GLOBAL JOURNAL OF MATHEMATICAL SCIENCES VOL. 5, NO. 1, 2006: 63 - 71
COPYRIGHT (C) BACHUDO SCIENCE CO. LTD. PRINTED IN NIGERIA. ISSN 1596 - 6208

ADAPTATION-II OF THE SURROGATE METHODS FOR LINEAR PROGRAMMING PROBLEMS

S. O. OKO

(Received 2 August, 2005; Revision Accepted 19 September 2005)

ABSTRACT

A linear programming problem seeks for a non-negative column vector, x, that maximizes a linear objective function, u^Tx, subject to $Ax \le b$, where A is a given matrix, and b and u are given column vectors. Using the same data, the dual problem to the primal seeks for a non-negative column vector, y, to minimize a linear objective function, b^Ty, subject to $A^Ty \ge u$. The surrogate methods exploit the Duality Theory to combine the two problems into one system of linear inequalities that treats the sign-restricted variables and the objective functions as constraints. Because the set of constraints in linear programming problems is sometimes a mixture of inequality and equality constraints, this paper modifies the surrogate methods and comes up with hybrids of the ones designed for a system of linear inequalities and those for a system of linear equations. The paper also proves that a feasible solution to the resulting linear inequality problem is made up of the primal and dual optimal solutions for the given primal problem and its associated dual. It goes further to prove the dual theorem as it relates to the surrogate methods.

KEYWORDS: Linear Programming, Duality Theory, Surrogate Methods.

1. INTRODUCTION

A linear programming, LP, problem is an optimization problem with a linear function, linear constraints and sign-restricted variables searching for an $x \in R^n$ to

$$\begin{aligned} &\text{maximize} && z = u^Tx \\ &\text{subject to} && Ax \le b \qquad\qquad (1) \\ &\text{and} && x \ge 0 \end{aligned}$$

given $A \in R^{m \times n}$, $b \in R^n$ and $u \in R^n$, where u, x and b are column vectors (Hillier and Lieberman, 1974; Wagner, 1975; Strang, 1976; Bradley, Hax and Magnanti, 1977).

For every LP problem (1), there is another LP problem related to it and which reverses the objective function and the direction of the functional constraints by asking for a column vector, $y \in R^m$ to

$$\begin{aligned} &\text{minimize} && z' = b^Ty \\ &\text{subject to} && A^Ty \ge u \qquad\qquad (2) \\ &\text{and} && y \ge 0. \end{aligned}$$

Problem (1) is called the primal problem while the related problem (2) is known as the dual (Hillier and Lieberman, 1974; Wagner, 1975; Strang, 1976; Bradley, Hax and Magnanti, 1977). But note that we have in no way said that the primal is always a maximization problem while the dual must be a minimization one. Because the dual of a dual is the primal, whichever is the given problem to be solved is taken as the primal and the related problem becomes the dual.

2. Preliminaries

Exploiting the relevant aspects of the duality theory (Hillier and Lieberman, 1974; Wagner, 1975; Strang, 1976; Bradley, Hax and Magnanti, 1977), we can reformulate the primal-dual pair of an LP problem into one system of linear inequalities, LI, so that like the simplex method, the surrogate methods can find x and y simultaneously. But unlike the simplex method, the objective function value is computed only after a solution to the combined system is found. The vital relationships utilized in the reformulation of the LP problem are summarized below as lemmas from Hillier and Lieberman, 1974; Wagner, 1975; Strang, 1976; Bradley, Hax and Magnanti, 1977.

Lemma 2.1 (Weak Duality Theorem)

If (i) x is primal feasible; and (ii) y is dual feasible; then (iii) $u^Tx \le b^Ty$

Lemma 2.2 (Sufficient Optimality Criterion)

If (i) x' is primal feasible; (ii) y' is dual feasible; and (iii) $u^Tx' = b^Ty'$; then (iv) x' is primal optimal and y' is dual optimal.

