ON THE TRUNCATED-NEWTON APPROACH
FOR THE HYDROPOWER GENERATION MANAGEMENT
L. F. ESCUDERO
I.B.M. SCIENTIFIC CENTER

In the paper we present the motivation for using the Truncated Newton methodology when obtaining the superbasic stepdirection in an algorithm that maximises the hydropower generation in a multireservoir, multiperiod power system. The decision variables are the water to be released from and stored in each reservoir in each time period over a given time horizon. The function that relates the hydropower generation with the decision variables has a special structure -- that allows to use second-order information without requiring too-much computer storage and - time-consuming.

Keywords: NON-LINEAR PROGRAMMING. NUMERICAL ALGORITHMS. ELECTRIC POWER GENERATION.

1. INTRODUCTION.

PROBLEM DESCRIPTION.

This paper reports the motivation for using second-order information about the objective function of a nonlinear network flow problem so that the computer storage and time-consuming are affordable even for very-large cases. The problem consists in the maximization of the hydropower generated along a time horizon (usually, one year) by a multireservoir power system. The decision variables are the amount of water to be released from and stored in each of the interconnected reservoirs in each time period (usually, a week). The constraints are of two kinds: linear equations to ensure flow balance and bounds on the variables.

The purpose of the bounds on the variables is threefold. First, to ensure that the water released serves the flood control, irrigation and navigational purposes. Second, to ensure that the amount of water released from a given reservoir to any of its directly downstream reservoirs must not exceed its canal capacity, so that the overflow is not used for producing electricity. Third, to penalise the amount of water stored that exceeds a safety capacity in a given reservoir.

The constraints form a linear system that can be represented by a special direct graph whose nodes and arcs correspond to the state of the reservoirs in each time period and the decision variables, respectively.

Assuming a multireservoir system of 40 reservoirs and a planning horizon of 52 time periods, the graph has over 2000 nodes and 4000 arcs. The dimensions of the problem are affordable for linear objective functions given -- the current state-of-the-art of special data structures for storing and updating the network (see /5/, /13/, /14/, /15/ among others); a sound nonlinear network algorithm must require a computing time with the same order of magnitude of the linear ones /7/.

Recently, new algorithms have been designed for specializing linear primal data structures to nonlinear network flow problems; see /1/, /8/, /17/, /18/, /20/ and related references. Apart from recent developments (see /2/, /3/, /7/, /8/ among others), the methods use a linearized subproblem to generate step-directions so that the reduced gradient methodology is used in the basic-nonbasic environ-

- L.F. Escudero - I.B.M. Centro Científico - Pd de la Castellana, 4 - Madrid 1
- Article rebut el Maig del 1983.
moment /19/ or in the more efficient basic-superbasic-nonbasic environment /16/. The convergence is very poor since no second-order information about the objective function is used. The current methods that use the Hessian matrix are designed for solving medium scale problems.

This work proposes to use the Truncated Newton method to solve the hydropower generation problem. It was introduced in /9/ for general unconstrained nonlinear problems, extended in /10/ to linearly constrained problems, and specialized in /8/ to network flow problems. The new algorithm (whose details and computational experience with real-life cases are reported elsewhere /11/) uses second-order information in a very large scale network flow problem, given the special structure of both the objective function and constraints.

The paper is organized as follows. Section 2 presents the network flow formulation of the reservoirs management problem. Section 3 describes the hydropower generation function. Section 4 summarizes the variable-reduction environment of the algorithm. Section 5 presents the approach for obtaining the superbasic stepdirection in the hydropower generation problem and introduces the new concept of independent superbasic sets. Section 6 describes the de-activating process and introduces the new concept of independent sets of candidate nonbasic arcs to be de-activated. And finally, Section 7 outlines some topics for future research and experimentation.

2. PROBLEM FORMULATION.

Let J denote the set of reservoirs, T the time periods set, WcJ the water storage reservoirs (and, then, J/W is the set of run-off river plants), EoJ the set of reservoirs that are not used for hydropower generation, Pj (Qj) the set of reservoirs directly upstream (downstream) from reservoir j. The decision variables are denoted r^t_{ji}, the amount of water released from reservoir j to reservoir i in period t and s^t_{ij} the amount of water stored in reservoir k at the beginning of period t. The flow balance equations are

\[ \sum_{i \in P_j} r^t_{ji} = s^t_{ij} + \sum_{i \in O_j} r^t_{ji} + s^t_{ij} + 1, j = b_j \]

\[ v_t, j \in J \quad (2.1) \]

where b^t_j is the next exogenous inflow to reservoir j in time period t, and s^t_{ij} and s^t_{ij} are fixed.

Letting X be a vector of all decision variables and b a vector of the exogenous inflows, system (2.1) can be written

\[ AX = b \quad (2.2) \]

where A is the node-arc incidence matrix. For \( |Q_j|=1 \) this network is referred to as a temporally expanded arborescence /18/. Since in our case, it is not excluded, although it is not frequent, that \( |Q_j| \geq 1 \) let us term it temporally expanded quasi-arborescence. Each column of A corresponds to an arc (whose flow is r^t_{ji} or s^t_{ij}) and each row to a node for each pair (period, reservoir). The nonzero elements in a column are +1 in the row corresponding to the node where the arc originates and -1 in the row where the arc terminates.

In addition, there is a root node which represents the Exit in all time periods for the j reservoirs j \in J such that 3 \in Q_j, i = |NJ|+1.

The bounds on water released and stored are given by

\[ l^t_{ji}, s^t_{ij}, s^t_{ij} \quad \forall i \in Q_j, j \in J \quad (2.3a) \]

\[ m^t_{ij}, s^t_{ij}, s^t_{ij} \quad v_t, j \in W \quad (2.3b) \]

such that usually, m^t_{ij} = m_j and M^t_{ij} = M_j \quad v_t, W, where m_j and M_j are constant values, being --m^t_{ij}=M^t_{ij} \quad v_t,W.

Following a traditional approach /16/, matrix A can be partitioned as follows.

A = (B S N)

where the columns of B form a basis and correspond to the basic arcs, and the columns of S and N correspond to the superbasic and nonbasic arcs, respectively; let B, S and N denote the related basic, superbasic and nonbasic sets of arcs. Nonbasic arcs are temporarily fixed to their bounds, and the flow in sets B and S vary between their bounds.

By construction of A it can be shown...
(1) Any basis \( B \) may be ordered such that it is upper triangular /6/.

(2) The arcs corresponding to columns in any basis form a spanning tree of the network /5/.

(3) A maximal basis spanning tree for a given feasible solution avoids a basic-supercritical degenerate pivot /8/; otherwise, null steps are more frequent than in problems with a general structure.

