Qüestiió - V. 9, n.º 1 (març 1985) pp. 55-74

ON NONLINEAR REPLICATED NETWORKS

L. F. ESCUDERO
IBM MADRID SCIENTIFIC CENTER

In this paper we describe a new type of network flow problems that basically consists of the classical transshipment problem with the following extensions: (1) The replication of a network by producing subnetworks with identical structure, such that they are linked by so-called linking arcs; (2) The objective function terms related to the linking arcs are non-differentiable nonlinear functions. We also describe an implementation of a linearly constrained nonlinear programming algorithm which is fast and can solve large-scale replicated network flow problems; the major ideas incorporated are: (i) The sparsity and the structure of the constraints system is used to reduce the time and computer storage requirements; (ii) The new concept of independent superbasic sets is introduced, so that it allows to obtain in parallel independent pieces of the solution at each iteration; (iii) The predecessor, depth, transverse and reverse structures are specialized for the case of replicated networks; (iv) A bending, backtracking linesearch that allows to activate more than one basic-superbasic arc at each iteration; (v) A procedure for pricing nonbasic arcs in the presence of non-differentiable terms in the objective function.

Keywords: REPLICATED NETWORKS, INDEPENDENT SUPERBASIC ARCS, MAXIMAL BASIS SPANNING TREES, BENDING-BACKTRACKING LINESEARCH

1. INTRODUCTION. PROBLEM DESCRIPTION.

This paper describes an implementation of a nonlinear method for solving a special class of transshipment problems. The problem consists of finding the local optimum of a non-differentiable nonlinear function of a problem with a single-commodity flow in a special capacitated direct network satisfying suply-and-demand constraints. The speciality of the network structure is as follows: There are some applications, mainly in the communications, hydroelectric power and water distribution fields, on which a given network (so-called basic network) is replicated on a number of segments, such that they are linked by special arcs (so-called linking arcs) on some of the nodes and, then, producing a so-called replicated network. Let J denote the set of nodes (so-called basic nodes) in the basic network; i_k and j_k denote the from-node and to-node of a given arc, say k; let P_j denote the set of from-nodes of the arcs in the basic network whose to-node is the node j, and Q_j denote the set of to-nodes in the basic network whose from-node is the node j for $j \in J$; note that $\exists j \in J | P_j = \{\emptyset\}$ and $\exists i \in J | Q_i = \{\emptyset\}$. Let T denote the set (and, then, |T| the number) of replications on the basic network; let $W \subseteq J$ denote the set of basic nodes with linking arcs, and, then, $|W|>1$ by definition; let n(t,j) denote the node of the replicated network for $j \in J$ that is obtained by replicating the basic network on the segment t for $t \in T$; let a(t,j) denote the linking arc that connects the segment t-1 and t for $t \in T/\{1\}$ and $j \in W$ (note that without loss of generality we may assume that the set W is replicated on all segments); let a(t,j,i) denote the arc that connects tha basic nodes j and i on the segment t for $t \in T$, $j \in J$ and $i \in Q_j$, such that n(t,j) is the from node and n(t,i) is the to-node; we may assume w.l.o.g that the node n(t,i) for $Q_i = \{\emptyset\}$ is the same node (the root) for all segments.

An example of replicated networks. In the hydroelectric power problem, the basic network is given by a river such that the nodes represent the reservoirs and the arcs are given by the sections of the river that con-

- Laureano F. Escudero - IBM Scientific Center - Pº de la Castellana 4 - Madrid
- Article rebut el setembre de 1984.

nect the reservoirs; the single-commodity is the water that is stored in the reservoirs from one period to the next one, or is released from one reservoir to its direct downstream reservoirs (the cardinality of this set is not necessarily one); the segments are the time units (usually, weeks or days) of the planning horizon (such that if it consists of, say 52 weeks then the basic network is replicated 52 times); the basic nodes on which the replication is based are the reservoirs that may store water from one period to the next one; the linking arcs of any reservoir that may store water consist of its connexions along the planning horizon; the flow of the linking arcs is the water stored by the reservoir at the end of a given period and, then, available to be released or stored on the next periods; and the supply-and-demand is given by the water net exogenous inflow. The purpose of the bounds on the arcs 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 does not exceed its canal capacity; otherwise, the overflow is not used for producing electricity; and, third, to penalise the amount of stored water that exceeds a safety capacity in a given reservoir. The objective function consists of the maximization of a non-differentiable nonlinear function (typically, the generation of electricity) over the planning horizon. See /12/, /16/, /17/, /23/, /24/, /26/, among others. Assuming a multi-reservoir system of 25 reservoirs and a planning horizon of 52 time periods, the replicated network has over 1250 nodes and 2500 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/, /14/, /15/, /18/ among others); a sounding nonlinear network algorithm should require a computing time with the same order of magnitude of the linear ones /6/.

Recently, new algorithms have been designed for specializing linear primal data structures to nonlinear network flow problems; see /1/, /6/, /7/, /16/, /19/, /20/, /23/, /24/, and related references. Apart from recent developments (see /2/, /4/, /6/, /7/, /17/, among others), the methods use a linearized subproblem to generate stepdirections so that the reduced gradient methodology is used in the basic-nonbasic environment /24/ or in the more efficient basic-superbasic-nonbasic environment /21/; 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 transshipment problem for replicated networks, by using the new concept so-called independent superbasic sets; it allows to obtain in parallel independent pieces of the solution at each iteration. The TN method was introduced in /9/ for general unconstrained nonlinear problems, extended in /11/ to linearly constrained problems, and specialized in /6/ to general network flow problems. The new algorithm uses second-order information and allows non-differentiable terms in the objective function, given the special structure of both the objective function and constraints.

The paper is organized as follows. Sections 2 and 3 present the formulation and objective function of the problem. Section 4 summarises the variable-reduction environment of the algorithm. Section 5 presents the approach for obtaining the superbasic stepdirection in nonlinear replicated network flow problems, and introduces the new concept of independent superbasic sets. Section 6 describes the de-activating process and the procedure for pricing nonbasic arcs in the presence of non-differentiable terms in the objective function. Section 7 is devoted to the bending, backtracking linesearch. And, finally, Section 8 reports some computational experience.

2. PROBLEM FORMULATION.

Consider the replicated network described before. Associated with each node $n(t,j)$ is a value b_{tj} representing the amount of suply of a commodity available at the node; a negative value represents a demand for the commodity at the node. The decision variables are denoted r_{tji}, the flow on the arc $a(t,j,i)$ and s_{tj}, the flow on the linking arc $a(t,j)$. The flow balance equations for the nodes $\{n(t,j)\}$ for $j\in J$ and $t\in T$, except for the root node are

$$-\sum_{i\in P_j} r_{tij} - s_{tj} + \sum_{i\in Q_j} r_{tji} + s_{t+1,j} = b_{tj} \qquad (2.1)$$

where $s_{\ell j}$ is fixed for $\ell=1$ and $\ell=|T|+1$, and it is dummy for $\ell\in T$ and $j\in J/W$. The equation for the root node is not shown; it is implicity satisfied, since the related s-variables are dummy and the right-hand-side is the negative value of the summation of the b_{tj}-values of the other nodes and the $s_{\ell j}$-values for $\ell=1$ and $\ell=|T|+1$ for $\forall j\in W$. Letting X be a vector of all decision variables and b a vector of the supply-and-demand, system (2.1) can be written

$$AX = b \qquad (2.2)$$

where A is the node-arc incidence matrix. Each column of A corresponds to an arc and each row to a node of the replicated network. Multiple (or parallel) arcs $a(t,j,i)$ between the nodes j and i are allowed per each segment t; the arcs can each be given explicitly with associate objective function coefficients and bounds and would be treated as distint arcs.

The bounds on the arcs of the replicated network are given by

$$l_{tji} \le r_{tji} \le u_{tji} \qquad \forall i\in Q_j,\ j\in J,\ t\in T \qquad (2.3a)$$

$$m_{tj} \le s_{tj} \le M_{tj} \qquad \forall j\in W,\ t\in T/\{1\} \qquad (2.3b)$$

Following a traditional approach /21/, matrix A can be partitioned as follows.

$$A = (B\ S\ N) \qquad (2.4)$$

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

By construction of A it can be shown /5/ that any basis B may be ordered such that it is upper triangular, and the arcs corresponding to columns in any basis form a spanning tree of the network. It can also be shown /7/ that a maximal basis spanning tree for a given feasible solution avoids a basic-superbasic degenerate pivot; otherwise, null steps are more frequent than in problems with a general structure.

Let Z be the variable-reduction matrix whose columns form a basis for the null space of A, given AZ=0, such that

$$Z = \begin{pmatrix} -B^{-1}S \\ I \\ 0 \end{pmatrix} \begin{matrix} \}\ n \\ \}\ a-n \\ \ \end{matrix} \qquad (2.5)$$

where n denotes the number of nodes (without including the root) and a denotes the number of arcs. Let Basic-Equivalent-Path (BEP) define the unique path in the basis spanning tree that leads from the node i_k to the node j_k of the superbasic or nonbasic ark k; let β_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 $p(.)_k$ is the predecessor of node (.) in the BEP of arc k; it has a reverse orientation if $p(j_{k'})_k=i_{k'}$. Let ρ_k denote the column $(B^{-1}S)_k$ and, then, $\rho\equiv(B^{-1}S)$. The nonzero elements of ρ_k for $k\in\bar{S}\cup\bar{N}$ are +1 for a forward orientation and -1 for a reverse orientation.

3. THE OBJECTIVE FUNCTION.

The important properties of the objective function in nonlinear replicated networks are quasi-separability and non-differentiability. Let c_{tji} denote the objective function coefficient per unit of flow along the arc $a(t,j,i)$. The objective function term for the node $n(t,j)$ of the replicated network can be expressed as follows.

$$h_{tj} = K_{tj} \sum_{i\in Q_j} c_{tji}\, r_{tji} \qquad \forall t\in T,\ j\in J \qquad (3.1)$$

where

$$K_{tj} = \begin{cases} \Psi(s_{tj}, s_{t+1,j}) & \forall j\in W \\ 1 & \forall j\in J/W \end{cases} \qquad (3.2)$$

such that $\psi(.)$ is a nonlinear function. Note that $K_{tj}=0$ for the root node. Then, the objective function could be written

$$\max\left\{\sum_{t\in T}\sum_{j\in J} h_{tj}\right\} \tag{3.3}$$

Let PL denote the set of <u>pure linear</u> arcs a(t,j,i), VL the set of <u>linear</u> arcs a(t,j,i) with <u>variable coefficients</u>, and NL the set of <u>nonlinear</u> arcs a(t,j). Arc $a(t,j,i)\in PL$ for $j\in J/W$. Arc $a(t,j,i)\in VL$ for $j\in W$; note that h_{tj} is a linear function if s_{tj} and $s_{t+1,j}$ are fixed. Arc $a(t,j)\in NL$ for $j\in W$.