S. O. Oko, Department of Mathematics, Ebony State University, Abakaliki, Nigeria

Lemma 2.3 (Unboundedness and Infeasibility Property)

i) If $\exists x$ primal feasible and $\nexists y$ dual feasible, then $u^T x = +\infty$;

ii) If $\nexists x$ primal feasible and $\exists y$ dual feasible, then $u^T x = -\infty$;

The converse of Lemma 2.2 is

Lemma 2.4 (Strong Duality Theorem)

If (i) x^* is primal optimal; and (ii) y^* is dual optimal; then (iii) $u^T x^* = b^T y^*$.

Lemma 2.4 is what all the texts on LP problem refer to as *the dual theorem* because it is the fundamental theorem *of the duality theory*. However the proof is centered on the simplex method. Since the surrogate methods are primarily designed to get a feasible solution, Lemma 2.2 is very crucial in adapting those methods for LP problems. The proofs of these lemmas can be found in any of the references given. However the proof of Lemma 2.4 will be presented later as part of Theorem 3.1 and as it relates to the surrogate methods.

3. The Transformation

Lemma 2.2 is a sufficient condition for optimality. Therefore in our search for the x and y that satisfy the functional and sign constraints in (1) and (2), we must make sure that they also satisfy the equality $u^T x = b^T y$ so that they are not only feasible but also optimal. However,

$$u^T x = b^T y \iff u^T x \le b^T y \text{ and } u^T x \ge b^T y.$$

By Lemma 2.1, once x and y are primal and dual feasible, respectively, they automatically satisfy the relationship $u^T x \le b^T y$. Therefore to guarantee that equality is satisfied, all we need to do is to include the other half of the pair of inequalities as

$$-u^T x + b^T y \le 0. \qquad (3)$$

With (1), (2) and (3) therefore, we can transform an LP problem into an LI problem that seeks for an $x \in R^n$ and a $y \in R^m$ such that

$$\begin{pmatrix} A & O \\ O & \bar{A} \\ -I_n & O \\ O & -I_m \\ -u' & b' \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \le \begin{pmatrix} b \\ \bar{u} \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad (4)$$

where $\bar{a}^i = -a_i / \|a_i\|$ for $i = 1, 2, \ldots, n$, rows of $\bar{A}$;

$\bar{u}_i = -u_i / \|a_i\|$ for $i = 1, 2, \ldots, n$, rhs of the dual;

$(-u', b') = (-u^T, b^T) / \|(-u^T, b^T)\|$, the normalized optimality row;

and $\|v\|$ is the norm of a vector, v.

Recall (Oko, 1992) that the rows of A are assumed normalized or must be normalized before applying the surrogate methods. But this does not mean that its columns, which are the rows in the dual problem, are normalized too. In other words (Oko, 1992)

$$\|a^i\| = 1 \not\Longrightarrow \|a_i\| = 1.$$

Therefore for correct application of the surrogate methods, the columns of A, (i.e. the rows of A^T) and $(-u^T, b^T)$ must be normalized as defined in (4) for $\bar{a}^i$, $\bar{u}_i$ and (u', b').

Let us denote the coefficient matrix in (4) by A', the variables by w and the right-hand side by f. Then in our compact notation, (4) can be written as

$$A'w \le f. \qquad (5)$$

Theorem 3.1

Let $w^{(k)T} = (x^{(k)T}, y^{(k)T})$ be the k^{th} iterate at which the surrogate algorithms terminate normally without aborting. Then we claim that

(i) $x^{(k)}$ is primal optimal; and (ii) $y^{(k)}$ is dual optimal.

Consequently,

(iii) $u^T x^{(k)} = b^T y^{(k)}$ = the required optimal objective function value.