(4) Let \( Z \) be the variable-reduction matrix whose columns form a basis for the null space of \( A \), given \( A Z = 0 \), such that /16/

\[
Z = \begin{pmatrix} \left(-B^{-1}\right) & I & 0 \end{pmatrix}
\]

where \( n \) denotes the number of nodes --- without including the root and denotes the number of arcs. Let Basic-Equivalent Path (BEP) define the unique path --- in the basis spanning tree that leads --- from originating node, say \( i_k \) in supercritical or nonbasic arc \( k \) to determining node, say \( j_k \); let \( \beta_k \) denote the set of arcs in the BEP of arc \( k \). Arc \( k' \in \beta_k \) --- has a forward orientation in the BEP of arc \( k \) if \( p(i_k, k') = j_k \), where \( p(.) \) is the predecessor of arc \( . \) in the BEP of \( k \); it has a reverse orientation if --- or second-order information in this very large-scale problem.

(5) The temporally expanded arborescence networks have the property that \( a-n \); note that, usually, \( a-n \gg n \) in a general network. As a result, \( |\beta_k| \) is very small. — Any time segment of our network is a quasi-arborescence and, then, \( |\beta_k| \) is still relatively small; this property will be exploited in the related algorithm.

3. THE HYDROPOWER GENERATION FUNCTION.

The important property of the hydropower generation function is spatial and temporal quasi-separability. See /2/, /7/, /8/ for separable functions; /8/ for temporal separable and spatially nonseparable functions in very large-scale problems. The hydropower generated in reservoir \( j \) at time period \( t \) can be expressed as follows.

\[
h_{tj} = K_{tj} I_{tji} V_{tj} \quad j \in J/E \quad (3.1)
\]

where \( K_{tj} \) may be constant or a nonlinear function \( f_{nt} \) of the \( n \)-variables for each reservoir in set \( J/E \). Let \( j \) be termed linear reservoir (and, then, \( (t,j,1) \) is a pure linear arc) if \( K_{tj} = K_{j} \), where \( K_{j} \) is a constant; otherwise, \( j \) is termed nonlinear reservoir such that --- function \( K_{tj} \) can be written

\[
K_{tj} = f_{nt}(c_{t_j} + c_{t+1,j})/2 \quad (3.2)
\]

where \( c_{t_j} \) is the water level in reservoir \( j \) at the beginning of time period \( t \), such that

\[
c_{t_j} = f_{nt}(s_{t_j}) \quad (3.3)
\]

Let PL denote the set of pure linear arcs --- \( (t,j,1) \), VL the set of linear arcs \( (t,j,1) \) --- with variable coefficients, and NL the set of nonlinear arcs \( (t,j) \). Arc \( (t,j,1) \in PL \) if --- \( K_{tj} = K_{j} \). Arc \( (t,j,1) \in VL \) if \( K_{tj} \) is a nonlinear function given by \((3.2)-(3.3))\;\;\;\text{note that} \; h_{tj} \; \text{is a linear function of } \sum_{j'} h_{tj'}.\quad \text{if } s_{t_j} \quad \text{and ---}

\[
s_{t+1,j} \text{ are fixed.} \quad \text{Arc } (t,j) \in NL \text{ if } K_{tj} \text{ is given by } (3.2)-(3.3), \text{ since } s_{t_j} \text{ and } s_{t+1,j} \text{ are nonlinear variables.}
\]

Without penalizing any amount of water stored or released, the objective can be written

\[
\max \sum_{t \in T} \sum_{j \in J/E} h_{tj} \quad (3.4)
\]

Let \( q_{tji} \) for \( (t,j,1) \in PL \) and \( q_{tji} \) for --- \( (t,j) \in NL \) denote the gradient elements related to the \( r \)-variables and \( s \)-variables, respectively. Note that the gradient related to set PL is constant, \( q_{tji} = 0 \) and \( q_{tji} = 0 \) \( \forall j \in E \), --- and \( q_{tji} \) for \( (t,j,1) \in VL \) does not change for fixed values of \( s_{t_j} \) and \( s_{t+1,j} \). The Hessian matrix \( G \) has the form
such that $G_3$ is a two-diagonal matrix for all $|Q_j| = 1$ for $j \in J$ and $G_4$ is a symmetric tri-diagonal matrix; see an example in form (3.6).

\[ G_3 = \begin{bmatrix} r_{312} & x & x \\ r_{412} & x & x \\ r_{512} & x & x \\ r_{612} & x & x \\ r_{712} & x & x \\ s_{31} & s_{41} & s_{51} & s_{61} & s_{71} \end{bmatrix} \]

\[ G_4 = \begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ s_{31} & s_{41} & s_{51} & s_{61} & s_{71} \end{bmatrix} \]

Note that the off-diagonal elements in a given column of matrix $G_3$ have the same value for $|Q_j| > 1, j \in W/E$.

Let $R_{ji}$ denote the power generation capacity of canal $(j,i)$ for $j \in J/E$, and $T_{tj}$ a safety upper bound on the amount of water to be stored in reservoir $j$ for $j \in W$ at the time period $t$; note that upper bound $u_{ji}$ may be regarded as the maximum physical capacity of the segment $(j,i)$ in the river. Now, since water -- overflow $r_{tji} - R_{ji}$ (where $r_{tji} = r_{tj} - u_{ji}$) cannot be used for hydropower generation, and it could be interesting to penalise the excess of water stored $s_{tj} - T_{tj}$, the objective can be expressed by (3.7) instead of using (3.1) and (3.4).

\[
\max \left\{ \sum_{t \in T} \sum_{j \in J/E} h_{tj} - \sum_{t \in T} \sum_{j \in W} P_{tj} \max(0, s_{tj} - T_{tj}) \right\} \quad (3.7a)
\]

\[
h_{tj} = k_{tj} \sum_{i \in Q_j} \min(r_{tji} - R_{ji}) \quad (3.7b)
\]

and $P_{tj}$ is the unit penalty for excess on water stored. The nondifferentiability introduced by (3.7) can be treated without great difficulty (see Section 6).


Let $d$ define the step direction from feasible solution, say $\bar{x}$ such that the new iterate -- can be expressed

\[ x = \bar{x} + \alpha d \quad (4.1) \]

where scalar $\alpha$ is the steplength. Given eqs. (2.2) and matrix partition (2.5), by linearity it results

\[
\begin{pmatrix} B & S & N \end{pmatrix} \begin{pmatrix} d_B \\ d_S \\ d_N \end{pmatrix} = 0 \quad (4.2)
\]

being $d = (d_B^T, d_S^T, d_N^T)^T$. The basic step direction $d_B$ is used to satisfy the constraints system (2.2), the elements of nonbasic stepdirection $d_N$, say $d_k$ for $k \in N$ are temporarily fixed to zero, and the superbasic stepdirection $d_S$ is used to maximise the objective function.