Let g_{tji} for $a(t,j,i)\in PL\cup VL$ and g_{tj} for $a(t,j)\in NL$ denote the gradient elements related to the r-arcs and s-arcs, respectively. Note that the gradient related to the set PL is constant, and g_{tji} for $a(t,j,i)\in VL$ does not change for fixed values of s_{tj} and $s_{t+1,j}$. The Hessian matrix G has the form

$$G = \begin{array}{c|c|c|c|} & PL & VL & NL \\ \hline & G_1=0 & 0 & 0 \\ \hline & 0 & G_2=0 & G_3 \\ \hline & 0 & G_3^t & G_4 \\ \hline \end{array} \tag{3.4}$$

such that G_3 is a two-diagonal matrix for $|Q_j|=1$ $\forall j\in W$ and G_4 is a symmetric tri-diagonal matrix; see an example in form (3.5)

G_3 = (3.5a)

	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}
r_{312}	X	X			
r_{412}		X	X		
r_{512}			X	X	
r_{612}				X	X
r_{712}					X

Note that the elements related to the same pair (t,j) in a given column of matrix G_3 differ only in the coefficient c_{tji} for all $i\in Q_j$, for $|Q_j|>1$, $j\in W$.

G_4 = (3.5b)

	s_{31}	s_{41}	s_{51}	s_{61}	s_{71}
s_{31}	X	X			
s_{41}	X	X	X		
s_{51}		X	X	X	
s_{61}			X	X	X
s_{71}				X	X

Let R_{tji} and T_{tj} denote the intermediate bounds for the flow on the arcs a(t,j,i) and a(t,j), respectively such that the overflow $r_{tji}-R_{tji}$ is not used in the objective function and the overflow $s_{tj}-T_{tj}$ is to be penalised. Then, the objective can be expressed by (3.6) instead of using (3.1) and (3.3).

$$\max\left\{\sum_{t\in T}\sum_{j\in J} h_{tj} - \sum_{t\in T}\sum_{j\in W} P_{tj}\max\{0, s_{tj}-T_{tj}\}\right\} \tag{3.6a}$$

where

$$h_{tj} = K_{tj}\sum_{i\in Q_j} c_{tji}\min\{r_{tji}, R_{tji}\} \tag{3.6b}$$

and P_{tj} represents a penalty coefficient or function. The nondifferentiability introduced by (3.6) can be treated without great difficulty (see Section 6).

As an example of the objective function on nonlinear replicated networks, consider again the hydroelectric power problem. The electricity to be generated by the reservoir j at the time period t can be given by the expression (3.1), where $c_{tji}=1$, K_{tj} $\forall t\in T$ is a constant for the (run-of-the river) reservoir $j\in J/W$, and $\Psi(.)$ is a nonlinear function on the average of the water levels, say w_{tj} and $w_{t+1,j}$ at the beginning of the periods t and t+1, respectively; w_{tj} itself is a nonlinear function on the stored water s_{tj} for $t\in T$ and $j\in W$. Then, the objective (3.3) may express the maximization of the amount of electricity generated along the planning horizon. The function (3.3) is temporal quasi-separable and spatial separable; see /2/, /4/, /7/ for separable functions and /16/, /24/ for temporal-separable and spatial-nonseparable functions. The intermediate bound R_{tji} may be given by a constant per each pair (j,i) and represents the power generation capacity of the associated turbine. The upper bound u_{tji} may be given by a constant per each pair (j,i) and may be regarded as the maximum physical capacity of the section (j,i) of the river. The intermediate bound T_{tj} may be regarded as a safety bound on the amount of water to be stored in reservoir j for $j\in W$ at the time period t. Now, since the water overflow $r_{tji}-R_{tji}$ cannot be used for hydroelectric power generation and the excess of stored water $s_{tj}-T_{tj}$ must be penalised, the objective can be expressed by (3.6); see in /17/ a different approach.

4. SKELETAL ALGORITHM FOR OBTAINING FEASIBLE-INCREASING SOLUTIONS.

Let d define the stepdirection from feasible solution, say $\bar{X}$ such that the new iterate can be expressed

$$x_k := \bar{x}_k + \underline{\alpha}_k d_k \qquad \forall k \in \bar{B} \cup \bar{S} \tag{4.1}$$

where $\{\underline{\alpha}_k\}$ is the steplength vector (see Section 7). Given eqs. (2.2) and matrix partition (2.5), by linearity it results

$$(B \;\; S \;\; N)\begin{pmatrix} d_B \\ d_S \\ d_N \end{pmatrix} = 0 \tag{4.2}$$

being $d \equiv (d_B^t, d_S^t, d_N^t)^t$. The basic stepdirection d_B is used to satisfy the constraints system (2.2), the nonbasic stepdirection d_N is temporarily fixed to zero, and the superbasic stepdirection d_S is used to maximise the objective function (3.6).

At each iteration, the problem then becomes determining vectors d and $\underline{\alpha}$, such that $\{\underline{\alpha}_k d_k\}$ is feasible and increasing enough; the algorithm must be globally and, if possible, Q-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

$$d = Zd_S \tag{4.3}$$

such that

$$d_{k'} = -\sum_{k \in \bar{S}} \rho_{k'k} d_k \qquad \forall k' \in \bar{B} \tag{4.4}$$

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

$$\max\{h^t d_S + 1/2 d_S^t H d_S\} \tag{4.5}$$

where the reduced gradient h and the reduced Hessian H can be written

$$h \equiv Z^t g = g_S - S^t \mu_B \tag{4.6}$$

$$H \equiv Z^t G Z \tag{4.7}$$

such that the basic estimation μ_B of the constraints Lagrange multipliers solves the system

$$g_B = B^t \mu_B \tag{4.8}$$

and $g = (g_B^t, g_S^t, g_N^t)^t$. Note that the solution of problem (4.5) and, then, the solution d_S of system

$$Hd_S = -h \tag{4.9}$$

is feasible-ascent for a positive definite matrix -H and a maximal basis spanning tree.

Solving the n-system (4.8) when the arcs corresponding to the columns of B form a spanning tree does not need a great computational effort /5/, but the LP simple rules for updating μ_B do not apply when the objective function is nonlinear (even if basic set $\bar{B}$ does not change). From other point of view, using (4.8) in (4.6) could be computationally advantageous, since (1) a-n=n for $|Q_j|=1$ $\forall j \in J$, W=J and then, $|\beta_k|$ for $k \in \bar{S} \cup \bar{N}$ could be small, (2) the cardinality of the set to be used while iteratively solving problem (4.5) could be much smaller than a-n, and (3) β_k must be used, in any case, for obtaining the stepdirection d and the steplength $\underline{\alpha}$. Then, based on the computational effort to be required, we suggest the two following alternative formulas for obtaining h_k $\forall k \in \bar{S}$

$$h_k = \begin{cases} g_k - \sum_{k' \in \beta_k} \rho_{k'k} g_{k'} & \text{for } |\bar{B} \cup \bar{S}| > 0.5 \sum_{k \in \bar{S}} |\beta_k| \\ g_k - \pi_{i_k} + \pi_{j_k} & \text{, otherwise} \end{cases} \tag{4.10a}$$

where $\pi_{jk} = 0$ for j_k being the root node, and the Lagrange multipliers estimation $\mu_B = (\pi_1, \ldots, \pi_n)^t$ is obtained by recursion from (4.8) such that

$$g_{k'} = \pi_{i_{k'}} - \pi_{j_{k'}} \qquad \forall k' \in \bar{B} \tag{4.10b}$$

Matrix H is likely very dense even for sparse matrices Z and G. Since we are dealing with large-scale problems, we cannot afford to use matrix H, nor any of its approximations suggested in the literature. We suggest to use the Tuncated Newton method at independent series of iterations (see Section 5), such that matrix H does not need to be stored and the computer effort 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/.

The steplength $\{\underline{\alpha}_k\}$ must be feasible and $\{\underline{\alpha}_k d_k\}$ must be increasing enough; it is interesting it may allow to activate as many as possible superbasic arcs (see Section 7). Being d ascent, a feasible $\{\underline{\alpha}_k\}$ must be such that $0<\underline{\alpha}_k \le \alpha_k$, where the upper bound α_k keeps feasibility on arc k; α_k is given by the expression

$$\alpha_k = |(ab_k - \bar{X}_k)/d_k| \quad \forall k \in \bar{E} \cup \bar{S} \quad \bar{a}_k \qquad (4.11)$$

where ab_k takes the active bound on the direction of the sign of d_k. Let $\underline{a}_k$, $\bar{a}_k$ and $\bar{\underline{a}}_k$ denote the lower bound (l_{tji} or m_{tj}), 'intermediate' bound (R_{tji} or T_{tj}, see Section 3) and upper bound (u_{tji} or M_{tj}) of the feasible flow in arc k, respectively; the active bound ab_k is obtained as follows

$$\begin{aligned}
&(1)\ \text{If } d_k<0 \wedge \bar{X}_k \le \bar{a}_k,\ ab_k:=\underline{a}_k \\
&(2)\ \text{If } d_k<0 \wedge \bar{X}_k > \bar{a}_k,\ ab_k:=\bar{a}_k \\
&(3)\ \text{If } d_k>0 \wedge \bar{X}_k < \bar{a}_k,\ ab_k:=\bar{a}_k \\
&(4)\ \text{If } d_k>0 \wedge \bar{X}_k \ge \bar{a}_k,\ ab_k:=\bar{\underline{a}}_k
\end{aligned} \qquad (4.12)$$

After obtaining the step $\{\underline{\alpha}_k d_k\}$ at the current iteration (here, termed <u>major iteration</u>), a new iterate is obtained and, theoretically, the algorithm continues till $\|h\| = 0$ or the superbasic set is empty and, then, the de-activating process is executed; the Lagrange multipliers if the solution is optimal or their estimates if the solution is quasi-optimal are used for selecting the nonbasic arc to be de-activated (see Section 6).

<u>Data structures</u>.

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 (the manifold) 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 pivotting and, then, the new (maximal) basis spanning tree may be easily obtained by using LP special data structures without any matrix manipulation.

Our data structures are a mixture of those described in /24/ for the predecessor array but extended to the case for which $\exists j \in J$ with $|Q_j|>1$, and /5/ for the transversal, reverse and depth arrays.

Let define the <u>stem node</u> of a given superbasic arc as the first common node of the paths to the root node of its two ending nodes. Assume that a pivotting must be performed between the bounding (outgoing) basic arc, say k' and the non-bounding (ingoing) superbasic arc, say k such that $k' \in \beta_k$; see in Section 7 (conditions (7.15)-(7.16)) the procedure that we suggest for chosen the arc k. Let define the <u>pivot's path</u> as the path connecting the ending node of arc k and the ending node of arc k', such that it does not include the stem node. The specialization of our procedures for performing the pivotting takes advantage of the structure of the <u>basic neighbourhood</u> of a given node, say $\ell_1 \equiv n(t,j)$, since it is included by the set of nodes
$\{\ell_2\} \subset \{n(t-1,j), n(t+1,j), n(t,i) \ \forall i \in P_j \cup Q_j\}$
for which the arc $a(\ell_1,\ell_2)$ is basic.