Proofs of parts (i) and (ii)

The original proof of convergence (Oko, 1992) and those in two other papers (Oko, 2005a; Oko, 2005b) established a steady convergence to a feasible solution. Therefore if the iterations terminate in a normal way without aborting, then $(x^{(k)T}, y^{(k)T})$ is a feasible solution for (4). Now

$$
\begin{aligned}
w^{(k)} \text{ feasible } &\Longleftrightarrow A'w^{(k)} \le f \text{ and } w^{(k)} \ge 0 \\
&\Longleftrightarrow Ax^{(k)} \le b,\ \bar{A}y^{(k)} \le \bar{u},\ -x^{(k)} \le 0,\ -y^{(k)} \le 0,\ -u'x^{(k)} + b'y^{(k)} \le 0 \\
&\Longleftrightarrow x \text{ is primal feasible, } y \text{ dual feasible and } u'x^{(k)} - b'y^{(k)} \ge 0.
\end{aligned}
$$

But by Lemma 2.1, $u'x^{(k)} - b'y^{(k)} \le 0$.

Therefore $u'x^{(k)} - b'y^{(k)} \le 0$ and $u'x^{(k)} - b'y^{(k)} \ge 0 \Longleftrightarrow u'x^{(k)} = b'y^{(k)}$

With $x^{(k)}$ primal feasible, $y^{(k)}$ dual feasible, and $u'x^{(k)} = b'y^{(k)}$, then by Lemma 2.2

(i) $x^{(k)}$ is primal optimal; and (ii) $y^{(k)}$ dual optimal.

Proof of part (iii)

It should be noted that what is computed during the search for a feasible solution for (4) is not $u^T x^{(k)}$ and/or $b^T y^{(k)}$ per se, but rather it is the arithmetic expression, $u'x^{(k)} - b'y^{(k)}$. Therefore the value for $u^T x^{(k)}$ or $b^T y^{(k)}$ has to be computed only after a normal termination has occurred for the algorithms. Let us assume that $u^T x^{(k)} \ne b^T y^{(k)}$ at the time the iterations have terminated in a normal way without being aborted abnormally for inconsistency.
Then by the definitions in (4),

$$
\begin{aligned}
u^T x^{(k)} \ne b^T y^{(k)} &\Longleftrightarrow u^T x^{(k)} / \| -u^T, b^T \| \ne b^T y^{(k)} / \| -u^T, b^T \| \\
&\Longleftrightarrow u'x^{(k)} \ne b'y^{(k)} \text{ and } (x^{(k)T}, y^{(k)T}) \text{ is not a solution for (4).}
\end{aligned}
$$

But this contradicts not only the already proven first part of the theorem, but also the established proofs of convergence to a solution when the iterations terminate without aborting for inconsistency! Therefore our assumption is false and so Theorem 3.1 holds.

4. Implementation

A' is a (2m+2n+1) by (n+m) sparse matrix. It is made up of 10 blocks, 6 of which are zero and identity matrices. Likewise f is an (2m+2n+1)-vector with blocks of zero elements. The zero and identity matrices need not be stored. Since neither A' nor f is recomputed during the search for a solution, storing them as they are will be most inefficient in space requirements and in computation time. Table 1 summarizes the actual space requirements.

Table 1: Storage Requirements

Constraints	Augmented Matrix	Required No. Locations
Primal	$A\|b$	2m(n+1)
Dual	$\bar{A}\|\bar{u}$	2n(m+1)
Variables	$-I_{m+n}\|O$	0
Optimality	$(-u', b')\|0$	2(n+m)
Total	$A'\|f$	4(mn+m+n)

In Table 1, we have multiplied each required number of locations by 2 because floating-point numbers are better computed in double precision arithmetic to improve accuracy.

Equality Constraints and Unrestricted Variables

By the theory of LP problems, the primal constraints are paired up with the dual variables, and the primal variables with the dual constraints (Hillier and Lieberman, 1974; Wagner, 1975; Bradley, Hax and Magnanti, 1977). If a primal constraint is an inequality constraint, its associated dual variable is restricted in sign. But if it is an equality constraint, the associated dual variable has no sign restriction. Similarly, a sign-restricted primal variable gives rise to an associated dual inequality constraint, while an unrestricted primal variable results in a dual equality constraint. These correspondences are summarized in Table 2 below.