At each iteration, the problem then becomes determining vector $d$ and scalar $\alpha$, such that $\bar{x} + \alpha d$ is feasible and ascent enough, and the algorithm is globally and, if possible, $O$-superlinearly convergent. Direction $d$ is feasible if system (4.2) is satisfied. Since $d_N = 0$ and $d_S$ is allowed to be free, it results

\[ d_B = -B^{-1} S d_S \quad (4.3) \]

such that

\[ d = d_B + d_S \quad (4.4) \]

The ascent enough stepdirection $d_S$ can be obtained by 'solving' the problem

\[ \max(\tau d_S + 1/2 d_S^T H d_S) \quad (4.5) \]

where the reduced gradient $\tau$ and the reduced Hessian $H$ can be written

\[ h \tau g = g_S - S^T u_B \quad (4.6) \]

\[ H \tau g = g_S \quad (4.7) \]

such that the basic estimation $u_B$ of the constraints Lagrange multipliers solves the system

\[ g_B = B^T u_B \quad (4.8) \]
Matrix $H$ is likely very dense even for sparse matrices $Z$ and $G$. Since our problem is very large, we cannot afford to use matrix $H$, nor any of its approximations suggested in the literature. We suggest to use the Truncated Newton method at independent series of iterations (see Section 5), such that matrix $H$ does not need to be stored and the computer time and storage are within affordable limits. Note that system (4.9) is not needed to be completely solved at every iteration for getting, under mild conditions, a $Q$-superlinear rate of convergence /9/.

While maximizing in a given manifold, it is possible that either a basic or a superbasic arc strikes a bound during the search. If a superbasic arc strikes a bound then it becomes nonbasic, the cardinality of the basic-superbasic set on is reduced by one, and the search continues. If a basic arc strikes a bound then it is exchanged with an appropriate superbasic arc and the resulting new superbasic arc is made nonbasic. Note that the related pivoting and, then, the new (maximal) basis spanning tree may be easily obtained by using LP special data structures without any matrix manipulation; we use the data structures described in /18/.

5. OBTAINING THE SUPERBASIC STEPDIRECTION.

5.1. BRIEF REVIEW OF THE TRUNCATED NEWTON-METHOD.

See in /9/, /10/ the motivation for using the Truncated Newton methodology when 'solving' sistem $(4.9)$; it is a natural extension of the conjugate gradient method for solving system

$$-H_d\mathbf{s} = \mathbf{h} = 0 \quad (5.1)$$

At each iteration, say $i$ (here, termed minor iteration) of the conjugate gradient method, a steedirecțion $\mathbf{s}_i(\alpha)$ is obtained as a linear combination of the residual error $e^{(i-1)}$, $H_d^{(i-1)} \mathbf{h}$ and the steedirecțions...
Let $d_S^{(1)} = \alpha_S^{(1)} + e_S^{(1)} \phi_S^{(1)}$ be the solution —
(provably, inexact) of system (5.1), where —
$\alpha^{(1)}$ is the (exact) steplength that solves —
the quadratic problem

$$
\min \left\{ e^{(1)'} t^{(1)} S^{\frac{1}{2}} \left( e^{(1)} - e^{(1)} t^{(1)} S^{\frac{1}{2}} e^{(1)} \right) \right\}
$$

(5.2)

Note that $e^{(1)}$ can also be written

$$
e^{(1)} = e^{(1)} - \alpha^{(1)} \phi_S^{(1)}
$$

(5.3)

If $|| e^{(1)} ||$ satisfies the accuracy tolerance $\varepsilon_1$
within major iteration $\ell$, then $d_S = d_S^{(1)}$ is the
truncated solution of system (5.1). The procedure is as follows:

**SKELETAL ALGORITHM A1**

1. **Assign**
   - $e^{(0)} = -h$
   - $e^{(1)} = e^{(0)}$
   - $q^{(1)} = H e^{(1)}$
   - If $e^{(1)} > e^{(1)} - \varepsilon_1 || e^{(1)} ||$ and $q^{(1)} > q^{(1)}$ and $d_S > d_S^{(1)}$
     - stop
   - $d_S^{(1)} = e^{(1)} - \alpha^{(1)} \phi_S^{(1)}$

2. **Stopping rule**
   - If $|| e^{(1)} || > \varepsilon_1 || e^{(1)} ||$ and $d_S > d_S^{(1)}$
     - stop
   - $i = 2$

3. **Solution of system**
   - $q^{(1)} = H e^{(1)} - \varepsilon_1 || e^{(1)} ||$ and $q^{(1)} > q^{(1)}$ and $d_S > d_S^{(1)}$
   - stop
   - $d_S^{(1)} = e^{(1)} - \alpha^{(1)} \phi_S^{(1)}$

4. **Stopping rule**
   - If $|| e^{(1)} || > \varepsilon_1 || e^{(1)} ||$ and $d_S > d_S^{(1)}$
     - stop
   - $i = i + 1$, go to (2)

It can be shown /9/ that for $\varepsilon_1 > 0$ small enough, the
—
steplength $\alpha > 0$ is ascent enough in the vicinity
of the local optimal point $x$ of the
——
current manifold (if $-H$ is positive definite
—
and $\eta > 0$ for $\ell = 0$), and the above algo-
——
ritm is globally convergent; in addition, if

- $\hat{H}$ is pd the rate of convergence on $[x] - x$
---

superlinear iff

$$
\lim || e^{(t)} || / || h || = 0 \text{ (i.e., } \eta > 0) \text{ (5.4)}
$$

---

such that its order is $t+1$, where $0 < t$ if

$$
\lim || e^{(t)} || / || h || = t < 0
$$

(5.5)

Thus, tolerance $\eta$ can be written

$$
\eta = \min \{ \beta(0), \gamma(1) \}
$$

(5.6)

for $0 < \gamma < 1$ and $\gamma > 0$; for $t = 1$ the rate of con-
vergence is quadratic as the Newton method.

If $|| h ||$ is large ($X$ is away from $X$), only
—
few minor iterations are required for obtain-
ing $d_S^{(1)}$ when $X$ is getting close to $X$ then $|| h || = 0$ which implies $\eta = 0$ and, then, $d_S$ is getting close to a Newton steploration. To-
---
tolerance $\varepsilon_1$ is a safeguard against unstabili-
——
ties on calculating $q^{(1)}$. Although $\varepsilon_1$ only
---
needs to be positive, it is other safeguard;
---
it avoids that $d_S$ is not ascent (e.g., $-H$
---
is not pd). Typical values, $\varepsilon_1 = \varepsilon_1^2 / 2$, where
---
$\varepsilon_1$ is the machine precision in floating
——
—
point calculations (in our case, 10E-15), $\varepsilon_1
——
= 0.1$ and $\gamma = 0.1 / \ell'$, where $\ell'$ is the
——
major iteration of the subproblem defined by
the current manifold; $\ell'$ is reset to 1 when
set $\bar{S}$ is changed.