The predecessor array is used for obtaining the initial transversal and reverse arrays. The predecesor and transversal arrays are used for obtaining the initial depth array; it is used for obtaining the distances of the endnodes of a given superbasic or nonbasic arc to its stem such that, by using this information and the predecessor array, its BEP is obtained. The depth, transversal and predecessor arrays are used for obtaining the constraints Lagrange multipliers estimation μ_B. The predecessor and transversal arrays are used for updating the depth array. The predecessor, reverse and depth arrays are used for updating the transversal array.

Although the nonzero elements of the columns of matrix ρ can be computed (by using the depth and predecessor arrays) whenever they are needed at each iteration, we suggest to (partially) compute matrix ρ at the beginning of each iteration since the algorithm described in this paper uses the same columns very frequently. Note that it is possible to split the matrix ρ in different (so-termed independent) submatrices such that there is not any link in matrix ρ, nor in the objective function among the related (so-termed independent) sets (see Section 5.2); the optimization of each related manifold may be performed at each independent iteration. Note that it is only required, at each of these iterations, to updating the related subarrays of the four data structures and the related submatrix of matrix ρ.

Let $\bar{S}^{(p)}$ (see below) denote the set of superbasic arcs $\{\ell\}$ related to the independent submatrix $\rho^{(p)}$; let $q(.)$ denote the order on which any superbasic arc (.) is located in the set $\bar{S}^{(p)}$. Let the integer array BEPA be such that BEPA(i) stores for i=1 to nbep a basic arc, say k' such that $\exists k \in \bar{S}^{(p)} | k' \in \beta_k$ where $nbep = \sum_{\ell} |\beta_\ell| \forall \ell \in \bar{S}^{(p)}$. Let the binary array BEPS be such that BEPS(i) gives the orientation of the basic arc BEPA(i) in the BEP of the related superbasic arc. Then the nonzero elements of the column related to any arc $k \in \bar{S}^{(p)}$ in the submatrix $\rho^{(p)}$ are as follows: ± 1 for the basic arcs stored in BEPA(i) for $i = \underline{i}+1$ to $\underline{i}+|\beta_k|$, where $\underline{i} = \sum_{\ell} |\beta_\ell|$ such that $\{\ell\} \subset \bar{S}^{(p)}$ denotes the subset of superbasic arcs with $q(\ell) < q(k)$; the nonzero element will be +1 for BEPS(i)=0 (forward orientation) and -1 for BEPS(i)=1 (reverse orientation). Note that the structures BEPA, BEPS related to the different independent submatrices $\rho^{(p)}$ are not required to be simultaneously allocated; note also that, since the value of nbep may drastically change from one independent set to another, the structures are dynamically allocated whenever they are needed.

5. OBTAINING THE SUPERBASIC STEPDIRECTION.

5.1. BRIEF REVIEW OF THE TRUNCATED NEWTON METHOD.

See in /9/, /11/ the motivation for using the Truncated Newton (TN) methodology when 'solving' sistem (4.9); it is a natural extension of the conjugate gradient method for solving system

$$Hd_S + h = 0 \tag{5.1}$$

At each iteration, say i (here, termed minor iteration) of the conjugate gradient method, a stepdirection $\delta_S^{(i)}$ is obtained as a linear combination of the residual error $e^{(i-1)} = -Hd_S^{(i-1)} - h$ and the stepdirections $\{\delta_S^{(j)}\}$ of previous minor iterations such that they are conjugate.
Let $d_S^{(i)} = d_S^{(i-1)} + \alpha^{(i)} \delta_S^{(i)}$ be the solution (probably, inexact) of system (5.1), where scalar $\alpha^{(i)}$ is the (exact) steplength that solves the quadratic problem

$$\min\{e^{(i-1)^t} \alpha \delta_S^{(i)} - 1/2\, \alpha^2 \delta_S^{(i)^t} H \delta_S^{(i)}\} \tag{5.2}$$

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

$$e^{(i)} = e^{(i-1)} - \alpha^{(i)} H \delta_S^{(i)} \tag{5.3}$$

If $\| e^{(i)} \|$ satisfies the test (5.4), then $d_S \equiv d_S^{(i)}$ is the truncated solution of system (5.1); the tolerance η_ℓ in test (5.4) is dynamically updated at each major iteration ℓ by using expression (5.6).

Skeletal algorithm A1

Assign $e^{(0)} := -h;\ d_S^{(0)} := 0;\ \beta^{(0)} := 0;\ \delta_S^{(0)} := 0;$

Obtain $z^{(0)}$: $Wz^{(0)} = e^{(0)}$

Do i=1 to τ_1

  $\delta_S^{(i)} := -z^{(i-1)} + \beta^{(i-1)} \delta_S^{(i-1)};\ q^{(i)} := H \delta_S^{(i)}$

  If $\delta_S^{(i)^t} q^{(i)} \geq -\varepsilon_1 \| \delta_S^{(i)} \|_2^2$ then

    do;

    If i=1 then $d_S := \delta_S^{(i)}$: else $d_S := d_S^{(i-1)}$

    stop;

    end;

  $\alpha^{(i)} := -e^{(i-1)^t} z^{(i-1)} / \delta_S^{(i)^t} q^{(i)}$

  $d_S^{(i)} := d_S^{(i-1)} + \alpha^{(i)} \delta_S^{(i)}$

  $e^{(i)} := e^{(i-1)} - \alpha^{(i)} q^{(i)}$

  If $\| e^{(i)} \|_{1+t} / \| h \|_{1+t} \leq \eta_\ell$ then do; $d_S := d_S^{(i-1)}$; stop; end;

  Obtain $z^{(i)}$: $Wz^{(i)} = e^{(i)}$

  $\beta^{(i)} := e^{(i)^t} z^{(i)} / e^{(i-1)^t} z^{(i-1)}$

End;

$d_S := d_S^{(\tau_1)}$

Where W is a preconditioning (positive definite) matrix and

$$\| \delta_S^{(i)} \|_2^2 = \| e^{(i-1)} \|_2^2 + \beta^{(i-1)^2} \| \delta_S^{(i-1)} \|_2^2$$

see /11/ and Section 8.

It can be shown /9/ that for $\varepsilon_1>0$ small enough, d_S is an ascent stepdirection, the steplength $\alpha=1$ is ascent enough in the vecinity of the local optimal point $\mathring{X}$ of the current manifold (if -H is positive definite (pd) and $\eta_\ell\to 0$ for $\ell\to\infty$), and the above algorithm is globally convergent; in addition, if $-\mathring{H}$ is pd the rate of convergence on $\{X\}\to\mathring{X}$ is superlinear iff

$$\lim\|e\|/\|h\|\to 0 \quad (\text{i.e.}, \eta_\ell\to 0) \tag{5.5a}$$

such that its order is t+1, where $0<t\leq 1$ iff

$$\lim \|e\|_{1+t}/\|h\|^{1+t}_{1+t} < \infty \tag{5.5b}$$

Thus, tolerance η_ℓ can be written

$$\eta_\ell = \max\{\varepsilon_1, \min\{\eta_0, \gamma\|h\|^t_{t+1}\}\} \tag{5.6}$$

for $0<\eta_0<1$ and $\gamma>0$; for t=1 the rate of convergence is quadratic as the Newton method. If $\|h\|$ is large (X is away from $\mathring{X}$), only few minor iterations are required for obtaining d_S; when X is getting close to $\overset{*}{X}$ then $\|h\|\to 0$ which implies $\eta_\ell\to 0$ and, then d_S is getting close to a Newton stepdirection. Tolerance τ_1 is a safeguard againt unstabilities on calculating $q^{(i)}$. Tolerance ε_1 is also used as other safeguard; it avoids that d_S is not ascent (e.g., if -H is not pd). Typical values, $\varepsilon_1 = \varepsilon_M^{1/2}$ where ε_M is the machine precision in floating point calculations (10E-15 in our case), $\tau_1 = 3|\bar{S}|$, $\gamma = 1$ and $\eta_0 = \min\{0.01, 1/\ell'\}$ where ℓ' is the major iteration of the subproblem defined by the current manifold; ℓ' is reset to 1 whenever set $\bar{S}$ is changed.

5.2. INDEPENDENT SETS OF SUPERBASIC ARCS.

Note that the TN method /9/, /11/ does not require the calculation of any Hessian matrix, but the product

$$q^{(i)} \equiv H\delta_S^{(i)} = Z^t G Z \delta_S^{(i)} \tag{5.7}$$

For obtaining $q^{(i)}$, the superbasic set $\bar{S}$ is partitioned into, say $|P|$ disjoint so-termed <u>independent superbasic sets</u>, such that

$$\bar{S} \triangleq \bigcup_{p\in P} \bar{S}^{(p)} \tag{5.8}$$

$$\bar{S}^{(p)} \cap \bar{S}^{(q)} = \{\emptyset\} \qquad \forall p,q\in P \tag{5.9}$$

For stating the two alternative necessary conditions for the unique valid partition (5.9), let $\bar{B}^{(p)}\subset\bar{B}$ define the set of basic arcs covered by the superbasic arcs included in set $\bar{S}^{(p)}$; that is,

$$\bar{B}^{(p)} \triangleq \bigcup_{k\in\bar{S}^{(p)}} \beta_k \tag{5.10}$$

and let $\bar{C}^{(p)}\triangleq\bar{B}^{(p)} \cup \bar{S}^{(p)}$ define the set of basic and superbasic arcs to be used for obtaining $d_S^{(p)}$.

(i) Superbasic arc k will be included in set $\bar{S}^{(p)}$ if the following condition is satisfied

$$\bar{B}^{(p)} \cap \beta_k \neq \{\emptyset\} \tag{5.11}$$

That is, two superbasic arcs will belong to the same independent superbasic set if any flow change in one of them <u>affects the other's solution feasibility</u>. Note that condition

$$\beta_k \cap \beta_\ell \neq \{\emptyset\} \tag{5.12}$$

is not sufficient, since it could be possible that β_k and β_ℓ are disjoint sets and the following condition is satisfied

$$(\bar{B}^{(p)} \cap \beta_k \neq \{\emptyset\}) \wedge (\bar{B}^{(p)} \cap \beta_\ell \neq \{\emptyset\}) \tag{5.13}$$

(ii) Superbasic arc k will be included in set $\bar{S}^{(p)}$ if the following condition is satisfied

$$\exists G_{gg'} \neq 0 \mid (g\in\{k\}\cup\beta_k) \wedge (g'\in\bar{C}^{(p)}) \tag{5.14}$$

That is, two superbasic arcs, say k and ℓ will belong to the same independent set if any flow change in one of them <u>affects the other's objective function coefficient</u>.