For convenience, it is advisable to arrange the constraints so that the primal inequality constraints are grouped together, preferably as the first set of m_1 constraints, say, such that $0 \leq m_1 \leq m$. Similarly, the first n_1 primal variables should be the sign-restricted ones such that $0 \leq n_1 \leq n$. With a 5-type classification, we shall have

1. Primal constraints as type 1;
2. Dual constraints as type 2;
3. Primal sign restricted variables as type 3;
4. Dual sign restricted variables as type 4; and
5. The optimality constraint as type 5.

This will facilitate handling of the problem without storing A′ and f as they are.

Table 2: Primal-Dual Correspondences

One Problem	The Other Problem
Maximization of objective function	Minimization of objective function
Coefficients of objective function	Right-hand sides of constraints
i^{th} constraint, $a^i x \leq b_i$	i^{th} variable, $y_i \geq 0$
i^{th} constraint, $a^i x = b_i$	i^{th} variable, y_i, is unrestricted
j^{th} variable, $x_j \geq 0$	j^{th} constraint, $(a_j)^T y \geq u_j$
j^{th} variable, x_j unrestricted	j^{th} constraint, $(a_j)^T y = u_j$
Inconsistency	Unbounded function value
Unbounded function value	Inconsistency

5. The Algorithms

The surrogate algorithms for solving LP problems are hybrids of those for solving LI problems (Oko, 1992) and those for LE problems (Oko, 2005c). The essential definitions and formulae we used for our searches were

$I = \{i \mid 1 \leq i \leq m\}$

$d = Ax - b,$ the distances of x from the m hyperplanes, i.e. the error in x

$c_p = A(a^p)^T,$ i.e. $c_{pi} = a^i \cdot a^p$, the cosine of the angle between H_p and H_i

$g_i = (d_i - rc_{pi}) / \sqrt{(1-(c_{pi})^2)} \ \forall i \in I$ and $1-(c_{pi})^2 \neq 0$

$r =$ the distance of x from the most violated half-space, H_p.

To accommodate the enlarged but sparse system, the following formulae listed below will be in use. Note that just as the distance of the point x from the i^{th} hyperplane of a functional constraint is defined as

$$a^i x - b_i \qquad \forall i,$$

its distance from the j^{th} hyperplane with respect to variable-constraint j is

$$-e^j x - 0 = -x_j \qquad \forall j \quad \text{where } e^j \text{ is the } j^{th} \text{ row of the identity matrix } I_n.$$

These distances and the cosines for the formula for g_i are summarized in Table 3.

Table 3: Formulae for Distances and Cosines

Constraint Matrix	Distance	Cosine/Dot-product				
		a^p	$\bar{a}^p$	$-e^p$	$-e^p$	$(-u', b')$
A	$Ax - b$	$A(a^p)^T$	0	$-a_p$	0	$-Au'^T$
$\bar{A}$	$\bar{A}y - \bar{u}$	0	$\bar{A}(\bar{a}^p)^T$	0	$-\bar{a}_p$	$\bar{A}b'^T$
$-I_n$	$-x$	$-(a^p)^T$	0	e_p	0	u'^T
$-I_m$	$-y$	0	$-(\bar{a}^p)^T$	0	e_p	$-b'^T$
$(-u', b')$	$-u'x + b'y$	$-u'(a^p)^T$	$b'(\bar{a}^p)^T$	u'_p	$-b'_p$	1

With the above formulae, it is obvious that we will need no computations for a lot of the data/information required since such can be retrieved from other sources.

We shall modularize the algorithms into sub-algorithms for specific tasks.

