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>(9,0,0)</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>(0,1)</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>(7,0,1)</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>(1,0)</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>(0,2)</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>(5,0,2)</td>
<td>23</td>
</tr>
<tr>
<td>5</td>
<td>(1,1)</td>
<td>11</td>
<td>1</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>(0,3)</td>
<td>12</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>(3,0,3)</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>(2,0)</td>
<td>14</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since \(K(7) = d\), the algorithm terminates at step 4 with:

\[x^* = (x_1^*, x_2^*, x_3^*) = (0, x^T(7)) = (0,2,0) \quad \text{and } f^* = 14.\]
Example 2:

\[
\begin{align*}
\text{Min } f(x) &= x_1 + 2x_2 + 7x_3 + 10x_4 + 20x_5 \\
& \quad + 7x_1 + 3x_2 + 96x_3 + 56x_4 + 15x_5 \equiv 1 \pmod{113} \\
& \quad 0 \leq x_i < 113, x_i \text{ integer for all } i.
\end{align*}
\]

Since \((a_i', Q) = (7, 113) = 1\), the problem in triangularized form will be given by 7.1.2.

By applying the Equivalence Algorithm it is easily seen that the problem becomes:

\[
\begin{align*}
\text{Min } f(x) &= (x_1) + (2x_2 + 7x_3 + 10x_4 + 20x_5) \\
x_1 &\equiv 97 - 65x_2 - 46x_3 - 8x_4 - 99x_5 \pmod{113} \\
& \quad 0 \leq x_i < 113, x_i \text{ integer for all } i.
\end{align*}
\]

Applying the screening procedure we get:

\[
c_1a_4 = 8 \leq c_4 = 10 \rightarrow \\
x_4^* = 0.
\]

Therefore:

\[
S = (1)
\]

and

\[
T = (2, 3, 5).
\]

The following table summarizes the computations of the Primal-Dual Algorithm.

Note that on iteration 20, the algorithm terminates at step 3 since \(h(20) \geq F(19)\).

The optimal solution:

\[
x^* = (5, 0, 2, 0, 0)
\]

was calculated on iteration 13 with

\[
f^* = 19.
\]
Table 7.1.2

<table>
<thead>
<tr>
<th>i</th>
<th>$x^T(i)$</th>
<th>h(i)</th>
<th>K(i)</th>
<th>v(i)</th>
<th>G(i)</th>
<th>$x^S(i)$</th>
<th>F(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0,0,0)</td>
<td>0</td>
<td>0</td>
<td>97</td>
<td>$\infty$</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>2</td>
<td>(1,0,0)</td>
<td>2</td>
<td>65</td>
<td>32</td>
<td>94</td>
<td>32</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>(2,0,0)</td>
<td>4</td>
<td>17</td>
<td>80</td>
<td>29</td>
<td>--</td>
<td>34</td>
</tr>
<tr>
<td>4</td>
<td>(3,0,0)</td>
<td>6</td>
<td>82</td>
<td>15</td>
<td>27</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>(0,1,0)</td>
<td>7</td>
<td>46</td>
<td>51</td>
<td>13</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>(4,0,0)</td>
<td>8</td>
<td>34</td>
<td>63</td>
<td>12</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>(1,1,0)</td>
<td>9</td>
<td>111</td>
<td>99</td>
<td>11</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>8</td>
<td>(5,0,0)</td>
<td>10</td>
<td>99</td>
<td>111</td>
<td>10</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>(2,1,0)</td>
<td>11</td>
<td>63</td>
<td>34</td>
<td>9</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>10</td>
<td>(6,0,0)</td>
<td>12</td>
<td>51</td>
<td>46</td>
<td>8</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>(3,1,0)</td>
<td>13</td>
<td>15</td>
<td>82</td>
<td>7</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>(7,0,0)</td>
<td>14</td>
<td>3</td>
<td>94</td>
<td>6</td>
<td>--</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>(0,2,0)</td>
<td>14</td>
<td>92</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>19</td>
</tr>
<tr>
<td>14</td>
<td>(4,1,0)</td>
<td>15</td>
<td>80</td>
<td>17</td>
<td>3</td>
<td>--</td>
<td>19</td>
</tr>
<tr>
<td>15</td>
<td>(8,0,0)</td>
<td>16</td>
<td>68</td>
<td>29</td>
<td>2</td>
<td>--</td>
<td>19</td>
</tr>
<tr>
<td>16</td>
<td>(1,2,0)</td>
<td>16</td>
<td>44</td>
<td>53</td>
<td>2</td>
<td>--</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>(5,1,0)</td>
<td>17</td>
<td>32</td>
<td>65</td>
<td>1</td>
<td>--</td>
<td>19</td>
</tr>
<tr>
<td>18</td>
<td>(9,0,0)</td>
<td>18</td>
<td>20</td>
<td>77</td>
<td>0</td>
<td>--</td>
<td>19</td>
</tr>
<tr>
<td>19</td>
<td>(2,2,0)</td>
<td>18</td>
<td>109</td>
<td>101</td>
<td>0</td>
<td>--</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>(6,1,0)</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Type 2: Problems having \((a'_1, Q) > 1\).