### 5.2 INDEPENDENT SETS OF SUPERBASIC ARCS

Note that algorithm A1 does not require the
---
calculation of any Hessian matrix, but the
—
product

$$
q^{(i)} = H e^{(i)} = z^{(i)} G z^{(i)}
$$

(5.7)

For obtaining $q^{(i)}$, superbasic set $\bar{S}$ is par-
---
titioned into, say $|P|$ disjoint so-termed indepen-
---
dent superbasic sets, such that

$$
\bar{S} = \bigcup_{P \in P} \bar{S}(P)
$$

(5.8)

For stating the necessary and sufficient con-
---
ditions for the unique valid partition (5.9), let
---
$S(\bar{S} = S(P)$ define the set of basic arcs co-
---
covered by the superbasic arcs included in set
---
$\bar{S}(P)$; that is,

$$
\bar{S}(P) = \bigcup_{k \in \bar{S}(P)} S_k
$$

(5.10)
(i) Superbasic arc \( k \) will be included in set \( \mathfrak{S}(p) \) if the following condition is satisfied
\[
\mathfrak{B}(p) n_{\mathfrak{S} \setminus \{ k \}} \neq \emptyset
\]  
(5.11)
That is, two superbasic arcs will belong to the same independent superbasic set (and, then, they will be simultaneously used for obtaining the independent superbasic stepdirection \( d_{\mathfrak{S}}(p) \)) if, at least, any flow change in one of them effects the other’s solution feasibility. Note that condition
\[
\mathfrak{B}(p) n_{\mathfrak{S} \setminus \{ k \}} \neq \emptyset \quad \text{is not sufficient, since it could be possible that } \mathfrak{B}(p) \text{ and } \mathfrak{B}(k) \text{ are disjoint sets and the following condition is satisfied}
\]
\[
\mathfrak{B}(p) n_{\mathfrak{S} \setminus \{ k \}} \neq \emptyset \quad \text{and} \quad \mathfrak{B}(p) n_{\mathfrak{S} \setminus \{ k \}} \neq \emptyset
\]  
(5.12)

(ii) Two superbasic arcs, say \( k \) and \( \ell \), will belong to the same independent set if any flow change in any of them effects the other’s objective function coefficient.

Let \( \mathfrak{c}(p) \) define the set of basic and superbasic arcs to be used for obtaining \( d_{\mathfrak{S}}(p) \).

Superbasic arc \( k \) will be included in set \( \mathfrak{S}(p) \) if the following condition is satisfied
\[
3_{G_{\mathfrak{g}}} \neq 0 \text{ such that}
\]
\[
(g \epsilon(k) u_{\mathfrak{S}_k}) \wedge (g' \epsilon \mathfrak{c}(p))
\]  
(5.13)

Then, arc \( k \) will be included in set \( \mathfrak{S}(p) \) if conditions (5.11) or (5.14) are satisfied.

Now, assume that there is other independent set, say \( \mathfrak{S}(q) \) for which, at least, one of the two following conditions is satisfied
\[
\mathfrak{B}(q) n_{\mathfrak{S} \setminus \{ k \}} \neq \emptyset
\]  
(5.15a)
\[
G_{\mathfrak{g}} \neq 0 \text{ such that}
\]
\[
(g \epsilon(k) u_{\mathfrak{S}_k}) \wedge (g' \epsilon \mathfrak{c}(q))
\]  
(5.15b)

In this case, the new independent set will be
\[
\mathfrak{S}(p) \cup \mathfrak{S}(q) \cup \{ k \} u_{\mathfrak{S}_k}
\]  
(5.16)

Since \( \mathfrak{B}(k) \) is small, \( G \) very sparse and, likely, \( \mathfrak{c}(p) \) and \( \mathfrak{c}(q) \) are also small (note that the de-activating process is only used for optimal and quasi-optimal solutions in the current manifold) then the number of situations for which conditions (5.15) are satisfied, could be reduced without too-much time-consuming. In any case, analysing the two above conditions is not time-consuming.

Let \( \mathfrak{V}_{\mathfrak{g}} \epsilon \mathfrak{c}(p) (\mathfrak{S}(p)) \) define the set of variable-coefficient linear variables whose coefficients are only related to nonbasic set \( \mathfrak{S} \).

Let \( \mathfrak{c}(p) \) define the set of basic and superbasic arcs to be used for obtaining \( d_{\mathfrak{S}}(p) \).

Note that at major iteration \( \ell \), \( |P| \) independent iterations are consecutively executed; note also that there are (i) minor iterations to be executed at each iteration \( p \) for \( p \in P \).

The advantages of using independent sets at successive major iterations are as follows.

1. The computational time for obtaining vector \( q(1) := \mathfrak{z}(G \mathfrak{c}(1)) \) is drastically reduced.

2. Faster minor iterations at the price of more (but much cheaper) major iterations. Note that the elements of matrix \( G \) out of set \( \mathfrak{c}(p) \) are not modified after obtaining \( d_{\mathfrak{S}}(p) \). Note also that only the elements of matrices \( G_3 \) and \( G_4 \) related to set \( \mathfrak{c} \) and the gradient related to set \( \mathfrak{c} \). PLU factors are to be evaluated at the beginning of a given major iteration.

3. Strong reduction on the number of arcs -- (i.e.), cardinality of \( \mathfrak{c}(p) \) to be used -- or obtaining the steplength related to set \( \mathfrak{c}(p) \). Note also that only the terms of set \( \mathfrak{c}(p) \) in the objective function -- (3.7) are to be recomputed for obtaining the objective function value \( F(X_{\mathfrak{g}}(p)) \) related to each trial step.
5.3. REDUCED GRADIENT USED AS A SUPERBASIC STEPDIRECTION.

Assume that the p-th step direction is being obtained for $p \in P$. Assume that $
abla'(q) = \{0\}$; in this case, a LP-network subproblem must be maximised, such that the related step direction $d_{\mathcal{S}}'$ can be expressed

$$d_{\mathcal{S}}' = \sum_{k \in \mathcal{S}} \delta_{\mathcal{S}}(k) \cdot s_{\mathcal{S}}(k),$$

where $s_{\mathcal{S}}$ is the $\mathcal{S}$-step direction.

Note that, in this case, $\alpha^0(p) = \alpha^0_{\mathcal{S}}(p)$. See also Section 6.6.