5.1 Surrogate-I for LP Problems

The most violated half-space and the most violated manifold of two half-spaces are chosen. If there is no violated half-space, x is a solution. If there is a violated half-space but no violated manifold, then the orthogonal projection of x onto the most violated half-space is a solution. Otherwise that orthogonal projection replaces x and the process is repeated with the new x.

Step 1. Call Initialize;

Step 2. Call Choose1 (p, r_k, T_p); $g = 0$;

Step 3. If $r_k \le \delta$ then output $u^Tx^{(k)}$, $x^{(k)}$, $y^{(k)}$ and stop; else go to Step 4;

Step 4. For Type = 1 to 5;

Step 4.1 Call Dotprd (c, Type, T_p, p);

Step 4.2 Call Choose2 $(r_k, c, \text{Type}, s, g, T_s, v)$;

Step 5. Call Update (T_p, p, r_k, g, k);

Step 6. If $g \le \delta$ then output $u^Tx^{(k+1)}$, $x^{(k+1)}$, $y^{(k+1)}$ and stop;

else go to Step 2 with $k = k + 1$.

5.2 Surrogate-II for LP Problems

The most violated half-space and the most violated manifold of two half-spaces are chosen. If there is no violated half-space, x is a solution. If there is a violated half-space but no violated manifold, then the orthogonal projection of x onto the most violated half-space is a solution. Otherwise the orthogonal projection of x onto the most violated manifold (not the most violated half-space as in 5.1) replaces x and the process is repeated with the new x.

Steps 1 - 4 are the same as those in section 5.1 above.

Step 5. If $g \le \delta$ then go to Step 6; else go to Step 8;

Step 6. Call Update (T_p, p, r_k, g, k);

Step 7. Output $u^Tx^{(k+1)}$, $x^{(k+1)}$, $y^{(k+1)}$ and stop;

Step 8. $\beta_s = g/\sqrt{(1-v^2)}$; $\beta_p = r_k - \beta_s v$;

Step 9. Call Update (T_p, p, β_p, g, k); Call Update $(T_s, s, \beta_s, g, k+1)$;

Step 10. Go to Step 2 with $k = k + 2$.

5.3 Algorithm for Surrogate-III

This is similar to Surrogate-II but 3 (not 2) most violated half-spaces are selected and the infeasible x is projected orthogonally onto their manifold.

Steps 1 - 8 are the same as those in 5.2;

Step 9 $r_{k+1} = \sqrt{((r_k)^2 + g^2)}$; $g = 0$;

Step 10. For Type = 1 to 5;

Step 10.1 Call Dotprd (c, Type, T_p, p); Call Dotprd ($\hat{c}$, Type, T_s, s);

Step 10.2 $c = (\beta_p c + \beta_s \hat{c})/r_{k+1}$; Call Choose2 ($r_{k+1}$, c, Type, t, g, T_t, v);

Step 11. If $g \le \delta$ then go to Step 12; else go to Step14;

Step 12. Call Update (T_p, p, β_p, g, k); Call Update (T_s, s, β_s, g, k+1);

Step 13. Output $u^T x^{(k+2)}$, $x^{(k+2)}$, $y^{(k+2)}$ and stop;

Step 14. $\lambda_t = g/\sqrt{(1-v^2)}$; $\lambda_s = \beta_s(1 - \lambda_t v / r_{k+1})$; $\lambda_p = \beta_p(1 - \lambda_t v / r_{k+1})$;

Step 15. Call Update (T_p, p, λ_p, g, k); Call Update (T_s, s, λ_s, g, k+1);

Step 16. Call Update (T_t, t, λ_t, g, k+2);

Step 17. Go to Step 2 with k = k + 3.

5.4 Algorithm for Surrogate-R

R ($2 \le R \le m$) violated and distinct constraints are chosen for computation of a linear combination to serve as a surrogate constraint. The outward normal, $\hat{a}_k$, of the surrogate is then used to update x and the search for a solution continues.