When the Equivalence Algorithm is applied to knapsack problems having \((a'_1, Q) > 1\), the resulting problem will usually consist of more than one constraint.

Example 3:

Consider example 2 of Chapter 3.

\[
\begin{align*}
\text{Min } f(x) &= 3x_1 + 4x_2 + 5x_3 + 7x_4 \\
5x_1 + 9x_2 + 3x_3 + 4x_4 &\equiv 1 \pmod{10} \\
0 \leq x_i < 10, \ x_i \text{ integer for all } i.
\end{align*}
\]

Note that \((a'_1, Q) = (5, 10) = 5\).

As shown in Chapter 3, this problem may be written equivalently as:

\[
\begin{align*}
\text{Min } f(x) &= (3x_1 + 4x_2) + (5x_3 + 7x_4) \\
\begin{pmatrix} 5 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &\equiv \begin{pmatrix} 9 \\ 8 \end{pmatrix} - \begin{pmatrix} 7 & 6 \\ 4 & 2 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} \pmod{10} \\
0 \leq x_i < 10, \ x_i \text{ integer for all } i
\end{align*}
\]

where

\[
S = (1,2) \quad T = (3,4) \\
D = \begin{pmatrix} 5 & 1 \\ 0 & 2 \end{pmatrix} \quad K = \begin{pmatrix} 7 & 6 \\ 4 & 2 \end{pmatrix}
\]

From example 1 Chapter 5, we know that for any right hand side \(v \in G(D,10)\) the problem

\[
\begin{align*}
\text{Min } g(x^S) &= 3x_1 + 4x_2 \\
Dx^S &\equiv v \pmod{10} \\
0 \leq x_i^S < 10, \ x_i^S \text{ integer}
\end{align*}
\]

is optimized by the initial solution.
Applying the screening procedure (see Theorem 6.5.1) we see that the problem:

\[ \text{Min } g(x^S) = 3x_1 + 4x_2 \]
\[
\begin{pmatrix}
5 & 1 \\
0 & 2 \\
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\end{pmatrix}
\equiv
\begin{pmatrix}
6 \\
2 \\
\end{pmatrix} \pmod{10}
\]

\[ g(x^S) \leq c_4 = 7 \]

has an initial solution of \((1,1)\) with \(g(1,1) = 7\).

Therefore we can set \(x_4^* = 0\) and set \(T = (3)\).

The following table summarizes the computations of the Primal-Dual Algorithm.

Note that on iteration 4, the algorithm terminates at step 3 since \(h(4) \geq F(3)\).