5.4. OBTAINING VECTOR $q^{(i)}$ IN THE TRUNCATED NEWTON METHOD.

Assume that $q^{(i)}$ is related to the superbasic step direction $d_{\mathcal{S}}(p)$; see Section 5.1. Let $\pi(v)_p, \pi(v)_{\mathcal{S}}$, and $\pi(v)_{\mathcal{S}}(p)$ define the complement of set $\pi(v)_{\mathcal{S}}(p)$.

1. Obtain intermediate vector $z^{(i)}_{\mathcal{S}}(p) = \pi(v)_{\mathcal{S}} q^{(i)}_{\mathcal{S}}(p)$, such that

$$q^{(i)}(p) = \sum_{k \in \pi(v)_{\mathcal{S}}(p)} \delta_{\mathcal{S}}(k) \cdot s_{\mathcal{S}}(k),$$

where $s_{\mathcal{S}}$ is the $\mathcal{S}$-step direction.

Note that the $(0, \mathcal{S})$ matrix $\rho$ is not required to be explicitly stored; we use data structures very similar to those described in /18/.

2. Obtain intermediate vector $z^{(i)}_{\mathcal{S}}(p)$, such that

$$z^{(i)}_{\mathcal{S}}(p) = \sum_{k \in \pi(v)_{\mathcal{S}}(p)} \delta_{\mathcal{S}}(k) \cdot s_{\mathcal{S}}(k),$$

where $s_{\mathcal{S}}$ is the $\mathcal{S}$-step direction.

Computation of vector $z^{(i)}_{\mathcal{S}}(p)$ is very fast since each row of $G_3$ and $G_4$ has only two and three nonzero elements, respectively, the elements of $\delta_{\mathcal{S}}(k)$ related to set $\pi(v)_{\mathcal{S}}(p)$ are zero (and, therefore, the related columns of $G_3$ and $G_4$ are not used), and only the rows of the Hessian matrix related to set $\pi(v)_{\mathcal{S}}(p)$ are used.

3. Obtain vector

$$q^{(i)}(p) = \sum_{k \in \pi(v)_{\mathcal{S}}(p)} \delta_{\mathcal{S}}(k) \cdot s_{\mathcal{S}}(k),$$

such that

$$q^{(i)}(p) = \pi(v)^{\mathcal{T}}(p) \cdot \pi(v)_{\mathcal{S}}(p) \cdot \delta_{\mathcal{S}}(p) \cdot s_{\mathcal{S}}(p),$$

where $\pi(v)^{\mathcal{T}}(p)$ is the transpose of $\pi(v)_{\mathcal{S}}(p)$.

6. DE-ACTIVATING STRATEGY.

6.1. DEFINITIONS.

Let the following stopping tests (with values true and false) for the optimization on the manifold, provided that the solution $X = X + \Delta$ is feasible and ascent.

$$t_1 := ||h|| \leq \varepsilon_2 \Rightarrow d ||/1+P(x)|| \leq 2 \Rightarrow \varepsilon_3 \Rightarrow \text{termination},$$

$$t_2 := ||h|| \leq \varepsilon_5 \Rightarrow \text{major iterations}.$$
Let us define indicator $\gamma_k$ for $k\in \tilde{N}$ as follows. $\gamma_k=0$ means that nonbasic arc $k$ is not a candidate to be de-activated; otherwise, it takes the sign of its de-activating direction (+ for up-direction and - for down-direction). A nonbasic arc will not be a candidate to be de-activated if it is an unsafe arc, the pricing is not favorable or it is a blocked arc; see Sections 6.2 and 6.3.

Let $\tilde{B}$ define the set of nonbasic arcs to be de-activated; that is, the arcs that will be moved from the nonbasic set to the superbasic set. Let $\tilde{N}_B = \bigcup_{p \in \tilde{B}} (p)$ and $\tilde{B}(p) \cap N \{d \} = \{g\}$, where $\tilde{B}(p)$ is the independent nonbasic set to be de-activated and, then, joined with independent superbasic set $\tilde{N}_B$. A candidate nonbasic arc will not be de-activated if $|B|$ is at its (upper) bound and there is, at least, other candidate arc with higher first-order guarantee for a stronger increase in the objective function; see Section 6.4.

6.2. PRICING NONBASIC ARCS.

When a solution on the current manifold is quasi-optimal, $\gamma_k=0$ for all $k\in \tilde{U}$; when the solution is optimal, set $\tilde{U}$ is declared empty.

The nonbasic Lagrange multipliers estimation $\lambda_k$ can be written

$$\lambda_k = \begin{cases} g_k & \textbf{k} \in \tilde{N} \\ -\tilde{g}_k & \textbf{k} \not\in \tilde{N} \end{cases}$$

(6.1)

For $\tilde{N}_k = \tilde{N}_k$, (lower bound), $\gamma_k=1$ if $g_k<\epsilon_7$; otherwise, $\gamma_k=0$.

For $\tilde{N}_k = \tilde{N}_k$, (upper bound), $\gamma_k=1$ if $\lambda_k<\epsilon_7$; otherwise, $\gamma_k=0$. Note that $g_k$ is the usual $k$-th gradient element for $\tilde{N}_k = \tilde{N}_k$, (intermediate bound), $g_k$ for $k\in \tilde{B}$ is assigned as follows: $g_k^+ = g_k$, for $g_k^+ < k\in \tilde{B}$ or $g_k^+ = g_k$, for $g_k^+ > k\in \tilde{B}$, otherwise (see below).

In general, $g_k^+$ and $g_k^-$ are the gradient elements of arc $k$ ($g_k^+=g_k$ for $\tilde{N}_k=\tilde{N}_k$) related to its up-direction and down-direction, respectively. Note that $g_k^+ = g_k$ and $g_k^- = g_k$ for $\tilde{N}(t, j)$, see (3.7); it is assigned $\max(0, g_k^- - g_k^+)$ for obtaining $g_k^+$ for $\tilde{N}(t, j)$ such that $g_k^+ = g_k^+ \text{ for } t = t$, $\tilde{t}$. Note that the computation of $g_k$, (and, then, $\tilde{g}_k$)

for $k\in \tilde{B}$ and $\tilde{k}\in \tilde{B}$, is based on $\rho_k, \ell$ and $\gamma_k$; then, the ambiguity of $g_k^+$ for $k\in \tilde{B}$ is solved by blocking the nonbasic arc that satisfies test $t_5$ (see below); note that there is not a null step for $t_5$ since $\ell$ is used as the step-direction of set $\tilde{B}$ (see Section 6.6) and a maximal basis spanning tree is assumed (i.e., $\tilde{k}\in \tilde{B}$) and $\gamma_k$ such that $\tilde{g}_k$, is at any of its bounds.