Step 1. Call Initialize; $k_0 = 0$;

Step 2. Call Choose1 (p, r_k, T_p);

Step 3. If $r_k \le \delta$ then output $u^T x^{(k0)}$, $x^{(k0)}$, $y^{(k0)}$ and stop; else go to Step 4;

Step 4. $P = \{p, T_p\}$; $q = 1$; $\hat{a}_k = 0$; $g = 0$;

Step 5. If $T_p = 1$ then for j = 1 to n; $\hat{a}_{kj} = a_{pj}$;

elseif $T_p = 2$ then for j = 1 to m; $\hat{a}_{k,n+j} = \bar{a}_{pj}$;

elseif $T_p = 3$ then $\hat{a}_{kp} = -1$; elseif $T_p = 4$ then $\hat{a}_{k,n+p} = -1$; else $\hat{a}_k = (-u', b^T)$;

Step 6. For Type = 1 to 5;

Step 6.1 Call Dotprd (c, Type, 6, k);

Step 6.2 Call Choose2 (r_k, c, Type, s, g, T_s, v);

Step 7. If $g \le \delta$ or $\{s, T_s\} \subseteq P$ then go to Step 12; else go to Step 8;

Step 8. $P = P \cup \{s, T_s\}$; $q = q+1$; $\beta_s = g/\sqrt{(1-v^2)}$; $\beta_k = r_k - \beta_s v$;

Step 9. $r_{k+1} = \sqrt{((r_k)^2 + (g)^2)}$;

Step 10. If $T_s = 1$ then For j = 1 to n; $\hat{a}_{k+1,j} = (\beta_k \hat{a}_{kj} + \beta_s a_{sj}) / r_{k+1}$;

elseif $T_s = 2$ then For j = 1 to m; $\hat{a}_{k+1,n+j} = (\beta_k \hat{a}_{k,n+j} + \beta_s \bar{a}_{sj}) / r_{k+1}$;

elseif $T_s = 3$ then For j = 1 to n; $\hat{a}_{k+1,j} = (\beta_k \hat{a}_{k,j} - \beta_s e_{sj}) / r_{k+1}$;

elseif $T_s = 4$ then For j = 1 to m; $\hat{a}_{k+1,n+j} = (\beta_k \hat{a}_{k,n+j} - \beta_s e_{sj}) / r_{k+1}$;

else $\hat{a}_{k+1} = (\beta_k \hat{a}_k + \beta_s(-u', b^T)) / r_{k+1}$;

Step 11. g = 0; Go to Step 6 with k = k + 1;

Step 12. $(x^{(k0+q)T}, y^{(k0+q)T}) = (x^{(k0)T}, y^{(k0)T}) - r_k \hat{a}_k$;

Step 13. If $g \le \delta$ then output $u^T x^{(k0+q)}$, $x^{(k0+q)}$, $y^{(k0+q)}$ and stop; else go to Step 14;

Step 14. For i = 1 to m; $d_i = d_i - r_k \sum_{j=1}^{n} a_{ij} \hat{a}_{kj}$;

Step 15. For i = 1 to n; $đ_i = đ_i - r_k\sum_{j=1}^{m}\bar{a}_{ij}\hat{a}_{k,n+j}$;

Step 16. $d_z = d_z - r_k(-u', b^T)\cdot\hat{a}_k$;

Step 17. Go to Step 2 with $k = k+1$ and $k_0 = k_0 + q$.

5.5 Subalgorithm Initialize;

This subalgorithm reads in the given parameters and uses them to initialize the remaining working data

Step 1. Input m, n, m_1, n_1, A, b, u, δ;

Step 2. $\bar{A} = -A^T$; d = b; $đ = -u$; $u' = u$;

Step 3. Normalize the rows of A|d, $\bar{A}$|đ and $(-u', b^T)$; /* b is now used for b' */

Step 4. $x^{(0)} = 0$; $y^{(0)} = 0$; k = 0; /* Zero for x & y is a convenient arb. choice. */

Step 5. d = -d; đ = -đ; $d_z = 0$; /* Dist. of x & y from pri, dua. & opt. ½-spaces */

Step 6. Return.