The optimal solution:

\[ x^* = (0,2,1,0) \]

was calculated on iteration 2 with

\[ f^* = 13. \]

\[ \text{Table 7.1.3} \]

<table>
<thead>
<tr>
<th>i</th>
<th>(x^T(i))</th>
<th>h(i)</th>
<th>K(i)</th>
<th>v(i)</th>
<th>G(i)</th>
<th>x^S(i)</th>
<th>F(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>(0,0)</td>
<td>(9,8)</td>
<td>\infty</td>
<td>(1,4)</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>(7,4)</td>
<td>(2,4)</td>
<td>13</td>
<td>(0,2)</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>10</td>
<td>(4,8)</td>
<td>(5,0)</td>
<td>2</td>
<td>---</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.2 Problems with More Than One Constraint

Example 4:

Consider example 3 of Chapter 3.

\[
\begin{align*}
\text{Min } f(x) &= x_1 + x_2 + 7x_3 \\
4x_1 + 5x_2 + 11x_3 &\equiv 7 \pmod{12} \\
2x_1 + 3x_2 + 2x_3 &\equiv 5 \pmod{24} \\
0 &\leq x_i < 24, \ x_i \text{ integer for all } i
\end{align*}
\]

which may be written equivalently as

\[
\begin{align*}
\text{Min } f(x) &= x_1 + x_2 + 7x_3 \\
\begin{pmatrix} 2 & 1 & 4 \\ 0 & 2 & 10 \\ 0 & 0 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} &\equiv \begin{pmatrix} 23 \\ 6 \\ 0 \end{pmatrix} \pmod{24} \\
0 &\leq x_i < 24, \ x_i \text{ integer for all } i.
\end{align*}
\]

This is a special case (see 6.4) in which:

\[S = (1,2,3)\]

and \(T = \) the empty set,

so the Primal-Dual Algorithm reduces to a single iteration.

It was shown (see example 2, Chapter 5) that the initial solution \(x = (10,3,0)\) was optimal for this iteration and is therefore optimal for the entire problem with \(f^* = 13\).

Example 5:

Consider an ILPC in Smith Canonical Form.

\[
\begin{align*}
\text{Min } f(x) &= cx \\
A'x &\equiv b' \pmod{\delta} \\
x &\geq 0, \text{ integer}
\end{align*}
\]
where
\[ c = (1, 1, 2, 2, 3, 7, 8) \]
\[
A' = \begin{pmatrix}
7 & 9 & 21 & 17 & 3 & 23 & 0 \\
0 & 1 & 19 & 15 & 0 & 1 & 12 \\
0 & 5 & 16 & 34 & 1 & 45 & 1 \\
1 & 0 & 0 & 4 & 93 & 0 & 1
\end{pmatrix}
\]
\[
b' = \begin{pmatrix}
4 \\
3 \\
2 \\
4
\end{pmatrix}
\]

and
\[
\delta = \begin{pmatrix}
24 \\
24 \\
48 \\
96
\end{pmatrix}
\]

Multiplying rows 1 and 2 by \(96/24 = 4\) and row 3 by \(96/48 = 2\), we get the system

\[ Ax \equiv b \pmod{Q} \]

where
\[
A = \begin{pmatrix}
28 & 36 & 84 & 68 & 12 & 92 & 0 \\
0 & 4 & 76 & 60 & 0 & 4 & 48 \\
0 & 10 & 32 & 68 & 2 & 90 & 2 \\
1 & 0 & 0 & 4 & 93 & 0 & 1
\end{pmatrix}
\]
\[
\begin{pmatrix}
16 \\
12 \\
4 \\
4
\end{pmatrix}
\]

and \( Q = 96 \).

To determine if the system is solvable and, if so, to triangularize the matrix \( A \), we first apply the Equivalence Algorithm giving the system:

\[
Dx^S = d - Kx^T \pmod{96}
\]

where

\[
D = \begin{bmatrix}
1 & 0 & 0 & 4 & 21 & 13 \\
0 & 2 & 0 & 20 & 2 & 14 \\
0 & 0 & 4 & 28 & 20 & 12 \\
0 & 0 & 0 & 32 & 0 & 16 \\
0 & 0 & 0 & 0 & 24 & 12 \\
0 & 0 & 0 & 0 & 0 & 48
\end{bmatrix}
\]

\[
d = \begin{pmatrix}
64 \\
88 \\
4 \\
48 \\
60 \\
48
\end{pmatrix}
\]

\[
K = \begin{pmatrix}
24 \\
58 \\
48 \\
64 \\
24 \\
0
\end{pmatrix}
\]

\[
S = (1, 2, 3, 4, 5, 7)
\]

\[
T = (6)
\]

The following table summarizes the computations of the Primal-Dual Algorithm.
<table>
<thead>
<tr>
<th>i</th>
<th>$x^T(i)$</th>
<th>$h(i)$</th>
<th>$K(i)$</th>
<th>$v(i)$</th>
<th>$G(i)$</th>
<th>$x^S(i)$</th>
<th>$F(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0</td>
<td>64 88 4 60</td>
<td>0 0 60</td>
<td>0 0 60</td>
<td>0 0 60</td>
<td>0 0 60</td>
<td>5 25 5 2</td>
</tr>
<tr>
<td>2</td>
<td>1 7 24 24</td>
<td>40 30 52 36</td>
<td>58 48 64 24</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>48 --</td>
</tr>
<tr>
<td>3</td>
<td>2 14 48 48</td>
<td>16 68 4 12</td>
<td>20 32 48 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>41 --</td>
</tr>
<tr>
<td>4</td>
<td>3 21 72 72</td>
<td>88 10 52 84</td>
<td>78 48 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>34 --</td>
</tr>
<tr>
<td>5</td>
<td>4 28 0 0 0 0</td>
<td>64 48 0 0 0 0</td>
<td>40 64 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>27 --</td>
</tr>
<tr>
<td>6</td>
<td>5 35 24 24</td>
<td>40 86 52 36</td>
<td>2 48 32 24</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>20 --</td>
</tr>
</tbody>
</table>
Table 7.2.1 (continued)

<table>
<thead>
<tr>
<th>i</th>
<th>$x^T(i)$</th>
<th>$h(i)$</th>
<th>$K(i)$</th>
<th>$v(i)$</th>
<th>$G(i)$</th>
<th>$x^S(i)$</th>
<th>$F(i)$</th>
</tr>
</thead>
</table>
| 7 | \[
\begin{pmatrix}
48 \\
60 \\
0 \\
48 \\
0
\end{pmatrix}
\] & \[
\begin{pmatrix}
16 \\
28 \\
4 \\
48 \\
12
\end{pmatrix}
\] | 13 | -- | 56 |
| 8 | \[
\begin{pmatrix}
72 \\
22 \\
48 \\
64 \\
72 \\
0
\end{pmatrix}
\] & \[
\begin{pmatrix}
88 \\
66 \\
52 \\
80 \\
84 \\
48
\end{pmatrix}
\] | 6 | -- | 56 |
| 9 | 8 | 56 |

The algorithm terminates on the ninth iteration since $h(9) \geq F(8)$. The optimal solution is given by:
\[
x^* = (5, 25, 5, 1, 2, 0, 1)
\]
and $f^* = 56$
and was actually computed on the first iteration. On succeeding iterations, the Enumerative Algorithm quickly determines that the auxiliary problem:
\[
\text{Min } g(x^S)
\]
\[
Dx^S \equiv v(i) \pmod{96}
\]
\[
x^S \geq 0, \text{ integer}
\]
\[
g(x^S) \leq G(i)
\]
has no solution. Thus $x^S(i)$ is not defined for $i > 1$.

This problem, as most of the problems in this chapter, was arbitrarily generated.
CHAPTER 8

CONCLUDING REMARKS

8.1 Advantages of the Algorithm

By applying the Equivalence Algorithm to the ILPC, it is readily determined whether the problem is solvable and, if so, the problem is transformed to the format required by the Primal-Dual Algorithm.

The Primal-Dual Algorithm itself has the advantage that on each iteration a feasible solution to the original problem is known. That this is a reasonably good feasible solution is suggested by the initial ordering of the variables and the criterion by which indices are chosen to be candidates for the sequence $S$ (see 3.8.1).

Current group algorithms for solving the ILPC in Smith Canonical Form require computation time proportional to $D = \prod_{i=1}^{m} \delta_i$ (see 1.3). Only at termination do these algorithms ascertain whether a feasible solution even exists. The amount of computation time required by the Primal-Dual Algorithm is less dependent on $D$ and more dependent on a variety of factors such as the variation in the objective function coefficients and the number of variables and constraints. This is dramatically illustrated
by example 5 of Chapter 7 where \( D = \prod_{i=1}^{4} \delta_i = 2,654,208 \). But because the problem has four constraints and only seven variables, the Primal-Dual Algorithm can solve it easily in 9 iterations.

It should be noted that for a fixed number of variables as the number of non-trivial constraints increases both the number of roots to the system of congruences and the number of indices in the sequence \( T \) decrease. Thus the efficiency of the Primal-Dual Algorithm increases as the number of non-trivial problem constraints increases. This is in direct contrast to group type algorithms where an increase in the number of non-trivial constraints usually results in a significant increase in the size of \( D \) and thus in the amount of computation.

8.2 Modifications of the Procedure

The following are possible variations in the procedure for solving an ILPC as presented in this paper.

Variation 1

To apply the Equivalence Algorithm and the first iteration of the Primal-Dual Algorithm to an ILPC requires relatively little computation time. In return one can determine:

1) if the problem is solvable
2) a "good" feasible solution (and thus upper bounds on all variables with non-zero objective function coefficients)
and 3) a rough estimate of the computational efficiency of the Primal-Dual Algorithm. At this point the choice can be made to either continue with the Primal-Dual Algorithm or switch to some other scheme which may be able to utilize this information and solve the particular problem quickly.

**Variation 2**

It may be that an exact solution to the ILPC is not required. Then an approximate solution may be obtained by altering step 7 of the Primal-Dual Algorithm as follows (see 5.5):

1) Compute the initial solution \( x^S = b_1 \) to the system:

\[
Dx^S = v(i) \pmod{Q} \\
\]

\( x^S \geq 0 \), integer.

2) If \( g(b_1) \leq G(i) \) set \( x^S(i) = b_1 \), else leave \( x^S(i) \) undefined.

This eliminates the Enumerative Scheme described in Chapter 5 from each iteration of the Primal-Dual Algorithm. Thus computation is simplified. Also Lemma 5.3.2 guarantees that the initial solution is, if not optimal, at least a "good" approximation to the optimal solution of the auxiliary problem defined at step 7. It is easily seen that this approximation does not affect the finiteness of the Primal-Dual Algorithm, although at termination the "optimal" solution is an approximation also.
Variation 3]

The method of Glover summarized in Appendix A has an alternate version applicable to knapsack problems which generates at most D vectors x(i) (see Glover 1969).

If after the Equivalence Algorithm is applied a knapsack problem remains, then the alternate version of Glover's Method may be substituted at step 2 of the Primal-Dual Algorithm.

The advantage of this variation is that if two distinct vectors \( x^T(i) \) and \( x^T(j) \) \( (i < j) \) generate the same element \( v(i) = v(j) \), then \( x^T(j) \) is dropped. This helps reduce the number of iterations of the Primal-Dual Algorithm and guarantees that this number is less than D.

Two additional variations will be briefly mentioned as they will appear in detail in later papers.

Variation 4]

Replace the Enumerative Algorithm outlined in Chapter 5 with a more accelerated version based on the Double Description Theorem of Chapter 4.

Variation 5]

Modify the Primal-Dual Algorithm to solve the corresponding general integer linear programming problem (see 1.2).
APPENDIX A

GLOVER'S METHOD OF INTEGER PROGRAMMING
OVER A FINITE ADDITIVE GROUP

An abstract of Glover's Method as applied to solving the ILPC is presented.

The algorithm generates a sequence of nonnegative integer vectors \( x(i) = (x_1^i, \ldots, x_n^i) \) such that associated with each \( x(i) \) is a cost:
\[
C(i) = cx(i)
\]
and a group element:
\[
\alpha(i) \equiv Ax(i) \pmod{Q}
\]
where \( \alpha(i) \) is reduced modulo \( Q \) (see Def. 2.1.5). If
\[
\alpha(i) = b
\]
then \( x(i) \) is a feasible solution to the ILPC.

Each \( x(i) \) is generated from an earlier vector \( x(p) \) \((p < i)\), called the predecessor of \( x(i) \), by incrementing one of the components of \( x(p) \) by one.

If \( x_p^r \) is the component of \( x(p) \) that is incremented to give \( x(i) \) and \( e_r \) denotes the vector with a one in the \( r \)th component and zeros elsewhere then:
\[
x(i) = x(p) + e_r
\]
\[
C(i) = C(p) + c_r
\]
and \( \alpha(i) \equiv \alpha(p) + A^r \pmod{Q} \).
The strategy of the algorithm is to construct the sequence \( \{x(i)\} \) to satisfy the following conditions:

(i) If \( p \neq q \), then \( x(p) \neq x(q) \).
(ii) If \( p < q \), then \( C(p) \leq C(q) \).
(iii) \( x(i) \) is an optimal solution to the problem when \( a(i) = b \).
(iv) The sequence is finite, and \( a(i) = b \) for some \( x(i) \), if and only if the problem has a feasible solution.

The technique associates with each variable \( x_j \) a transition index \( t_j \) which indicates what the next predecessor will be if \( x_j \) is the next variable to be incremented. That is, if \( x(1), \ldots, x(k-1) \) denotes the current sequence, then:

\[
x(k) = x(t_r) + e_r \quad \text{for some } r \text{ between } 1 \text{ and } n.
\]

To determine the particular index \( r \) define a next cost:

\[
N_j = C(t_j) + c_j
\]

for each \( j = 1, \ldots, n \). In order for condition (ii) to hold the index \( r \) is selected by:

\[
N_r = \min \{N_1, N_2, \ldots, N_n\}.
\]

Initially \( x(1) \) is the 0-vector and \( t_j = 1 \) for all \( j \), so that \( x(2) \) will be one of the vectors \( e_1, e_2, \ldots, e_n \).

When \( x_j \) is incremented, \( t_j \) becomes \( \bar{t}_j \) according to the rule:

\[
\bar{t}_j = \min\{i: i > t_j \text{ and } r_i \geq j\}
\]

where \( r_i \) names the variable incremented to get \( x(i) \). This
rule insures that condition (i) will hold and that, if the algorithm is not permitted to stop when \( a(i) = b \) is generated, the method will generate every \( x \)-vector having non-negative integer components.

For proofs of the above conditions and illustrative examples see Glover [1969].

The strategy in generating the sequence of vectors \( x(i) \) corresponds quite closely to the strategy of the dual simplex method for solving the LP problem (see Spivey and Thrall [1970]) in that if \( x(i) \) is generated, it is optimal provided it is feasible.
BIBLIOGRAPHY

Barnett, I. A.


Birkhoff, G., and MacLane, S.


Bradley, G. H.


Bradley, G. H., and Wahi, P. N.

Dickson, L. E.


Dudley, U.


Garfinkel, R. S., and Nemhauser, G. L.


Glover, F.


Glover, F., and Litzler, L.

1969 "Extension of an Asymptotic Integer Programming Algorithm to the General Integer Programming Problem," Graduate School of Business, University of Texas (Austin, Texas).

Gomory, R. E.


Hu, T. C.

1969 Integer Programming and Network Flows, Addison-Wesley (Reading, Massachusetts).
Mathews, G. B.


McCoy, N. H.


Schreier, O., and Sperner, E.


Shapiro, J. F.


Shapiro, J. F., and Wagner, H. M.


Spivey, A., and Thrall, R. M.

Stewart, B. M.


Thrall, R. M., and Tornheim, L.