Sets, say $\bar{S}^{(p)}$ and $\bar{S}^{(q)}$ will be joined in one single set if the following condition is satisfied

$$(\bar{B}^{(p)} \cap \bar{B}^{(q)} \neq \{\emptyset\}) \vee (\exists G_{gg'} \neq 0 \mid g\in\bar{C}^{(p)} \wedge g'\in\bar{C}^{(q)}) \tag{5.15}$$

Let $\overline{VL}^{(p)} \subset VL\cap\bar{C}^{(p)}$ denote the set of basic and superbasic linear arcs whose variable-coefficients are not fixed at the current iteration. Let $\overline{NL}^{(p)}\triangleq NL\cap\bar{C}^{(p)}$ denote the set of nonlinear arcs in set $\bar{C}^{(p)}$. Then arc $a(t,j,i)\in VL\cap\bar{C}^{(p)}$ belongs to the set $\overline{VL}^{(p)}$

iff $a(t,j)\in\overline{NL}^{(p)}$ or $a(t+1,j)\in\overline{NL}^{(p)}$, given the special structure (3.4)-(3.5) of the matrix G.

Note that at a given major iteration, $|P|$ 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 sucessive major iterations are as follows.

(1) The Lagrange multiplier estimator π_i is not required to be calculated for those nodes {i} such that, at the given major iteration, $i\neq i_k$, $i\neq j_k$, for $k'\in\bar{B}^{(p)}$.

(2) The computational effort for obtaining vector $q^{(i)}:=Z^t(G(Z\delta_S^{(i)}))$ is drastically reduced.

(3) Faster minor iterations at the price of more (but much cheaper) major iterations. Note that the elements of the matrix G related to the arcs included in the set $\bar{C}^{(p)}/(\overline{VL}^{(p)}\cup\overline{NL}^{(p)})$ are not used for obtaining any stepdirection $d_S^{(p)}$, nor their gradient elements are required to be re-evaluated after obtaining it.

(4) Independent steplength upper bounds for each set $\bar{C}^{(p)}$ $\forall p\in P$.

(5) Strong reduction on the number of arcs (i.e., cardinality of set $\bar{C}^{(p)}$) to be used for obtaining the steplength related to set $\bar{C}^{(p)}$. Note also that only the terms of the objective function (3.6) related to set $\bar{C}^{(p)}$ are to be recomputed for obtaining the objective function value $F(X_{BS}^{(p)})$ related to each trial step.

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

Assume that $q^{(i)}$ is related to the superbasic stepdirection $d_S^{(p)}$.

Let $PL^{(p)}\triangleq\bar{C}^{(p)}\cap PL$, $VL^{(p)}\triangleq\bar{C}^{(p)}\cap VL$ and $\overline{VL}_n^{(p)}\triangleq VL^{(p)}/\overline{VL}^{(p)}$. Let $\bar{C}_n^{(p)}$ denote the complement of the set $\bar{C}^{(p)}$ in set $\bar{B}\cup\bar{S}\cup\bar{N}$

(1) Obtain intermediate vector $\underline{\delta}^{(i)}:=Z^{(p)}\delta_S^{(i)}$ such that

$$\underline{\delta}_\ell^{(i)}:=\begin{cases}-\sum_{k\in\bar{S}^{(p)}}\rho_{\ell k}\delta_k^{(i)} & \forall\ell\in\bar{B}^{(p)}\\ \delta_\ell & \forall\ell\in\bar{S}^{(p)}\\ 0 & \forall\ell\in\bar{C}_n^{(p)}\end{cases}\qquad(5.16)$$

(2) Obtain intermediate vector $\bar{\delta}^{(i)}:=G\underline{\delta}^{(i)}$, such that

$$\bar{\delta}_\ell^{(i)}:=\begin{cases}0 & \forall\ell\in\bar{C}_n^{(p)}\cup PL^{(p)}\cup\overline{VL}_n^{(p)}\\ \sum_{g'\in\overline{NL}^{(p)}}G_{3\ell g'}\underline{\delta}_{g'}^{(i)} & \forall\ell\in\overline{VL}^{(p)}\\ \sum_{g'\in\overline{VL}^{(p)}}G_{3g'\ell}\underline{\delta}_{g'}^{(i)}+\sum_{g'\in\overline{NL}^{(p)}}G_{4\ell g'}\underline{\delta}_{g'}^{(i)} & \forall\ell\in\overline{NL}^{(p)}\end{cases}$$

(5.17)

Computation of vector $\bar{\delta}^{(i)}$ is very fast since G_4 is a symmetric tridiagonal matrix, the elements related to the same pair (t,j) in a given column of matrix G_3 differ only in the coefficient c_{tji} $\forall i\in Q_j$, the elements of $\underline{\delta}^{(i)}$ related to set $PL^{(p)}\cup\overline{VL}_n^{(p)}$ are not used (since the related elements in matrix G_3 are zero), and only the rows of the Hessian matrix related to set $\overline{VL}^{(p)}\cup\overline{NL}^{(p)}$ are used.

(3) Obtain vector $q^{(i)}:=Z^{(p)t}\bar{\delta}^{(i)}=\bar{\delta}_S^{(i)}-(B^{-1}S)^{(p)t}\bar{\delta}_B^{(i)}$

such that $\bar{\delta}^{(i)}=(\bar{\delta}_B^{(i)t},\ \bar{\delta}_S^{(i)t},\ \bar{\delta}_N^{(i)t})^t$.

$$q_k^{(i)}=\bar{\delta}_k^{(i)}-\sum_{k'\in\beta_k}\rho_{k'k}\bar{\delta}_{k'}^{(i)}\qquad\forall k\in\bar{S}^{(p)}$$

(5.18)

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 $X=\bar{X}+\underline{\alpha}d$ is feasible-increasing.

$$t1:\ \|h^{(p)}\|_2\le\varepsilon_2\|X_S^{(p)}\|_2\vee\|X_{BS}^{(p)}-\bar{X}_{BS}^{(p)}\|_2/(1+\|X_{BS}^{(p)}\|_2)\le\varepsilon_3$$

$$\vee\|h^{(p)}\|_\infty\le\varepsilon_6\vee\bar{S}^{(p)}=\{\emptyset\}$$

t2: $|F(X^{(p)}) - F(\bar{X}^{(p)})|/(1+|F(X^{(p)})|) \le \varepsilon_4$
in the last τ_2 iterations
t3: $\|h^{(p)}\|_\infty \le \bar{\varepsilon}_q$

Each manifold has $|P|$ independent problems to be optimized, such that the optimization of each problem p for $p\epsilon P$ is interrupted once the following tests are satisfied for the current solution: t1 (optimal solution) or $\ulcorner t1 \wedge (t2 \vee t3)$ (quasi-optimal solution). Note that an optimal solution is assumed to be found in the current manifold if t1 for all p; it is quasi-optimal if $\exists p\epsilon P|\ulcorner t1$ and $(t2 \vee t3)$ for all $p|\ulcorner t1$. Typical values for the (positive) tolerances are: $\varepsilon_2=\varepsilon_3=\varepsilon_4=$ 10E-04, $\varepsilon_6=0.1$ and $\tau_2=3$. The quasi-optimality tolerance $\bar{\varepsilon}_q$ for a given manifold, say q is obtained as follows.

(1) $\bar{\varepsilon}_q := \max\{\varepsilon_6, \varepsilon_d\bar{\varepsilon}_{q-1}\}$, where $\bar{\varepsilon}_{q-1}=\|h^{(0)}\|_\infty$ for q=1.
(2) If $\bar{\varepsilon}_q \geqslant \|\bar{h}\|_\infty$ then $\bar{\varepsilon}_{q-1} := \|\bar{h}\|_\infty$ and go to (1).

Vector $\bar{h}$ takes the reduced gradient evaluated at the point $\bar{X}_S$, where the related superbasic set $\bar{S}$ includes:

(a) The given superbasic set at the end of the optimization on the previous manifold, and

(b) The just de-activated nonbasic set (see below):

The fraction ε_d is obtained so that the following expression holds

$$\ln\varepsilon_d = (1/(\tau_7+1))\ln(\varepsilon_6/\|h^{(0)}\|_\infty)$$

Typically, τ_7=7 or 5. Note that, at the τ_7-th de-activating process at most, ε_q equals the optimality tolerance ε_6.

In any case, $\bar{\varepsilon}_q$ is reset to ε_6 if the sosution of the previous manifold was optimal. Note that $\varepsilon_q \to 0$ for $q\to\infty$ and, then, the manifold principle holds /6/, /8/.

Let $\bar{U}$ define the set of unsafe arcs; a unsafe arc is a nonbasic arc that was made basic-superbasic after obtaining the optimal solution of any manifold and, again, become nonbasic.

Let us define indicator γ_k $k\epsilon\bar{N}$ 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 $\bar{D}$ 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 $\bar{D}\Delta\cup\bar{D}^{(p)}$ $\forall p\epsilon P$ and $\bar{D}^{(p)}\cap\bar{D}^{(q)}=\{\emptyset\}$, where $\bar{D}^{(p)}$ is the independent nonbasic set to be de-activated and, then, joined with the independent superbasic set $\bar{S}^{(p)}$. A candidate nonbasic arc will not be de-activated if $|\bar{D}|$ is at its (upper) bound and there is, at least, any other candidate arc with higher (first-order) guarantee of 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 then $\gamma_k=0$ $\forall k\epsilon\bar{U}$; the set $\bar{U}$ is declared empty if the solution is optimal.

Note that the basic arc k', such that $k'\notin\beta_\ell$ $\forall\ell\epsilon\bar{S}$ and $k'\epsilon\beta_k$ for any nonbasic arc k from the set $\bar{N}/\bar{U}$, may have its value $\bar{X}_{k'}$ at its intermediate bound $\bar{a}_{k'}$. Then, it results that the evaluation of its gradient element depends on the sign γ_k of the de-activating direction of the nonbasic arc k. Hence, the constraints Lagrange multipliers estimation μ_B cannot be used for obtaining the nonbasic Lagrange multipliers estimation λ. See formula (4.10) and let

$$\lambda_k^i = g_k^i - \sum_{k'\epsilon\beta_k} \rho_{k'k}\, g_{k'}^j, \qquad \forall k\epsilon\bar{N}/\bar{U} \qquad (6.1)$$

where λ_k^i is the Lagrange multiplier estimation related to the i-direction of the potential move of nonbasic arc k, and g_k^i (res. $g_{k'}^j$) gives the gradient element related to the i-direction of nonbasic arc k (res. the j-direction of basic arc k'). The directions can be + (up), - (down) and 0 (no move).

Note that $g_k^i = g_k^+$ for $\bar{X}_k = \underline{a}_k$ (lower bound) and $g_k^i = g_k^-$ for $\bar{X}_k = \bar{\underline{a}}_k$ (upper bound) such that $g_k^i = g_k$ where g_k is the usual gradient element related to arc k. Both gradient elements g_k^+ and g_k^- (and, then λ_k^+ and λ_k^-) are required for the nonbasic arc whose current solution is at its 'intermediate' bound $\bar{a}_k$. Note also that $g_{k'}^+ = g_{k'}^- = g_{k'}$, for $k' \in \beta_k$ such that $\bar{X}_{k'} \neq \bar{a}_{k'}$.

Gradient elements g_ℓ^+ and g_ℓ^- for nonbasic arc, say ℓ such that $\bar{X}_\ell = \bar{a}_\ell \wedge \gamma_\ell \neq 0$, and for basic arc, say ℓ such that $\bar{X}_\ell = \bar{a}_\ell$ are obtained as follows: $g_\ell^+ = 0$ and $g_\ell^- = K_{tj} c_{tji}$ for $\ell \equiv a(t,j,i)$; $g_\ell^+ = g_\ell$ for $\ell \equiv a(t,j)$ such that $\max\{0, s_{tj} - T_{tj}\} := s_{tj} - T_{tj}$ while obtaining, in the usual way, gradient element g_ℓ, and $g_\ell^- = g_\ell^+ + P_{tj}$; see (3.6). Element $g_{k'}^j$, for $k' \in \beta_k$ and $\bar{X}_{k'} = \bar{a}_{k'}$ is expressed as follows:

$$g_{k'}^j := \begin{cases} g_{k'}^+, & (g_k^i = g_k^+ \wedge \rho_{k'k} = -1) \vee (g_k^i = g_k^- \wedge \rho_{k'k} = +1) \\ g_{k'}^-, & \text{otherwise} \end{cases} \qquad (6.2)$$

Finally, indicator γ_k for nonbasic arc k such that $k \in \bar{N}/\bar{U}$ is assigned as follows. For $\bar{X}_k = \underline{a}_k$, $\gamma_k = +$ if $\lambda_k^+ > \varepsilon_7$; otherwise, $\gamma_k = 0$ where ε_7 is a positive tolerance (typically 0.1). For $\bar{X}_k = \bar{\underline{a}}_k$, $\gamma_k = -$ if $\lambda_k^- < -\varepsilon_7$; otherwise, $\gamma_k = 0$. For $\bar{X}_k = \bar{a}_k$, it results

$$\gamma_k := \begin{cases} 0 \mid \lambda_k^+ \leq \varepsilon_7 \wedge \lambda_k^- \geq -\varepsilon_7 \\ + \mid \lambda_k^+ > \varepsilon_7 \wedge \lambda_k^- \geq -\varepsilon_7 \\ - \mid \lambda_k^+ \leq \varepsilon_7 \wedge \lambda_k^- < -\varepsilon_7 \\ j \text{ such that } |\lambda_k^j| = \max\{|\lambda_k^+|, |\lambda_k^-|\}, \\ \quad \text{otherwise} \end{cases} \qquad (6.3)$$

Thus assign $\lambda_k := \lambda_k^i$ for $i = \gamma_k \neq 0$.

Note that expression (6.2) is based on the direction i and orientation $\rho_{k'k}$. The ambiguity on $g_{k'}^j$, for $k' \in \beta_k \cap \beta_\ell | \bar{X}_{k'} = \bar{a}_{k'}$ is solved by blocking the nonbasic arc, k or ℓ that satisfies test t5 (see below); note that there is not a null step for $\lceil$t5 since λ is used as the stepdirection of set $\bar{D}$ (see Section 6.6), the solution of a superbasic arc is not, by definition, at any of its bounds, and a maximal basis spanning tree is assumed (i.e., $\nexists k' \in \bar{B}^{(p)}$ $\forall p \in P$ such that $X_{k'}$ is at any of its bounds).

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

$$t4: \|h\|_\infty \leq \varepsilon_8 |\lambda_k|$$

where ε_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 $\lceil$t4.

6.3. BLOCKING NONBASIC ARCS.

A maximal basis spanning tree avoids degenerate basic-superbasic pivots, but it does not prevent null steps when a nonbasic arc is de-activated. Therefore, a mecanism 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 γ_k is set to 0 if otherwise a null step could not be prevented. Thus, $\gamma_k = 0$ if t5$\vee$t6$\vee$t7, where t5, t6 and t7 are the result of the following blocking tests:

$$t5: |\lambda_k| = \min\{|\lambda_k|, |\lambda_\ell| \mid \bar{X}_{k'} = \bar{a}_{k'}, \text{ for } k' \in \beta_k \cap \beta_\ell \wedge \ell \in \bar{N} \mid \gamma_\ell \neq 0 \wedge \gamma_k \gamma_\ell \rho_{k'k} \rho_{k'\ell} = -\}$$

Note that $\lceil$t5 if the flow in basic arc k' changes in the same direction for any flow change in the appropriate direction of arcs k and ℓ (given by γ_k and γ_ℓ, respectively).

t6 (case $\gamma_k = +$): $\exists k' \in \beta_k$ such that
$\rho_{k'k} = -1$ (reverse) $\wedge (\bar{X}_{k'} = \bar{\underline{a}}_{k'})$ or
$\rho_{k'k} = +1$ (forward) $\wedge (\bar{X}_{k'} = \underline{a}_{k'})$

t7 (case $\gamma_k = -$): $\exists k' \in \beta_k$ such that
$\rho_{k'k} = -1 \wedge (\bar{X}_{k'} = \underline{a}_{k'})$ or
$\rho_{k'k} = +1 \wedge (\bar{X}_{k'} = \bar{\underline{a}}_{k'})$

If t5$\vee$t6$\vee$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 $\bar{D}$ 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 up a given bound, say $\min\{\tau_3, \varepsilon_9 |\bar{N}/\bar{U}|\}$, where τ_3 and ε_9 are positive tolerances (typically, $\tau_3 = 60$

and $\varepsilon_9=0.5$). For reducing the computer effort and storage required by the Truncated-Newton method, it is interesting that the cardinality of any independent set $\bar{S}^{(p)} \cup \bar{D}^{(p)}$ does not exceed a given bound, say τ_4 (typically, 60). Note that the set $\bar{U}$ is declared empty if the current solution is 'optimal' (i.e., test t1 is satisfied).

Given the dimensions of our problem, we suggest to use a strategy for partial pricing if $|\bar{N}/\bar{U}| \geq \varepsilon_{10}(a-n)$, where ε_{10} is a positive tolerance (typically, 0.1), such that only a subset of $\bar{N}/\bar{U}$ is priced at each de-activating process. We suggest to pricing the arcs sequentially in set $\bar{N}/\bar{U}$, so that they will be candidate to be de-activated if the pricing result is favorable (see Section 6.2) and the blocking tests are not satisfied (see Section 6.3). Once $|\bar{D}|$ reaches the allowed bound, the next candidate scanned arc will replace the arc from set $\bar{D}$ with the worst pricing result till $\underline{r}$ is not greater than a given bound, say ε_{11} (typically, 0.1); $\underline{r}$ gives the ratio of the number of replacements to the number of candidate scanned arcs. When $\underline{r} \leq \varepsilon_{11}$ the scanning is interrupted; it will be restarted, at the next de-actiating process, by pricing the arc where it was left out.

Let the following de-activating tests:

t8 : $|\bar{D}| < \min\{\tau_3, \varepsilon_9 |\bar{N}/\bar{U}|\}$

t9 : $\lceil t8 \wedge (|\lambda_k| > \min\{|\lambda_\ell| \ \forall \ell \in \bar{D}\})$

t10: $|\bar{N}/\bar{U}| \geq \varepsilon_{10}(a-n)$

t11: $\underline{r} \leq \varepsilon_{11}$

Formally, $\bar{D} \underline{\Delta} \bar{D} \cup \{k\}$ for $\gamma_k \neq 0$ if t8 $\vee$ t9.
After the de-activating process, the unsafe set $\bar{U}$ is updated such that $\bar{U} \underline{\Delta} \bar{U} \cup \bar{D}$. If the current solution is 'optimal' in the given manifold and $\bar{D}=\{\emptyset\}$, stop since it is assumed that the optimal solution of the problem has been found.

Finding the most suitable values for the tolerances is a subject for experimentation, mainly for the multiple de-activating tolerance τ_3. Note that the stategies described in Sections 6.2, 6.3 and 6.4 produce the sufficient-long relaxing step required in /8/ for global convergence.

6.5. OBTAINING INDEPENDENT SET $\bar{D}^{(p)}$ TO BE DE-ACTIVATED.

Recall that $\bar{D} \underline{\Delta} \cup \bar{D}^{(p)} \ \forall p \in P$ and $\bar{D}^{(p)} \cap \bar{D}^{(q)} = \{\emptyset\}$. Let $\bar{C}_d^{(p)} \underline{\Delta} \bar{B}_d^{(p)} \cup \bar{D}^{(p)}$ where $\bar{B}_d^{(p)} \underline{\Delta} \cup \beta_k \ \forall k \in \bar{D}^{(p)}$. An arc, say k to be de-activated must be included in set $\bar{D}^{(p)}$ if any move $d_k \neq 0$ affects the solution feasibility or the objective function coefficient of any arc from set $\bar{C}^{(p)} \cup \bar{C}_d^{(p)}$; formally, $\bar{D}^{(p)} \underline{\Delta} \bar{D}^{(p)} \cup \{k\}$ for $k \in \bar{D}$ if t12 $\wedge$ (t13 $\wedge$ t14), where t12, t13, and t14 are the result of the following including tests:

t12: $|\bar{S}^{(p)}| + |\bar{D}^{(p)}| < \tau_4$

t13: $(\bar{B}^{(p)} \cup \bar{B}_d^{(p)}) \cap \beta_k \neq \{\emptyset\}$

t14: $\exists G_{gg'} \neq 0 | (g \in \{k\} \cup \beta_k) \wedge (g' \in \bar{C}^{(p)} \cup \bar{C}_d^{(p)})$

It is suggested to perform the testing in the following sequence: t12, t13, t14.

Now, to assure that sets $\bar{D}^{(j)}$ $\forall j \in P$ are independent, it is required to analyse if any move $d_k \neq 0$ $k \in \bar{D}^{(p)}$ would affect the solution feasibility or the objective function coefficient of any arc from the q-th set $\forall q \in P/\{p\}$; in that case, both sets $\bar{D}^{(p)}$ and $\bar{D}^{(q)}$ must be joined. Formally, $\bar{D}^{(p)} \underline{\Delta} \bar{D}^{(p)} \cup \bar{D}^{(q)} \cup \{k\}$ and $\bar{D}^{(q)} \underline{\Delta} \{\emptyset\}$ if arc k simultaneously satisfies t13 $\vee$ t14 for the p-th and q-th current independent sets and, besides, the following joining test is satisfied.

t15: $$\sum_{i \in \{p,q\}} |\bar{S}^{(i)}| + |\bar{D}^{(i)}| < \tau_4$$

If $\lceil$t15 then arc k is not de-activated and, then, $\bar{D} \underline{\Delta} \bar{D}/\{k\}$; if as a result, $\bar{D}=\{\emptyset\}$, then the set $\bar{C}$ must be revisited and partitioned in as many as possible independent sets and, as a final solution, the tolerance τ_4 must be temporarily incremented to a suitable value. Note that the procedure is executed during the de-activating process and, then, hopefully after many basic-superbasic arcs have been activated on the (sub)optimization of the previous manifold.

6.6. OBTAINING THE SUPERBASIC STEPDIRECTION AFTER DE-ACTIVATING. RELAXING STEP /8/.

The new ascent independent stepdirection $\bar{d}_S^{(p)} = (\underline{d}_S^{(p)t}, \underline{\underline{d}}_S^{(p)t})^t$, where $\underline{d}_S^{(p)}$ takes the direction related to the old superbasic set

$\bar{S}^{(p)}$ and $\underline{\underline{d}}_S^{(p)}$ is related to set $\bar{D}^{(p)}$ is obtained as follows.

$$\underline{d}_S^{(p)} = h^{(p)}$$
$$\underline{\underline{d}}_S^{(p)} = \{h_k \; k \in \bar{D}^{(p)}\}$$

where $h^{(p)}$ takes the reduced gradient related to set $\bar{S}^{(p)}$, and $h_k \equiv \lambda_k$. Note that a null step is avoided, since (a) $\exists k \in \bar{C}^{(p)}$ such that X_k is at any of its bounds, provided that the set $\bar{B}^{(p)}$ forms a maximal basis spanning tree, and (b) the flow change in set $\bar{B}_d^{(p)}$ has the appropriate direction (see Sections 6.2 and 6.3).

Although the relaxing step may not produce a strong increase in the objective function value, it is very cheap and the case for which $\| h^{(p)} \|$ is not small, if any is not very frequent.

7. SUPERBASIC MULTI-ACTIVATING LINESEARCH.

Barring exceptional circunstances, at most one basic-superbasic arc per each set $\bar{C}^{(p)}$ $\forall p \in P$ can be added to nonbasic set $\bar{N}$ in the algorithms based on manifold suboptimization and active set strategies (see e.g. /13/, /21/); so if say, 500 basic-superbasic arcs at the initial feasible solution will be active at the optimal solution, then the method would require at least 500 major iterations to converge. A better performance could be achieved by the following alternative approach.

Assume that the independent superbasic stepdirection $d_S^{(p)}$ has been obtained as follows: $d_k := ab_k - \bar{X}_k \;\; \forall k \in I^{(p)}$, and d_k is a Truncated-Newton direction for $\forall k \in I_n^{(p)}$ where $I_n^{(p)} \underline{\Delta} \bar{S}^{(p)} / I^{(p)}$. Set $I^{(p)}$ denotes the set of quasi-active arcs in superbasic set $\bar{S}^{(p)}$, such that it can be expressed

$$I^{(p)} \underline{\Delta} \{ k \in \bar{S}^{(p)} | \quad |ab_k - \bar{X}_k| \leq \varepsilon^{(p)} \} \tag{7.1}$$

where scalar $\varepsilon^{(p)}$ is given by

$$\varepsilon^{(p)} = \min \{ \varepsilon_{12}, \| \bar{X}_S^{(p)} - [\bar{X}_S^{(p)} + h^{(p)}]^{\#} \|_2 \} \tag{7.2}$$

where

$$[\bar{X}_S + h]_k^{\#} := \begin{cases} \min \{ (\bar{X}_S + h)_k, ab_k \} \text{ if } h_k > \varepsilon_1 \\ \bar{X}_S \text{ if } |h_k| \leq \varepsilon_1 \\ \max \{ (\bar{X}_S + h)_k, ab_k \} \text{ if } h_k < -\varepsilon_1 \end{cases} \tag{7.3}$$

for $\varepsilon_{12} > 0$ (typically, 0.01). By a slight abuse of the definition given in Section 4 (see expresion (4.12)), ab_k in expressions (7.1) and (7.3) takes the active bound in the direction of the sign of h_k.

Let

$$\alpha_S^{(p)} = \min \{ \alpha_k \; \forall k \in I_n^{(p)} \} \tag{7.4}$$
$$\alpha_B^{(p)} = \min \{ \alpha_k \; \forall k \in \bar{B}^{(p)} \} \tag{7.5}$$

denote the upper bounds on (scalar) steplength $\alpha^{(p)}$ for keeping feasibility on sets $\bar{S}^{(p)}$ and $\bar{B}^{(p)}$, respectively; α_k is the upper bound related to ark k as given by expression (4.11), where d_B is temporarily given by using expression (4.12) and, then, it takes the active bound on the direction of the sign of d_k. If $\alpha_B^{(p)} \leq \alpha_S^{(p)}$, generally, only one arc can be activated in set $\bar{C}^{(p)}$. But, if $\alpha_B^{(p)} > \alpha_S^{(p)}$, fewer major iterations could be required by allowing more than one superbasic arc from set $\bar{S}^{(p)}$ to be activated at each major iteration; given the special structure of matrix $\rho^{(p)}$ (submatrix of ρ related to set $\bar{C}^{(p)}$), the computer effort is likely to be within affordable limits.

Let the feasible solution $X_{BS}^{(p)}$ be expressed as follows.

$$X_S^{(p)} = \bar{X}_S^{(p)} + [\alpha^{(p)}]^{\# t} d_S^{(p)} \tag{7.6}$$

where

$$[\alpha^{(p)}]_k^{\#} := \begin{cases} 1 \text{ for } k \in I^{(p)} \\ \min \{ \alpha^{(p)}, \alpha_k \} \text{ for } k \in I_n^{(p)} \end{cases} \tag{7.7}$$

and

$$X_B^{(p)} = \bar{X}_B^{(p)} + d_B^{(p)} \tag{7.8}$$

where, now

$$d_B^{(p)} = \rho^{(p)} [\alpha^{(p)}]^{\# t} d_S^{(p)} \tag{7.9}$$

such that

$$\bar{\alpha}_B^{(p)} \geq 1 \tag{7.10}$$

where

$$\bar{\alpha}_B^{(p)} = \min \{ \alpha_{k'}, \; \forall k' \in \bar{B}^{(p)} | |d_{k'}| > \varepsilon_1 \} \tag{7.11}$$

The upper bound $\alpha_{k'}$ is obtained as in (4.11) by using the expression (4.12) for $ab_{k'}$ and considering that $\{d_{k'}\}$ is given by expression (7.9).

By assuming that the set $\bar{B}^{(p)}$ forms a maximal basis spanning tree, and following the same approach described in /3/, /6/, /10/, it can be shown that, under mild conditions, the solution $X_{BS}^{(p)}$ (7.6)-(7.9) is feasible and increasing enough, and the convergence $\{X_{BS}^{(p)}\} \to \hat{X}_{BS}^{(p)}$ is global with a Q-superlinear rate, provided that

$$\alpha^{(p)} := \beta^m \quad (7.12)$$

The scalar m is the first nonnegative integer 0,1,2,... that keeps feasible the basic solution $X_B^{(p)}$ and satisfies the condition

$$F(X_{BS}^{(p)}) - F(\bar{X}_{BS}^{(p)}) \geq \mu \left(\sum_{k \in I^{(p)}} h_k (X_k - \bar{X}_k) + \beta^m \sum_{k \in I_n^{(p)}} h_k d_k \right) \quad (7.13)$$

where $\beta \epsilon \{0;1\}$, $\mu \epsilon \{0;0.5\}$; typically, $\beta = 0.5$ and $\mu = 0.1$

If the point $X_B^{(p)}$ does not satisfy the condition (7.10), then the scalar m in (7.12) is updated such that

$$m = \ell n (\bar{\alpha}_B^{(p)} \times \alpha^{(p)}) / \ell n \beta \quad (7.14)$$

and the condition (7.13) is tested again.

Let $\gamma^{(p)} = \min\{1, \alpha_S^{(p)}\}$ so that if, at any major iteration, a fixed number τ_5 of trial steplengths fail to keep feasible the point $X_B^{(p)}$ or does not satisfy the Armijo-like condition (7.13), then $\gamma^{(p)}$ is used as the next trial value provided that $\gamma^{(p)} < \beta^m$. It is assumed that the steplength procedure has failed if $\alpha^{(p)} \leq \varepsilon_1$ or a fixed number τ_6 of trial steplengths does not produce a feasible solution that satisfies the condition (7.13). Typically, $\tau_5 = 2$ or 3, and $\tau_6 = 6$ or 7.

If $\overline{VL}^{(p)} \cup \overline{NP}^{(p)} = \{\emptyset\}$ (see Section 5.2), a LP-network flow subproblem is to be maximized. In that case, the condition (7.13) is always satisfied by the feasible solution $X_{BS}^{(p)}$ (7.6)-(7.9).

Note that the basic arc $k' \epsilon \bar{B}^{(p)}$ may belong to more than one set β_k $k \epsilon \bar{S}^{(p)}$. Assume that k' is the bounding basic arc; then, a superbasic arc, say k is selected for pivotting such that $k' \epsilon \beta_k | \alpha_k > \alpha^{(p)}$. As a result, $|\beta_\ell^{(n)}| \geq |\beta_\ell^{(o)}| \quad \forall \ell \epsilon \bar{S}^{(p)}, \quad \ell \neq k$, where (o) identifies an old BEP and (n) identifies a BEP after pivotting. The entering arc is chosen such that

$$|\beta_k^{(o)}| = \min\{|\beta_\ell^{(o)}| \quad \forall \ell \epsilon L^{(p)}\} \quad (7.15)$$

where the set $L^{(p)}$ is included by the arcs that satisfy the condition

$$|X_\ell - ab_\ell| > \varepsilon_{13} \max\{|X_\ell - ab_\ell|\} \quad (7.16)$$

for $\ell \epsilon \bar{S}^{(p)} | \alpha_\ell > \alpha^{(p)} \wedge k' \epsilon \beta_\ell^{(o)}$; ab_ℓ is given by expression (4.12) and ε_{13} is a given nonnegative tolerance (typically, 0.1). Since the linesearch procedure tends to reduce the cardinality of the superbasic set, finally it results that, hopefully, $|\bar{S}^{(p)}|$ is reduced by more than one and the value of $|\beta_\ell^{(n)}|$ still could be affordable.

8. COMPUTATIONAL EXPERIENCE.

An experimental prototype for solving the nonlinear replicated network described in Sections 1 to 3 has been written, based on the ideas presented in Sections 4 to 7. This section reports some computational results on some real-life problems coming from the hydroelectric power field. The prototipe, named NLRNET, was written in PL/I, compiled with the option OPT(2) and run on an IBM 370/158 computer operating under VM/CMS.

The problems are the following.

Problem I

$|J| = 7$, $W = J$,

$P_1 = \{\emptyset\}$, $P_2 = \{1\}$, $P_3 = \{\emptyset\}$, $P_4 = \{2,3\}$, $P_5 = \{4\}$, $P_6 = \{5\}$, $P_7 = \{4\}$, $P_8 = \{5,6,7\}$. $|T| = 4$.

Note that $Q_4 = \{5,7\}$ and $Q_5 = \{6,8\}$. It is a segment of Problem III.

Problem II

Its physical description is very simple. $|J| = 6$, $W = J$, $P_1 = \{\emptyset\}$, $P_j = \{j-1\}$ for $j = 2,\ldots,7$. (Note that Exit is given by the dummy reservoir $j = |J|+1$). Then, $Q_j = \{j+1\}$ $\forall j \epsilon J$. That is, there is not any run-of-river reservoir, each reservoir in J has only one directly downstream reservoir and each reservoir in J/{1} has only one directly upstream reservoir. Planning horizon: $|T| = 26$ bi-weeks.

Problem III

Same Problem I, but $|T|=26$.

Problem IV

$|J|=8$, $W=J/\{3\}$,

$P_1=\{\emptyset\}$, $P_2=\{\emptyset\}$, $P_3=\{1,2\}$, $P_4=\{3\}$, $P_5=\{3\}$,

$P_6=\{4\}$, $P_7=\{\emptyset\}$, $P_8=\{5,6,7\}$,

$P_9=\{8\}$. $|T|=26$.

Problem V

$|J|=4$, $W=J$,

$P_1=\{\emptyset\}$, $P_j=\{j-1\}$ for $2,\ldots,5$. $|T|=52$.

Problem VI

$|J|=6$, $W=\{2,3\}$,

$P_1=\{\emptyset\}$, $P_j=\{j-1\}$ for $i=2,\ldots.7$. $|T|=52$.

Problem VII

$|J|=6$, $W=J$,

$P_1=\{\emptyset\}$, $P_2=\{\emptyset\}$, $P_3=\{1,2\}$, $P_j=\{j-1\}$ for $j=4,5,6,7$. $|T|=52$.

Problem VIII

$|J|=6$, $W=J/\{6\}$,

$P_1=\{\emptyset\}$, $P_2=\{1\}$, $P_3=\{\emptyset\}$, $P_4=\{\emptyset\}$, $P_5=\{2\}$,

$P_6=\{3,4,5\}$, $P_7=\{6\}$. $|T|=52$.

Problem IX

$|J|=6$, $W=\{1,2\}$,

$P_1=\{\emptyset\}$, $P_2=\{\emptyset\}$, $P_3=\{1,2\}$. $P_4=\{3\}$, $P_5=\{2,4\}$,

$P_6=\{5\}$, $P_7=\{6\}$. $|T|=52$.

Problem X

$|J|=9$, $W=\{1,2,5,7\}$,

$P_1=\{\emptyset\}$, $P_2=\{1\}$, $P_3=\{2\}$, $P_4=\{3\}$,

$P_5=\{\emptyset\}$, $P_6=\{5\}$, $P_7=\{6\}$, $P_8=\{4,7\}$,

$P_9=\{8\}$, $P_{10}=\{9\}$. $|T|=52$.

Problem XI

$|J|=10$, $W=J/\{5,8,10\}$,

$P_1=\{\emptyset\}$, $P_2=\{1\}$. $P_3=\{\emptyset\}$, $P_4=\{2\}$,

$P_5=\{3,4\}$, $P_6=\{\emptyset\}$, $P_7=\{6\}$, $P_8=\{7\}$,

$P_9=\{5,8\}$, $P_{10}=\{9\}$, $P_{11}=\{10\}$. $|T|=52$.

Problem XII

$|J|=11$, $W=\{1,2,3\}$,

$P_1=\{\emptyset\}$, $P_2=\{\emptyset\}$, $P_3=\{1,2\}$, $P_4=\{\emptyset\}$,

$P_5=\{3,4\}$, $P_6=\{5\}$, $P_7=\{\emptyset\}$, $P_8=\{\emptyset\}$,

$P_9=\{6,7,8\}$, $P_{10}=\{\emptyset\}$, $P_{11}=\{9,10\}$,

$P_{12}=\{11\}$. $|T|=52$.

Problem XIII

$|J|=15$, $W=\{4,5,6\}$,

$P_1=\{\emptyset\}$, $P_2=\{1\}$, $P_3=\{2\}$, $P_4=\{\emptyset\}$,

$P_5=\{4\}$, $P_6=\{\emptyset\}$, $P_7=\{6\}$, $P_8=\{7\}$,

$P_9=\{8\}$, $P_{10}=\{2,3\}$, $P_{11}=\{10\}$,

$P_{12}=\{5,9,10\}$, $P_{13}=\{11,12\}$, $P_{14}=\{13\}$,

$P_{15}=\{14\}$, $P_{16}=\{15\}$. $|T|=52$.

Note that 'reservoir' $j=10$ is a splitter such that it does not generate electricity, being $Q_{10}=\{11,12\}$.

Problem XIV

$|J|=16$, $W=J$,

$P_1=\{\emptyset\}$, $P_2=\{1\}$, $P_3=\{2\}$, $P_4=\{\emptyset\}$, $P_5=\{\emptyset\}$,

$P_6=\{\emptyset\}$, $P_7=\{4,5\}$, $P_8=\{\emptyset\}$, $P_9=\{8\}$,

$P_{10}=\{3,6,7,9\}$, $P_{11}=\{\emptyset\}$, $P_{12}=\{11\}$,

$P_{13}=\{\emptyset\}$, $P_{14}=\{10,12,13\}$, $P_{15}=\{14\}$,

$P_{16}=\{15\}$, $P_{17}=\{16\}$. $|T|=52$.

Problem XV

$|J|=23$, $W=\{1,2,5,6,7,8,9,10,15,17,18,23\}$,

$P_1=\{\emptyset\}$, $P_2=\{1\}$, $P_3=\{2\}$, $P_4=\{2\}$, $P_5=\{3\}$,

$P_6=\{4,5\}$, $P_7=\{\emptyset\}$. $P_8=\{\emptyset\}$, $P_9=\{\emptyset\}$,

$P_{10}=\{7,8,9\}$. $P_j=\{j-1\}$ for $j=11,\ldots,14$, $P_{15}=\{6,14\}$, $P_j=\{j-1\}$ for $j=16,\ldots,24$. $|T|=52$. Note that $Q_2=\{3,4\}$.

For comparative purposes, the methods used for obtaining the superbasic stepdirection are as follows.

(i) Preconditioned Reduced Truncated Newton (PRTN) method as described in Section 5, algorithm A1 where the absolute value of the diagonal of the reduced Hessian matrix $(Z^tG_dZ)^{(p)}$ is used as the preconditioning (positive definite) matrix W for the p-th independent set, being G_d the diagonal of the Hessian matrix G (3.4). Note that the diagonal element in $(Z^tG_dZ)^{(p)}$ related to superbasic arc $k\epsilon\bar{S}^{(p)}$ is given by the expression

$$G_{kk} + \sum_{k'\epsilon\beta_k} G_{k'k'} \quad (8.1)$$

such that $G_{\ell\ell}=0$ for $\ell\equiv a(t,j,i), j\epsilon J$; i.e., ℓ is a non-linking arc and, then only matrix G_4 is used; see (3.4)-(3.5). Remark: the sign of the vector $q^{(i)}$ at a given minor iteration of the Truncated Newton method is changed if it has been assumed 'a priori' (and, in our case, it is an easy task) that matrix G_4 is positive definite.

(ii) Reduced Truncated Newton (RTN) method as described in Section 5, algorithm A1 for W=I.

(iii) Preconditioned Reduced Gradient (PRG) method, such that $d_S^{(p)}$ solves the system

$$Wd_S^{(p)}=h^{(p)} \quad (8.2)$$

where the matrix $(Z^tG_dZ)^{(p)}$ is used as the preconditioning matrix W. Note that the matrix $(Z^tG_dZ^{(p)}$ has only nonzero elements whenever the BEP's of two superbasic arcs, say k and ℓ intersect nonlinearly; i.e., they have one basic arc in common, at least (and, then $\beta_k\cap\beta_\ell\neq\{\emptyset\}$) and, besides $\nexists k'\epsilon\beta_k\cap\beta_\ell$ such that $k'\equiv a(t,j)$ for $j\epsilon W$. Then, the element in matrix $(Z^tG_dZ)^{(p)}$ related to superbasic arcs $k,\ell\epsilon\bar{S}^{(p)}$ is given by the expression

$$\sum_{k'\epsilon\beta_{\ell k}} \rho_{k'k}\rho_{k'\ell}G_{4_{k'k'}} \quad (8.3)$$

where $\beta_{\ell k} \triangleq \{k'\epsilon\beta_k\cap\beta_\ell | k'\equiv a(t,j),\ j\epsilon W\}$

Since matrix W must be positive-definite the sign of its elements is changed if it has been assumed 'a priori' that matrix G is negative-definite. System -- (8.2) is solved by a 2-step procedure that uses the Cholesky factor R of matrix W. If, while obtaining the factor R, the intermediate computation $\underline{R}_{kk}$ is such that $\underline{R}_{kk}\le\epsilon_1^2$ then $R_{kk}=\epsilon_1$ and, otherwise $R_{kk}=\underline{R}_{kk}^{1/2}$ where R_{kk} is a diagonal element of R; see in /22/ a different approach.

(iv) Reduced Gradient (RG) method, such that $d_S^{(p)}:=h^{(p)}$. Note also that $d_k:=h_k$ for $k\epsilon\bar{S}^{(p)}$ in methods PRTN, RTN and PRG provided that $\nexists\ell\epsilon\{k\}\cup\beta_k | \ell\equiv a(t,j), j\epsilon W$.

In all cases, the prototype starts with an all-artificial basis and terminates when the tests described in Section 6 are satisfied. The values of the tolerances are as follows: ϵ_1=sqrt(10E-15), $\epsilon_2=\epsilon_3=\epsilon_4$=10E-04, $\epsilon_6=\epsilon_7=0.1$, $\epsilon_8=0.9$, $\epsilon_9=1.0$, $\mu=\epsilon_{10}=\epsilon_{11}=\epsilon_{13}=0.1$, $\eta_0=\epsilon_{12}=0.01$, $\beta=0.5$, and $\tau_1=\tau_2=\tau_5=3$, $\tau_3=a-n$, $\tau_4=60$, $\tau_6=7$ and $\tau_7=5$.

Tables 1 and 2 contain a summary of the results of our computational experience on the problems described above.

The column headings of Table 1 mean the following:

mani. Number of manifolds that have been optimized; note that it is also the number of deactivating processes. Note also that the results are obtained by using the strategy that allows to deactivate as many non-blocked nonbasic arcs as possible, provided that the pricing is favorable and the upper bound for the number of superbasic arcs per each independent set is not violated.

ind. Number of independent superbasic sets.

msize. Maximum cardinality of the independent superbasic sets.

Mitn. Number of major iterations.

mitn. Number of minor iterations

nfval. Number of (partial) objective function evaluations. It helps to measure the performance of the linesearch parameters.

npiv. Number of pivottings. It is interesting to note the high number of pivottings that is required for all problems.

Table 1. Summary of the results for the test problems. Method PRTN.

Problem	#nodes	#arcs	mani	ind	msize	Mitn	mitn	nfval	npiv	nacti	ffeval (%)	fgeval (%)	fheval (%)	CPU time (secs)
PI	28	64	5	4	20	35	108	35	21	41	64	38	26	6
PII	156	312	10	22	37	76	326	81	36	85	48	22	29	51
PIII	182	416	6	7	59	115	419	116	64	149	43	27	26	84
PIV	208	416	6	15	21	57	201	59	42	63	14	16	15	123
PV	208	416	5	19	12	52	344	72	39	106	8	12	19	104
PVI	312	416	4	8	31	48	143	50	12	21	38	29	36	115
PVII	312	624	5	15	37	65	342	72	32	98	12	9	15	290
PVIII	312	572	6	17	29	55	321	63	15	109	8	6	21	103
PIX	312	416	5	16	18	31	47	36	11	25	24	23	29	96
PX	468	676	7	21	43	72	348	73	53	104	36	31	34	190
PXI	520	884	6	30	38	153	761	162	69	277	18	11	23	259
PXII	572	728	4	24	31	62	197	71	43	55	62	52	59	194
PXIII	780	988	5	16	14	41	79	45	22	31	31	35	38	365
PXIV	832	1664	8	32	43	310	709	389	169	532	20	16	22	559
PXV	1196	1872	9	31	57	233	782	233	153	302	47	25	29	657

Table 2. CPU-time (secs) comparison of the methods for the test problems

Problem	PRTN	RTN	PRG	RG
PI	6	23	11	192
PII	51	430	524	4833
PIII	84	201	196	>5000
PIV	123	904	5634	12631
PV	104	792	6089	6429
PVI	115	271	225	3824
PVII	290	1792	1542	9037
PVIII	103	642	2321	15940
PIX	96	401	634	>10000
PX	190	682	327	7622
PXI	259	543	3584	8094
PXII	194	609	804	17914
PXIII	365	1482	1302	>20000
PXIV	559	3645	--	--
PXV	657	2032	--	--

nacti. Number of activated (basic, superbasic) arcs. Note that nacti-npiv measures very accurately the performance of the bending, backtracking linesearch procedure.

ffeval. Average percent per iteration of the number of evaluations of the nonlinear terms in the objective function.

fgeval. Average percent per iteration of the number of evaluations of the non-constant elements in the gradient.

fheval. Average percent per iteration of the number of evaluations of the nonzero elements in the Hessian.

The most remarkable result shown in Table 1 is the performance of the concept of independent sets; the average size of the independent superbasic sets is relatively small compared with the cardinality of the related superbasic set. As a result, the computation of each major iteration is very cheap. The traditional criterion (number of evaluations of the objective function, gradient and Hessian) for the performance of an algorithm does not help too-much for analysing the performance of the concept of independent superbasic sets; by definition, only a fraction of the objective function, gradient and Hessian elements is to be evaluated at each major iteration. We prefer to use the parameters ffeval,fgeval and fheval.

The multiple superbasic activating strategy seems very promising, but more extensive experimentation is required to draw any conclusive remark; a computational comparison with the traditional single activating linesearch described in /13/ is planned by using test cases for which $S^I >> \overset{*}{S}$, where S^I and $\overset{*}{S}$ are the superbasic sets at the initial feasible and optimal solutions, respectively.

Surprisingly, the number of candidate arcs for being deactivated is not too-high and, then, we could not perform an extensive experimentation with the joining test t15. Although the overlapping of different BEP's is very frequent, blocking test t5 has not been fully experimented; this is due to the fact that it was very rare that a basic arc takes its intermediate value while, at the same iteration, all superbasic arcs whose BEP's include the basic arc also take a bounding value. Since a maximal basis spanning tree was built for the initial feasible solution and a blocking nonbasic arcs strategy was used, no null steps were found.

The results shown in Table 2 for the RTN method were included to show the necessity of preconditioning in an practical algorithm. By comparing the results for RTN and PRTN (the preconditioned algorithm), it is seen that RTN is 2 to 8 times as expensive to use as the preconditioned algorithm.

The results shown in Table 2 for the RG method show that the systematic using of the reduced gradient as the superbasic stepdirection have a poor convergence, if any; the strategy must be rejected for large-scale problems.

The PRG method is the exact Newton method for separable objective functions; since our case is quasi-separable then the good performance of the method in some problems is not a surprise. In any case, the computation of the Cholesky factor of the preconditioning matrix W is cheap; note that the cardinality of the independent superbasic sets is small and, probably, the matrix W is not very dense.

9. CONCLUSIONS.

In this paper we have presented a rough algorithm that takes into account second-order information for solving a type of large nonlinear network problems; its main ideas may be easily extended to the general sparse case.

Taking advantage of the structure of the objective function and constraints of the nonlinear replicated network; the main features of the algorithm are as follows. The predecessor, depth, tranverse and reverse arrays, together with the sequential storing of the BEP's related to each independent superbasic set, are the only data structures to be used for dealing with the tree. A new procedure for pricing nonbasic arcs in the presence of discontinuities in the objective function has been described. Null steps are prevented, since the basis spanning tree is kept maximal and an ad-hoc blocking de-activating strategy is used. A multiple and anti-zigzagging de-activating strategy with partial pricing is used. The new concept of independent superbasic sets is introduced so that the Truncated-Newton method and the linesearch procedure can be used for optimizing 'in parallel' the manifold of each independent basic-superbasic set. Given the special structure of the Hessian matrix G and the variable-reduction matrix Z, the computer effort for obtaining the vector $q^{(i)} \equiv Z^t(G(Z\delta_S^{(i)}))$, at each minor iteration i, is within affordable limits; note that the cardinality of each independent set is usually small. One of the main reasons (apart the nonlinearity of the objective function) for not using necessarily the estimation of the Lagrange multipliers of the nodes while evaluating the reduced gradient, is precisely the size of the basic equivalent path of the superbasic arcs. The main advantage that the multiple superbasic activating linesearch offers over the traditional single activating approach is that as many as $|\bar{S}|$ new arcs may become active in a single major iteration; however, the computational comparison between both linesearch strategies in large-scale problems is a subject for future experimentation; and so it is the procedure for selecting candidate nonbasic arcs, so that it would promote small independent sets of arcs to be de-activated.

Selecting subsets of candidate nonbasic arcs to be de-activated may be performed in several ways, so that the size of the set $\bar{S}^{(p)} \cup \bar{D}^{(p)}$ is kept small for all p; a precedure is described in /24/ for a temporal separable and spatial nonseparable objective function. In our case, the best procedure is left open at this point.

In Section 8 we have reported the results of the algorithm by using a set of real-life hydroelectric power generation problems; based on this computational experience, it seems that the ideas described in this paper are worthy of extensive experimentation. Computer effort is important because the full model with 25 reservoirs and 52 time periods is to be run for planning purposes under several assumed inflow patterns; however, in some cases, aggregating the last, say 14 weeks of the time horizon in 3 time periods (i.e., months) does not strongly deteriorate the planning goal.

10. ACKNOWLEDGEMENT.

I wish to thank Ron Dembo for various guiding comments and suggestions that have strongly improved the algorithm, Dick Rosenthal for making available his code to the author, and Bob Meyer for his useful suggestions on the subject.

11. REFERENCES.

/1/ P. BECK, L. LASDON and M. ENGQUIST,: "A reduced gradient algorithm for non-linear network problems". ACM Transactions on Mathematical Software 9 , 57-70 (1983).

/2/ K. BELLING-SEIB,: "An hybrid reduced Newton algorithm for nonlinear network problems", NETFLOWS83, Pisa (Italy). (1983).

/3/ D.P. BERTSEKAS,: "Projected Newton methods for optimization problems with simple constraints", SIAM, J. of Control and Optimization 20 , 221-226 (1982).

/4/ D.P. BERTSEKAS and E.M. GAFNI,: "Projected Newton methods and optimization of multicommodity flows", IEEE trans. on Automatic Control 28 , 1090-1096 (1983).

/5/ G.H. BRADLEY, G.G. BROWN and G.W. GRAVES: "Design and Implementation of large-scale primal transshipment algorithms", Management Science 24 , 1-24 (1977).

/6/ R.S. DEMBO,: "A primal Truncated Newton algorithm with application to large-scale nonlinear network optimization", NETFLOW 83, Pisa (Italy). See also: "A bending backtracking linesearch for linearly constrained optimization", Yale School of Organization and Management, working paper series B74, Yale University, Illinois, (1984).

/7/ R.S. DEMBO and J.G. KLINCEWICZ,: "A scaled reduced gradient algorithm for network flow problems with convex separable costs", Mathematical Programming 15, 125-147 (1981).

/8/ R.S. DEMBO and S. SAHI,: "A convergent framework for constrained nonlinear optimization", Yale School of Organization and Management, working paper series B69, Yale University, Illinois, (1983).

/9/ R.S. DEMBO and T. STEIHAUG,: "Truncated Newton algorithms for large-scale unconstrained optimization", School of Organization and Management, working paper, Yale University, Illinois (1980). See also Mathematical Programming 26, 190-212 (1983).

/10/ R.S. DEMBO and H. TOWITZKI,: "On the minimization of quadratic functions subject to box constraints",Yale School of Organization and Management, working paper series B71, Yale University, Illinois (1983).

/11/ L.F. ESCUDERO,: "On diagonally-preconditioning the Truncated Newton method for super-scale linearly constrained nonlinear programming", European J. of Operational Research 17, 401-414 (1984).

/12/ L.F. ESCUDERO,: "A motivation for using the Truncated Newton approach in a very large-scale nonlinear network problem", Mathematical Programming Studies (in press).

/13/ P.E. GILL and W. MURRAY,: "Safeguard steplength algorithms for optimization using descent methods", National Physical Laboratory, report NAC 37, Treddington (UK) (1974).

/14/ F. GLOVER, D. KARNEY and D. KLINGMAN,: "Implementation and computational comparisons of primal, dual and primal-dual computer codes for minimum cost network flow problems", Network 4, 191-212 (1974).

/15/ 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).

/16/ N. HANSCOM, V.H. NGUYEN and J.J. STRODIOT: "A nondifferentiable network programming algorithm and its application to the hydro generation scheduling", Mathematical Programming (to appear).

/17/ Y. IMURA and G. GROSS,: "Efficient large-scale hydro system scheduling with forced spill conditions", Proceedings of the IEEE-PES Winter meeting, N.Y. (IEEE Catalog 84 WM-006-3) (1984).

/18/ D. KARNEY and D. KLINGMAN,: "Implementation and computational study on an in-core; out-of-core primal network code", Operations Research 24, 1056-1077 (1976)

/19/ C.L. MONNA and M. SEGAL,: "A primal algorithm for finding minimum-cost flows in capacitated networks with applications". The Bell System Journal 61, 949-968 (1982)

/20/ R.R. MEYER,: "Two segment separable programming", Management Science 15T,385-395 (1979).

/21/ B. MURTAGH and M. SAUNDERS,: "Large-scale linearly constrained optimization", Mathematical Programming 14, 41-72 (1978).

/22/ S.G. NASH,: "Newton type minimization via the Lanczos method", SIAM J. of Numerical Analysis 21, 770-788 (1984).

/23/ R.E. ROSENTHAL,: "The status of optimization models for the operation of multireservoir systems with stochastic inflows and nonseparable benefits", Tennessee Water Resources Center, report 75, The university of Tennessee, Knoxville, (1980).

/24/ R.E.ROSENTHAL,: "A nonlinear network flow algorithm for maximization of benefit in a hydroelectric power system", Operations Research 29, 763-786 (1981).

/25/ P. WOLFE,: "Methods for linear constraints" in J. Abadie (ed.), Nonlinear programming (North-Holland, Amsterdam, 99-131) (1967).

/26/ M.Z. YAKIN,: "Deterministic and stochastic water reservoir management models", College of Business Administration, working paper CBA1982-98, University of Houston, Houston (Texas) (1982).