5.6 Subalgorithm Choose1 (p, r, T);

Choose p such that H_p is the most violated of all the half-spaces.

Step 1. $r = d_z$; T = 5; /* Choose the optimality constraint */

Step 2. For i = 1 to m_1; /* Choose from primal constraints or dual variables */

Step 2.1 If $r < d_i$ then $r = d_i$; p = i; T = 1;

Step 2.2 If $r < -y_i$ then $r = -y_i$; p = i; T = 4;

Step 3. For i = m_1+1 to m;

If $r < |d_i|$ then $r < |d_i|$; p = i; T = 1;

Step 4. For i = 1 to n_1; /* Choose from dual constraints or primal variables */

Step 4.1 If $r < đ_i$ then $r = đ_i$; p = i; T = 2;

Step 4.2 If $r < -x_i$ then $r = -x_i$; p = i; T = 3;

Step 5. For i = n_1+1 to n;

If $r < |đ_i|$ then $r = |đ_i|$; p = i; T = 2;

Step 6. Return.

5.7 Subalgorithm Dotprd (c, T_2, T_1, k);

The dot products of all the outward normals in T_2 with that of the most violated half-space k in T_1 are computed using a computed-go-to statement to select the required section. Type 6 is the constructed surrogate constraint for Surrogate-R.

Step 1. c = 0;

Step 2. Go to (3, 5, 7, 9, 11), T_2;

Step 3. If $T_1 = 1$ then $c = A(a^k)^T$; elseif $T_1 = 3$ then $c = -a_k$;

elseif $T_1 = 5$ then $c = -Au'^T$; elseif $T_1 = 6$ then For i = 1 to m; $c_i = \sum_{j=1}^{n} a_{ij}\hat{a}_{kj}$;

Step 4. Return;

Step 5. If $T_1 = 2$ then $c = \bar{A}(\bar{a}^k)^T$; elseif $T_1 = 4$ then $c = -\bar{a}_k$;

elseif $T_1 = 5$ then $c = \bar{A}b$; elseif $T_1 = 6$ then For $i = 1$ to n; $c_i = \sum_{j=1}^{m} \bar{a}_{ij}\hat{a}_{k,n+j}$;

Step 6. Return;

Step 7. If $T_1 = 1$ then $c = -(a^k)^T$; elseif $T_1 = 3$ then $c = e_k$;

elseif $T_1 = 5$ then $c = u'^T$; elseif $T_1 = 6$ then For $i = 1$ to n; $c_i = -\hat{a}_{ki}$;

Step 8. Return;

Step 9. If $T_1 = 2$ then $c = -(\bar{a}^k)^T$; elseif $T_1 = 4$ then $c = e_k$;

elseif $T_1 = 5$ then $c = -b$; elseif $T_1 = 6$ then For $i = 1$ to m; $c_i = -\hat{a}_{k,n+i}$;

Step 10. Return;

Step 11. If $T_1 = 1$ then $c_1 = -u'(a^k)^T$; elseif $T_1 = 2$ then $c_1 = b \cdot \bar{a}^k$;

elseif $T_1 = 3$ then $c_1 = u'_k$; elseif $T_1 = 4$ then $c_1 = -b_k$;

elseif $T_1 = 5$ then $c_1 = 1$; else $c_1 = (-u', b^T) \cdot \hat{a}_k$;

Step 12. Return.

5.8 Subalgorithm Choose2 (r, c, Type, s, g, T_s, v);

The algorithm chooses H_s (or H_t) such that $\partial H_p \cap \partial H_s$ (or $\partial H_p \cap \partial H_s \partial H_t$) is the most violated manifold of 2 (or 3) half-spaces.