For $\tilde{N}_k = \tilde{N}_k$, (intermediate bound), $\gamma_k$ is assigned as follows:

$$\begin{cases} 0 & \text{if } (g_k^+ - \epsilon_7)^\wedge (g_k^- - \epsilon_7) \\ + & \text{if } (g_k^+ - \epsilon_7)^\wedge (g_k^- - \epsilon_7) \\ j & \text{such that } (g_k^+ - \epsilon_7)^\wedge (g_k^- - \epsilon_7) \end{cases}$$

Finally, assign $g_k^+ = g_k^-$ for $t_5=\gamma_k$ and, then, expression (6.1) may be used for $\lambda_k$.

Let the following anti-zigzagging test for any nonbasic arc being priced out.

$$t_4 = ||h||^2 \epsilon_8 \gamma_k$$

where $\epsilon_8$ is a positive tolerance (typically, 0.9). When the solution on the current manifold is quasi-optimal, arc $k$ will not be considered as a candidate to be de-activated (and, then, $\gamma_k=0$) if $t_4$.

6.3. BLOCKING NONBASIC ARC.

A maximal basis spanning tree avoids degenerate basic-supercritical pivots, but it does not prevent null steps when a nonbasic arc is de-activated /8/. Therefore, a mechanism is needed for testing whether a nonbasic arc, say $k$, must be considered as a candidate to be de-activated. It may be carried out at the same time the nonbasic arc is priced (i.e., its Lagrange multiplier estimate is calculated) and, then $\gamma_k$ is set to 0 if otherwise a null step could not be prevented. Thus, $\gamma_k=0$ if $t_5\leq t_6\leq t_7$, where $t_5$, $t_6$ and $t_7$ are the result of the following blocking tests:

$t_5$: $\lambda_k = \min(||\lambda_k||, ||\gamma_k||)\tilde{R}_k = \tilde{a}_k$, for $k\in \tilde{B}\cap \tilde{R}_k$ such that $t_5\leq t_6\leq t_7$ and $\lambda_k = \min(||\lambda_k||, ||\gamma_k||)\tilde{R}_k = \tilde{a}_k$, for $k\in \tilde{B}\cap \tilde{R}_k$ such that $t_5\leq t_6\leq t_7$.
Note that if the flow in basic arc $k^*$ changes in the same direction for any flow change in the appropriate direction of arcs $k$ and $\ell$, given by $\lambda_k$ and $\lambda_\ell$, respectively.

t6 (Case $\gamma_k=\ast$): $k^* \in \mathbf{B}_k$ such that

- $p_{k^* k} = -1$ (reverse) or $p_{k^* k} = 1$ (forward) and
- $\bar{x}_{k^* k} = \bar{z}_{k^* k}$

t7 (Case $\gamma_k = \ast$): $k^* \in \mathbf{E}_k$ such that

- $p_{k^* k} = -1$ (reverse) or $p_{k^* k} = 1$ (forward) and
- $\bar{x}_{k^* k} = \bar{z}_{k^* k}$

If $t5 \lor t6 \lor t7$ we refer to arc $k$ as a blocked arc, and then, it will not be a candidate to be de-activated.

### 6.4. Obtaining Set $\mathbf{B}$ to be De-Activated

A multiple de-activating strategy is allowed such that as many as possible candidate nonbasic arcs are to be de-activated at a given bound, say $\min(\tau_3, \varepsilon_9 \abs{\overline{N} / \mathbf{B}})$, where $\tau_3$ and $\varepsilon_9$ are positive tolerances (typically $\tau_3=30$ and $\varepsilon_9=0.5$). For reducing the computational time and storage required by the Truncated-Newton method, it is interesting that the cardinality of any independent set $\mathbf{B}(p) \subset \mathbf{B}(p)$ does not exceed a given bound, say $\tau_4$ (typically, 60).

Given the dimensions of our problem, we suggest to use for partial pricing if $\min(\tau_3, \varepsilon_9 \abs{\overline{N} / \mathbf{B}}) \geq \tau_{10}(a-n)$, where $\tau_{10}$ is a positive tolerance (typically, 0.1), such that only a subset of $\overline{N} / \mathbf{B}$ is priced at each de-activating process. Basically, the procedure consists in pricing sequentially the arcs in set $\overline{N} / \mathbf{B}$ so that they will be candidate to be de-activated if the pricing result is favorable.

Once $\mathbf{B}$ reaches the allowed bound, the next candidate scanned arc will replace the arc from set $\mathbf{B}$ with the worst pricing result till $\tau_4$ is not greater than a given bound, say $\varepsilon_{11}$ (typically, 0.1), $\tau$ takes the ratio of the number of replacements to the number of candidate scanned arcs. When $\varepsilon_\varepsilon_11$ the -- scanning is interrupted; it will be restarted, at the next de-activating process, by -- pricing the arc where it was left out.

Let the following de-activating tests:

$t8 : \abs{\overline{B}} < \min(\tau_3, \varepsilon_9 \abs{\overline{N} / \mathbf{B}})$

t9 : $\lambda_{k^*} > \min(\abs{\lambda_\ell \psi \varepsilon \mathbf{D}})$ \& $t8$

t10 : $\abs{\overline{N} / \mathbf{B}} \geq \tau_{10}(a-n)$

$t11 : \varepsilon_\varepsilon_11$

Formally, $\delta_{k^*} \mathbf{D} u(k)$ for $\gamma_k \neq 0$ if $t8 \lor t9$.

After the de-activating process, if the solution on the current manifold is quasi-optimal, tolerance $\varepsilon_5$ is reset to $\varepsilon_5 := \min(\varepsilon_2, \varepsilon_5(1-\varepsilon_{12}))$ (even if $\mathbf{D}(\emptyset)$),

where $\varepsilon_{12}$ is a positive tolerance (typically, 0.3). If the solution is optimal, $\mathbf{B} \setminus \mathbf{D}(\emptyset)$ is declared empty and $\varepsilon_5 := \| \lambda_k \|_{2}$ for $\mathbf{D}(\emptyset)$ if $\mathbf{D}(\emptyset)$, stop since it is assumed that the optimal solution of the problem has been found; note that $(t5 \lor t6 \lor t7)$ for $t5 \lor t6 \lor t7 = 0$; that is, there are no blocked arcs when the optimal solution of the current manifold is found -- and there is no nonbasic arc with a favorable pricing result.

Finding the most suitable values for the tolerances is a subject for experimentation, mainly for the multiple pricing tolerance $\tau_3$.

### 6.5. Obtaining the Independent Set $\mathbf{B}(p)$ to be De-Activated

Recall $\delta_{k^*} \mathbf{B}(p)$ and $\mathbf{B}(p)$ and $\mathbf{B}(q)$ are independent sets. Let $\mathbf{C}(p), \mathbf{C}(p) \cup \mathbf{B}(p)$, and $\mathbf{B}(p)$ be $\mathbf{D}(p) \cup \mathbf{B}(p)$. An arc, say $k$, is to be de-activated could be included in set $\mathbf{B}(p)$ if any move $d_k \neq 0$ effects the solution feasibility or the objective function coefficient of any arc from set $\mathbf{C}(p) \cap \mathbf{B}(p)$ -- formally, $\mathbf{B}(p) \setminus \mathbf{B}(p) \cup \mathbf{B}(p)$, for $k \neq \mathbf{B}(p)$ if $t12 \lor (t13 \lor t14)$, where $t12$ for $w=1$, $t13$ and $t14$ are the result of the following including tests:

$t12 : \abs{\mathbf{B}(p)} + \abs{\mathbf{B}(q)} + \varepsilon_9 \geq \tau_4$

t13 : $(\mathbf{B}(p) \cup \mathbf{B}(p)) \cap \mathbf{B}(p)$

t14 : $3 \mathbf{G}_{k^*} \neq \emptyset \{ g \in G(k) \in \mathbf{B}(p) \}$

It is suggested to perform the testing in the following sequence:

$t12, t13, t14$

How, to assure that sets $\mathbf{B}(j) \forall j \in \mathbf{P}$ are independent, it is required to analyze if any $\mathbf{Q}$
\textbf{Questa} - V. 7, n. 2 (juny 1983)

ve $d \neq 0$, $k \in B(p)$ will effect the solution feasibility or the objective function coefficient of any arc from set $B(q)$ $\forall q \in P(p)$; - in that case, both sets must be joined. Formally, $\tilde{B}(p) \cup \tilde{B}(p) \cup B(q)$ and $\tilde{B}(q) \cup \tilde{B}(p)$ if --- t12$\wedge$(t15$\vee$t16) such that $\sigma \in |\tilde{B}(q)| \vee |\tilde{B}(q)|$ -- for t12 and where t15 and t16 are the result of the following joining tests:

\begin{align*}
t_{15} & : \{ (\tilde{B}(p) \cup \tilde{B}(p)) \wedge (\tilde{B}(q) \cup \tilde{B}(q)) \neq (\tilde{B}) \} \\
t_{16} & : \exists q \in P \wedge (g \in \tilde{C}(p) \cup \tilde{C}(p)) \wedge (g \in \tilde{C}(q) \cup \tilde{C}(q))
\end{align*}

It is suggested to perform the testing in the following sequence: t12, t15 and t16.

If $k \in \tilde{B}$ is not included in $B(p)$ (and, then, $j \in P(p) \vee t$, $\tilde{B}(j) \notin \tilde{B}(k)$ and $\tilde{B}(j) \notin B(p)$) or $B(q)$ is not joined to $B(p)$ due to t12 then $B(j)$ - the first case and $\tilde{B}(q)$ in the second case - will be used for obtaining $d(p)$ and $\sigma_m$.

Selecting nonbasic arcs to be de-activating may be performed in several ways; one of the main criteria could be reducing $|\tilde{B}(p)| \forall q \in P(p)$ and, then, increasing the number $|P(p)|$ of independent sets of arcs to be used for obtaining the stepdirections of the next manifold. Finding the best procedure is left open at this point.

\section{SOLVING THE NEWTON EQUATION AFTER DE-ACTIVATING}

The new ascent independent stepdirection --- $d_S(p) = (d_S(p), d_S(p), d_S(p))$, where $d_S(p)$ takes the stepdirection related to the old superbasic set $\tilde{B}(p)$ and $d_S(p)$ is related to set $\tilde{B}(p)$ is obtained as follows.

\begin{align*}
d_S(p) = & 0 \text{ if } \|h\|_2 = \varepsilon_2 \sqrt{\tilde{B}(p)} = (\emptyset) \\
& \text{Truncated-Newton direction in } \tilde{B}(p) = h, \text{ otherwise}
\end{align*}

\begin{align*}
g_S(p) = & \{ h_k \quad k \in B(p) \}
\end{align*}

where $h$ and $H$ are related to the current solution $x = x_0$ and $h_k = h_k$. Note that the new direction is a mixture of the scaled steepest ascent direction and an accurate truncated Newton direction; since the flow change in set $\tilde{B}(p)$ has the appropriate direction (see Sections 6.2 and 6.3) then a null step is avoided provided that $\tilde{B}(p)$ is a maximal basis spanning tree.

\section{FUTURE WORK}

The work covered by this paper is in progress and we cannot say too much about computational experience. We have experimented with a shorter version (6 reservoirs and 24 time periods) of the real-life problem. It seems that the ideas described in this paper (mainly, the concept of independent sets of superbasic and de-activated nonbasic arcs) are worthy of extensive experimentation; a careful implementation of the algorithm is planned. In the sequel of this paper /11/ we are planning to report the results of the current algorithm as well as the results of the modifications described below.

Computational time is important because the model is to be run frequently for planning purposes under several assumed inflow patterns. It is important to solve for the 6-reservoirs problem efficiently if the computing time for the full 40-reservoirs system is to be within affordable limits; in some cases, aggregating the last, say 14 weeks of the time horizon into 3 time periods (months) does not strongly deteriorate the planning objective.

Topics that are worthy of future experimentation are the following.

1) Selecting candidate nonbasic arcs.

The procedure could promote small independent sets of arcs to be de-activated, while de-activating as many as possible arc up a given bound and the first-order estimation of the objective function gains is not strongly deteriorated.

2) Maximizing the hydropower benefit function.

The important property of this function is temporal separability and spatial non-separability. Since there is an economic interaction between the decision variables of all reservoirs in the same time period, the whole system must be treated in the same run, even if the reservoirs are not physically interconnected. By using the strategies described in /18/ for selecting candidate nonbasic arcs,
it seems that the Truncated Newton methodology could be used to obtain independent superbasic stepdirections at each major iteration.

3) Activating as many as possible superbasic arcs at each major iteration.

Note that, barring exceptional circumstances, at most one basic-superbasic arc per set $C(p)$ per $p$ can be added to nonbasic set $N$ in the algorithms based on manifold suboptimization and active set strategies as above; so if, say 1000 arcs are active at the optimal solution $\bar{x}$ and the initial feasible solution is interior, then the method will require at least 1000 major iterations to converge. Let

$$a_B^{(p)} = \min \{|\bar{x}_k - a_k|/d_k| \quad W_k e_B^{(p)}\}$$

(7.1)

$$a_S^{(p)} = \min \{|\bar{x}_k - a_k|/d_k| \quad W_k e_S^{(p)}\}$$

(7.2)

denote the upper bounds on steplength $a^{(p)}$ for keeping feasibility on sets $B(p)$ and $S(p)$ respectively such that $a^{(p)} = \min \{a_B^{(p)}, a_S^{(p)}\}$. If $a^{(p)} = a_S^{(p)}$, $a$ is obtained with a traditional linesearch $/12/$ that preserves $\alpha$-superlinear convergence. If $a^{(p)} = a_B^{(p)}$, fewer major iterations could be required by allowing more that one superbasic arc from set $S(p)$ to be activated at each major iteration; given the special structure of matrix $\rho$ (recall that it is not explicitly stored), the time consuming is likely to be within affordable limits.

See in $/4/$ a nonlinear algorithm for a general system of linear constraints; the objective function $F(X)$ is expressed as a function of $Y$ such that $Y = \bar{A} X$, where $\bar{A}$ takes the submatrix related to 'active' constraints and bounds and, then $Y$ gives the solution vector of 'active' slack and structural variables. A key point in the new algorithm is the transformation $X = T(Y)$ such that the problem, at each major iteration, becomes: $\min_{\bar{a}} \arg \max_{l} f(\bar{Y} + \bar{a}_l)$: $l, \bar{y}, \bar{a} + \bar{s}_u, 0 < \bar{s}_a$, where $\bar{Y}$ is the feasible solution at the previous iteration. The feasible set is referred to as the active rectangle at the current iteration. The algorithm has, under mild conditions, $\alpha$-superlinear convergence.

In out context, a version of this algorithm is as follows. Assume that the independent superbasic stepdirection $d_s^{(p)}$ has been obtained as follows: $d_k = h_k \quad W_k I^{(p)}$, where $I^{(p)}$ denotes the set of quasi-active arcs in the $p$-th superbasic set such that

$$I^{(p)} = \{k | e_S^{(p)} | \bar{x}_k - a_k| < \epsilon(p)\}$$

(7.3)

where scalar $\epsilon(p)$ is given by

$$\epsilon(p) = \min_{1 \leq i \leq n} \{ |\bar{x}_i| |\bar{x}_j - a_j| \}$$

(7.4)

for $\epsilon_1 > 0$ (typically, 0.01), $[.]^\#$ will be described below and, by slight abuse of the active bound notation used in Section 4, $a_k$ denotes the active bound in the direction of the sign of $h_k$. Let $I^{(p)} = \{k | e_S^{(p)} | \bar{x}_k - a_k| < \epsilon(p)\}$, $d_k = 0$ for $k \in I^{(p)}$ is a Truncated Newton direction and $d_k = 0$ for $k \bar{\in} I^{(p)}$. Any new feasible solution $X = (X_B^{(p)} X_v^{(p)} X_S^{(p)})$, $x^{(p)} = (x_k \quad W_k e_S^{(p)})$, takes the form

$$x_S^{(p)} = x_S^{(p)} + a^{(p)} d_S^{(p)} \# [x] \#$$

(7.5a)

$$x_B^{(p)} = x_B^{(p)} + \rho(p) (X_S^{(p)} - z_S^{(p)})$$

(7.5b)

where $\rho(p)$ is the submatrix of $\rho$ related to set $C^{(p)}$, $0 < \rho(p)$, and $\# [.]^\#$ denotes the projection on the feasible superbasic rectangle for $x_S^{(p)}$ such that, it results

(i) Case $x_k < a_k$: $\epsilon_S^{(p)}$

$$[x]_k = \begin{cases} a_k \quad \text{if} \quad x_k \leq a_k \\ a_k \quad \text{if} \quad x_k > a_k \end{cases}$$

(ii) Case $x_k = a_k$: $\epsilon_S^{(p)}$

$$[x]_k = a_k$$

(iii) Case $x_k > a_k$: $\epsilon_S^{(p)}$

$$[x]_k = \begin{cases} a_k \quad \text{if} \quad x_k \leq a_k \quad (\text{case } \gamma_k = -1) \\ a_k \quad \text{if} \quad x_k \leq a_k \quad (\text{case } \gamma_k = +1) \end{cases}$$

Note that the strategy used for de-activating nonbasic arcs (see Section 6) prevents degenerate pivots, provided that a maximal basis -- spanning tree is used.

Following the same approach described in $/4/$, it can be shown that, under mild conditions, the solution $Y^{(p)}$ in $\rho_S^{(p)}$ is feasible and as--

448
\[ a(p) = m(p) \] (7.6)

where \( m(p) \) is the first nonnegative integer \( m (0, 1, 2, \ldots) \) that satisfies the two following conditions

\[ s_m s_B(p) \] (7.7)

\[ F(x_B) - F(x_A) \geq \mu(s_m(\gamma(p)) \sum_{k \in I_n} h_k d_k + \sum_{k \in I_n} h_k(x_k - y_k) \] (7.8)

where \( \varepsilon(0, 1), \mu(0, 0.5) \) (typically, \( \varepsilon = 0.5 \) and \( \mu = 10^{-4} \)) and the scalar \( \gamma(p) \) is given by

\[ \gamma(p) = \min\{1, \frac{d_k}{\sum_{k \in I_n} h_k(x_k - y_k)} \} \] (7.9)

Note that for \( m > 0 \), the right-hand side of (7.8) is nonnegative and it is positive if \( x_B(p) \) is not an optimal solution in the current manifold.

It is suggested to start the linesearch with \( m_0 \) such that \( s_m = \min(a(p), b_0) \) if at any major iteration a fixed number \( m_0 \) (typically, 2 or 3) of trial steps fail to satisfy the Armijo-like condition (7.8), then -- \( \gamma(p) \) is used as the next trial value.

The main advantage that the new linesearch offers over the manifold suboptimization-based algorithm as described in the above sections is that as many as \( \varepsilon \) new arcs may become active in a single major iteration. However, the computational performance of the new approach and the comparison with the manifold suboptimization alternative is a subject for future experimentation.

8. ACKNOWLEDGEMENT.

I wish to thank Ron Dembo for various guiding comments and suggestions that have strongly improved the algorithm. In particular many thanks go to Phil Gill and Walter Murray for providing their linesearch routine.

9. REFERENCES.


/7/ R.S. DEMBO, "The design and implementation of algorithms for large-scale nonlinear network optimization", NETFLOW 83, Pisa (Italy), 1983.


/14/ M.D. GRIGORIADIS, "On the implementation of primal, dual and parametric network-simplex methods", Department of Computer Science, working paper, Rutgers University, New Brunswick, 1982.


/20/ M.Z. YAKIN, "Deterministic and stochastic water reservoir management models", College of Business Administration, working paper CBA 1982-98, University of Houston, Houston (Texas) 1982.