Step 1. If Type = 1 then $q = m$; $q_1 = m_1$; $f = d$;

elseif Type = 2 then $q = n$; $q_1 = n_1$; $f = \bar{d}$;

elseif Type = 3 then $q = n_1$; $q_1 = n_1$; $f = -x$;

elseif Type = 4 then $q = m_1$; $q_1 = m_1$; $f = -y$; else $q = 1$; $q_1 = 1$; $f_1 = d_z$;

Step 2. If $c = 0$ then go to Step 4; else go to Step 3;

Step 3. For $i = 1$ to q;

Step 3.1 $f_i = f_i - rc_i$;

Step 3.2 If $1-(c_i)^2 \neq 0$ then $f_i = f_i / \sqrt{(1-(c_i)^2)}$;

Step 3.3 If $1-(c_i)^2 = 0$ & (($i \leq q_1$ & $f_i > 0$) or ($i > q_1$ & $f_i \neq 0$)) then abort; else continue;

Step 4. For $i = 1$ to q;

If $i \leq q_1$ & $g < f_i$ then $g = f_i$; $s = i$; T = Type; $v = c_s$;

elseif $i > q_1$ & $g < |f_i|$ then $g = |f_i|$; $s = i$; T = Type; $v = c_s$;

Step 5. Return.

5.9 Subalgorithm Update (T, p, r, g, k);

This algorithm updates x and y with the outward normal of the chosen half-space. If a solution is not yet found, d is also updated.

Step 1. If $T = 1$ then $x^{(k+1)} = x^{(k)} - r(a^p)^T$; elseif $T = 2$ then $y^{(k+1)} = y^{(k)} - r(\bar{a}^p)^T$;

elseif $T = 3$ then $x^{(k+1)} = x^{(k)} + r(e^p)^T$; elseif $T = 4$ then $y^{(k+1)} = y^{(k)} + r(e^p)^T$;

else $x^{(k+1)} = x^{(k)} + ru'^T$; $y^{(k+1)} = y^{(k)} - rb$;

Step 2. If $g \leq \delta$ then return; else go to Step 3;

Step 3. If $T = 1$ then $d = d - rA(a^p)^T$; elseif $T = 2$ then $\bar{d} = \bar{d} - r\bar{A}(\bar{a}^p)^T$;

elseif $T = 5$ then $d_z = d_z - r$;

Step 4. Return.

Conclusion

This paper modifies the surrogate methods and comes up with hybrids of the ones designed for a system of linear inequalities and those for linear equations. It is also shown that a feasible solution to the resulting linear inequality problem is made up of the primal and dual optimal solutions for the given primal problem and its associated dual. The dual theorem as it relates to the surrogate methods is proved. The surrogate algorithms for LP problems are given.

REFERENCES

Bradley, S. P., Hax, A. C. and Magnanti, T. L., 1977. Applied Mathematical Programming, Addison-Wesley Company, Reading, Massachusetts.

Hillier F. S. and Lieberman, G. J., 1974. Operations Research, 2nd ed., Holden-Day, Inc., San Francisco.

Oko, S. O., 1992 Surrogate Methods for Linear Inequalities, Journal of Optimization Theory and Applications, Vol. 72(2): 247-268.

Oko, S. O., 2005a. On the Convergence of the Surrogate Methods, Global Journal of Mathematical Sciences, 4(1&2): 81-87

Oko, S. O., 2005b. Rates of Convergence for the Surrogate Methods. Submitted to International Journal of Applied and Natural Sciences.

Oko, S. O., 2005c. Adaptation-I of the Surrogate Methods for Linear System of Equations. Global Journal of Mathematical Sciences, 4(1 & 2): 65-70

Strang, G., 1976. Linear Algebra and its Applications, Academic Press, Inc., New York.

Wagner, H. M., 1975. Principles of Operations Research, 2nd ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey.