Numerical Methods in Engineering '92
Ch. Hirsch et al. (Editors)
1992 Elsevier Science Publishers B.V.

NONLINEAR NETWORK FLOWS WITH SIDE CONSTRAINTS APPLIED TO SHORT TERM HYDROTHERMAL COORDINATION OF ELECTRICITY GENERATION

F. J. Heredia and Narcís Nabona

Statistics and Operations Research Dept., Universitat Politècnica de Catalunya. Campus Sud – FIB , Pau Gargallo 5 08071-Barcelona , Spain

Abstract : An specialized algorithm for solving the nonlinear network flow problem with linear side constraints has been developed and applied to solve the Short Term Hydrothermal Coordination problem. This paper covers a basic description of the algorithm and the problem, the computational results corresponding to a set of reservoir systems for hydroelectric generation located in Spain, and a first analysis of the computational solutions.

Key words : Hydrothermal Coordination, Nonlinear Network Flow Algorithms, Nonlinear Optimization, Side Constraints.

1. INTRODUCTION.

The Short Term Hydrothermal Coordination of Electricity Generation problem (HTC problem) deals with the study of a reservoir system for hydroelectric generation that must supply certain amounts of electric energy at time intervals of a given short period of time for wich water inflows in reservoirs and thermal plant availability is known and considered deterministic. Due to the special structure of HTC problem two different approaches have been used traditionally to solve it : dynamic programming and network flow methods.

The use of dynamic programming (ref. 1) is justified by the multi–stage modellization of the temporary evolution of the resevoir system. The evolution in time of this system is modelled by dividing the whole period of time into a certain number n of intervals each one corresponding to a stage of dynamic programming. The advantages of these methods are the ability to deal with any noncontinuous objective functions and contraints and the simplicity of the algorithm. Its main drawbacks are :

- An initial feasible point must be provided by the user.
- The variables are discretized.
- Constraints over several intervals can not be considered.
- The optimality condition is tied to the coarseness of the discretization of the variables.
- The Lagrange multipliers of the constraints are not known at the solution. (The value of the Lagrange multipliers of certain constraints at the optimal solution has an useful meaning in the economical understanding of the results).
- Important computation time and dimensionality problems.

The network flow approach is based on the fact that the structure of the reservoir system and the temporary evolution of each reservoir can be suitably modelled as a network. Nonlinear network flow algorithms (NNF) overcome most of the drawbacks presented by dynamic progamming :

- NNF algorithms are able to find feasible solutions.
- The state of reservoirs is described through real variables.
- The optimal solution found by NNF satisfies first order necessary conditions and, consequently, the value of the Lagrange multipliers is known.

Many researchers have focused their attention on the NNF algorithms. The usual way to tackle the problem is to combine a data structure of the type proposed by Bradley et al (ref. 2) for the linear network flow problem with the variant of the active set method with superbasic variables introduced by Murtagh & Saunders in (ref. 3). Two exemples of the use of this strategy can be found in Dembo (ref. 4) and Toint & Tuyttens (ref. 5). The aplication of NNF algorithms to the HTC problem has been reported by several authors (Rosenthal (ref. 6), Escudero (ref. 7)). Approaches to the inclusion of side constraints not based on basis partitioning techniques has also been proposed (Lyra and Tavares (ref. 8), Brännlund et al (ref. 9)).

2. THE MODEL

2.1. Description

The reservoir system at interval i can be viewed as a directed graph $\mathcal{G}_i$ with a node associated to each reservoir and arcs that represent water inflows and discharges. The connexion of graphs $\mathcal{G}_i$, $i = 1, \ldots, n$ by arcs of initial/final volumes provides a directed graph known as the *replicated network* .

The goal of HTC is the minimization of the sum of costs of thermal generation at each interval by the best policy of hydrogenerations over the period studied. Hydrogeneration is a nonlinear function of initial and final volume and water discharges. Usually the HTC solution must satisfy additional constraints, such as hydrogeneration limitations and/or irrigation constraints.

2.2. Objective function

There are may possible objective functions in a HTC problem. The objective function employed minimizes four groups of terms. Each group of terms consists in a sum of variables over all intervals of the period studied.

The first group are the thermal generation savings due to hydrogeneration, with negative sign. This is expressed as the product of total hydrogeneration of the interval by a negative cost constant (one for each interval).

The second group are approximate power loss values for each hydro-unit generation (at each interval). The losses of each unit are modelled with just a linear plus a quadratic term of power generation, without cross terms.

The third group are the economic premiums for having a hydro spinning reserve S -difference between maximum capacity and actual hydro power generation- at each interval (with negative sign). The economic premium P is proportional to the total spinning reserve, but only up to a certain amount $\bar{P}$. This truncated linear function is approximated by a continuous and differentiable exponential function of the type $\bar{P}(1 - e^{-KS})$.

Finally, there are penalty terms to avoid having flows at a spillage arc while flow of final volume of the corresponding reservoir is not at maximum capacity. These penalty terms are usually minimized just after phase I and are zero thereafter.

2.3. Hydrogeneration calculation.

In a variable-head reservoir, hydrogeneration H during an interval can be expressed as the product

$$H = \rho \times h_e \times d \qquad (1)$$

where ρ is the efficiency of the mechanical to electrical energy conversion, h_e is the equivalent water head and d is the discharge. Should H be considered as energy generated during the interval (in MWh), d would be the volume of water discharged (in m^3). In case H is considered an average power throughout the interval (in MW), d is then a flow (in m^3/s). Water head h is related to network variables through a function that gives reservoir head for stored volume v, which is a network variable. This function is most usually a polynomial whose coefficients have been adjusted beforehand to fit the reservoir shape. In the work reported this has been done with a third degree polynomial

$$h = c_b + c_l v + c_q v^2 + c_c v^3 \qquad (2)$$

(A fourth degree polynomial has also been used). Since the available network variables are, for a given interval i, the discharge d_i and the initial and final stored volumes v_{i-1} and v_i, the equivalent head h_e must be calculated from them. This can be made taking

$$h_e(v_i - v_{i-1}) = \int_{v_{i-1}}^{v_i} (c_b + c_l v + c_q v^2 + c_c v^3) \mathrm{d}v \qquad (3)$$

which leads to

$$h_e = c_b + \frac{c_l}{2}(v_{i-1} + v_i) + \frac{c_q}{3}(v_i - v_{i-1})^2 + c_q v_{i-1} v_i + \frac{c_c}{4}({v_{i-1}}^2 + {v_i}^2)(v_{i-1} + v_i) \qquad (4)$$

which corresponds to the head of the center of gravity of the water slice between volumes v_{i-1} and v_i in the reservoir. The efficiency ρ in hydrogeneration, changes with water head and discharge flow. It has been modelled as a quadratic function

$$\rho = \rho_b + \rho_{hl} h_e + \rho_{dl} d + \rho_{hd} h_e d + \rho_{hq} {h_e}^2 + \rho_{dq} d^2 \qquad (5)$$

Thus H is a very high order polynomial function of d_i, v_{i-1} and v_i. Assuming that there are r reservoirs, the total hydrogeneration H_i over interval i would be

$$H_i = \sum_{k=1}^{r} H_{ik} \qquad (6)$$

where H_{ik} stands for the hydrogeneration of reservoir k over interval i.

2.4. Hydrogeneration side constraints

There can be limitations to hydrogeneration, at all or some of the intervals, due to load and minimal —or maximal— thermal power generation. These limits take the form

$$H_i \leq H_{i\,\max} \quad i \in \bar{I} \qquad (7)$$

$\bar{I}$ being the set of intervals where hydrogeneration limits are placed.

These constraints are thus nonlinear. A possible simplification, which leads to acceptable results, considers fixed values γ_{ik} different for each reservoir k and interval i instead of the products $\rho \times h_e$. This yields the side constraints linear in d_{ik}

$$H_i \simeq \sum_{k=1}^{r} \gamma_{ik} d_{ik} \leq H_{i\,\max} \quad i \in \bar{I} \qquad (8)$$

2.5. Mathematical formulation

The HTC problem can be formulated as a nonlinear network flow problem with linear side constraints (NNS problem). The mathematical expression of NNS problem is :

$$\min \quad f(x) \tag{9}$$
$$\text{subj. to :} \quad Ax = r \tag{10}$$
$$Tx \leq b \tag{11}$$
$$0 \leq x \leq u \tag{12}$$

with :

(9) $f\colon \mathbb{R}^n \to \mathbb{R}$. $f(x)$ is nonlinear and twice continuously differenciable on the feasible set defined by the constraints (10) to (12). The variables $x \in \mathbb{R}^n$ represent the values of the flows (water) on the arcs of the replicated network .

(10) *Network Equations* : express the flow conservation at the nodes of the replicated network . Matrix $A \in \mathbb{R}^{m \times n}$ is the *node-arc incidence matrix* and $r \in \mathbb{R}^m$ is the supply/demand vector.

(11) *side constrains* : a set of t linear constraints like (8) that hydrogeneration discharge flows must satisfy ($T \in \mathbb{R}^{t \times n}$ and $b \in \mathbb{R}^t$).

(12) $u \in \mathbb{R}^n$ are the upper bounds imposed to the flows on each arc.

3. THE ALGORITHM

The algorithm proposed is an extension of the nonlinear network flow algorithms without side constrains (ref. 4, 5) where side constraints are treated by the methods developed by Kennington & Helgason (ref 10) for the management of the basic matrix in the linear version of NNS problem . An extensive description of this algorithm were presented in (ref. 11) and (ref. 12). Our algorithm follows the general framework of the active set methods with superbasic variables (Murtagh & Saunders (ref. 3)) and has the following main features.

3.1. Initial feasible solution

Problem NNS is solved with a two phase method. An initial feasible solution to NNS problem is found in two steps. The first one, called "phase 0", finds a "pseudo-feasible" solution that consists of a feasible solution to the NNS problem without side constraints. This pseudo-feasible solution is feasible for network constraints but, in general, will violate some of the side constraints. The minimization of this infeasibilities ("phase I") is carried out through an implementation of the Kennington & Helgason algorithm for linear network flow problem with side constrains .

3.2. Basic matrix

The basis of the active set constraints $\bar{B}$ has the following internal partition :

$$\bar{B} = \begin{bmatrix} B & C \\ D & F \end{bmatrix}$$

where B is the matrix corresponding to a rooted spanning tree of the graf $\mathcal{G}$, D are the side constraints coefficients of the arcs of the spanning tree, columns in C contain either node–arc incidence information or are void, while F contains side constraint coefficient of the arcs in C, or slack variable coefficients. The basic variable set includes a subset consisting of basic arcs that form a rooted spanning tree of the network graph, stored as usual via the predecessor, depth, thread and reverse thread vectors. The basic matrix is stored via the spanning tree vectors and through a reduced matrix $Q = F - DB^{-1}C$, called the *working basis*, whose order is the number of side constraints. This is all the information needed for the operations with the inverse of the basic matrix. The update of the inverse of the basic matrix is performed by updating the spanning tree and the inverse of the working basis Q. Numerical implementation details can be found in (ref. 11) and (ref. 12).

3.3. Optimization on the null space

A descent direction on the null space p_z is obtained solving the system

$$H_z p_z = -g_z \tag{13}$$

where g_z is the reduced gradient and H_z is the reduced hessian. Two different techniques to solve this system have been implemented : a truncated Newton method and a quasi–Newton method. The truncated–Newton method follows the strategy exposed by Dembo & Steihaug in (ref. 13). It is based on the solution of system (13) by a *conjugated gradient* (CG) method. The quasi–Newton method follows the methodology exposed by Murtagh & Saunders in (ref. 3). The Cholesky factors R of an approximation to the reduced Hessian ($R'R \approx H_z$) must be stored, updated and retriangularized whenever a change in the matrix H_z occurs. The algorithm makes use of the *quasi–active bounds* strategy (Toint & Tuyttens, (ref. 5))for finding the search direction p. In this strategy the algorithm to calculate the descent direction p acts only over the components of p associated with superbasic variables without quasi–active bounds, and takes the reduced gradient direction for the other components.

3.4. Linesearch

The linesearch strategy follows the method of Bertsekas (ref. 14) in order to reach more than one superbasic variable bound with the same descent direction. If such an step is not possible, the iterated point is found via cubic fit with backtracking.

4. COMPUTATIONAL RESULTS.

The algorithm described in the previous section has been coded in FORTRAN, producing the program called NOXCB. This program has been used to solve a set of real HTC problems that represent several reservoirs systems located in Spain.

4.1. HTC test problems.

The size of the test models is shown in table 1. The general characteristics of the reservoir systems optimized with NOXCB are :

1.- The dimension of the replicated network goes from small to large size.
2.- The number of side constraints ranges from 10% to 66% of the total number of equality constraints.
3.- The side constrains are relatively simple and sparse. They may consist either in the sum of certain arcs in certain time intervals or they can be hydropower limits (8) at some or all intervals. In case of hydropower limits the side constraint has nonzero elements only in the arcs of discharge with hydrogeneration, of the interval to which the side constraint refers.
4.- The objective function is higly nonlinear and has a very costly computational evaluation.

Table 1 : Reservoir system and replicated network dimension of test problems. ISC and HSC stand for *irrigation side constraints* and *hydrogeneration side constraints*

NAME	RESERVOIR SYS.			REP. NETWORK				
	reser.	arcs	inter.	nodes	arcs	# ISC	# HSC	% SC
EBRE01	3	12	26	79	390	26	–	33%
EBRE75	3	12	26	79	390	26	1	34%
EBRE78	3	12	26	79	390	26	3	36%
EBRE77	3	12	26	79	390	26	7	42%
EBRE76	3	12	26	79	390	26	13	49%
EBRE71	3	12	26	79	390	26	26	66%
SUMA01	20	69	36	721	3204	108	–	15%
SUMA75	20	69	36	721	3204	54	17	10%

4.2. Computational experiments.

Problem NNS can be solved either with the specialised code NOXCB or with a general purpose constrained nonlinear optimization package . Tests have been carried out in these two directions. Both methodologies require quite long subsidiary programs to prepare data from reservoir characteristics, natural inflows, thermal generation costs and time period and interval specifications. These programs have also been developed.

The general purpose code choosen is the well known MINOS package (ref. 15), an excellent general purpose code specially for nonlinear problems with linear constraints. The computer used is a SUN Sparc 2 Station.

4.2.1. Initial feasible solution.

NOXCB implements the phase 0/1 strategy described in section 3. This implementation is based on the LEXA and FXCB packages (ref. 16, 17). A first execution of the LEXA subroutines, wich solves a linear network flow problem, finds a pseudo–feasible solution (phase 0). From this point, FXCB subroutines proceed to eliminate the side constraint infeasibilities (phase 1). The results obtained are given in the table 2.

Table 2 : Initial feasible point with NOXCB. *# SCI* and *sum SCI* are the number and sum of the side constraints infeasibilities at the pseudo–feasible solution. If # SCI =0, the pseudo–feasible solution is feasible.

	iter. Ph.0	sec. Ph.0	# SCI	sum SCI	iter. Ph.1	sec. Ph.1
EBRE01	511	0.68	0	0.	–	–
EBRE75	511	0.68	0	0.	–	–
EBRE78	511	0.68	0	0.	–	–
EBRE77	511	0.68	3	5.7×10^6	37	0.92
EBRE76	511	0.68	4	6.2×10^6	39	0.66
EBRE71	511	0.68	7	8.5×10^6	61	0.97
SUMA75	3681	8.10	2	1.5×10^7	172	11.26

4.2.2. Optimal solution.

Once a feasible solution has been found, NOXCB begins the phase 2 procedure. Table 3 summarizes the results achieved for the test problems. The items shown in this table are :

OFE : number of objective function evaluations.
$f(x^0)$: objective function at x^0, the initial feasible point.
$f(x^*)$: objective function value at optimal point x^*.
ASC0 : number of non–key arcs at x^0. It is equivalent to the number of active side constraints.
ASC* : Number of active side constraints at x^*.
SBV : Number of superbasic variables at x^*.
$\|g_z^*\|_1$: 1–norm of the reduced gradient at x^*.
σ^*_{min} : This is the value of the Lagrange multiplier tolerance used in the pricing routines. In this routines, a non basic variable x_q is considered to

become superbasic only if its Lagrange multiplier σ_q satisfies $|\sigma_q| > \sigma_{min}$.

$||\pi^*||_1$: The 1–norm of the Lagrange multipliers of the network equations.

4.3. Performance evaluation of NOXCB

In order to evaluate the efficiency of the especialized code NOXCB against a general purpose code, the MINOS package has been employed to solve the same HTC problems. The performance evaluation has been focused on the following items :

1.- Computation of the initial feasible point x^0.
2.- Computation of x^* starting from different x^0.
3.- Computation of x^* starting from the same x^0 point.

The results of the numerical experiments presented in this paper show that, in general, the current version of NOXCB provides lower total and partial execution times than MINOS. However, the degree of efficiency depends strongly on the structure of the test problem and on the initial feasible point. In order to make realistic comparisons, the user parameters shared by both codes have been fixed to the same values.

4.3.1. Computation of the initial feasible point.

The results shown in Table 4 seem to indicate that the efficiency of both codes in finding an initial feasible point is similar, though there is a clear trend towards a better performance of NOXCB as the problem size increases or when there are more side constraints.

Table 4 : Phase 0/1 of NOXCB versus phase 1 of MINOS.

	iterations		seconds	
	NO	MI	NO	MI
EBRE01	511	33	0.68	0.46
EBRE75	511	33	0.68	0.46
EBRE78	511	109	0.68	1.53
EBRE77	548	123	1.6	1.49
EBRE76	550	192	1.34	2.49
EBRE71	572	311	1.65	4.10
SUMA75	3853	2501	19.36	157.70

4.3.2. Computation of x^*, from different x^0

Table 5 and Figure 1 summarize the results obtained with MINOS and NOXCB when each program finds the initial feasible point in its own way. There are two cases corresponding to problems EBRE76 and SUMA75 in figure 1. Besides the plot of objective function value versus time (A), there are also three other graphics for each case: the number of objective function evaluations (B), the cumulative computation time per iteration (excluding the objective function evaluations) (C) and the number of active side constraints (D), w.r.t. number of iterations. The total computation time is very much related to the number of function evaluations, because the proportion of computation time per iteration devoted to objective function evaluations is high. The philosophy of NOXCB iterations, though quite similar to that of MINOS, seems better suited to Hydrothermal Coordination problems as the number of function evaluations required are in most of the cases less than with MINOS. (The same line search routine is used by both codes). The lower total computation times of NOXCB bear a direct relation to this fact and to the improved convergence. On the other hand the computation time per iteration, once function evaluations have been excluded, is not always lower with NOXCB than with MINOS. This should not be surprising since some operations of NOXCB, even being partly related to network equations, are quite intrincated as compared to MINOS operations, e.g.: the Lagrange multipliers' calculation requires for NOXCB the solution of two network basis sytems of linear equations plus one solution with the Q equation system, whereas with MINOS it requires a single solution using a very sparse eta file. Time per iteration is related to problem size and to the number of active side constraints. For problems of low size like that of EBRE76, iteration time (with function evaluation excluded) is lower than with MINOS. For bigger problems as SUMA75 iteration times (with function evaluation excluded) gets to be, at some iterations, a bit higher with NOXCB than with MINOS, and is related to the number of of active side constraints, as shown in the graphics where the change of slope in the cumulative time per iteration coincides for NOXCB with the change in number of active side constraints. A detailed analysis of this feature is out of the scope of this paper.

4.3.3. Computation of x^* from the same x^0

The behaviour of the nonlinear algorithms used to solve HTC problems depends strongly on the initial feasible point. Therefore, the ability to compare the behaviour of both codes relies on the possibility of introducing iterated feasible points of one of the codes into the other. This is cumbersome since it means transforming the coding of the basic and superbasic variables and that of the basis, from the system used by one program into that of the other one.

Experiments comparing the behaviour of both algorithm starting from the initial feasible solution found by MINOS have been developed for some of the test HTC problems.

Figure 2. shows the graphics of objective function value versus computation time of MINOS and NOXCB starting from the same initial feasible point (computed with MINOS). The results illustrated correspond to problems EBRE01 and SUMA01 and show that NOXCB converges faster than MINOS. The solutions reached in both cases coincide with those obtained with NOXCB when using its own phase 0 and 1 procedures.

Table 3 : Phase 2 of NOXCB.

	sec.	iter.	OFE	$f(x^0)$	$f(x^*)$	ASC0	ASC*	# SBV	$\|\|g_z^*\|\|_1$	σ_{min}^*	$\|\|\pi^*\|\|_1$
EBRE01	12.085	210	254	-2.572×10^4	-5.059×10^4	0	0	0	0.	1.0×10^{-2}	6.1×10^{-1}
EBRE75	18.236	265	384	-2.572×10^4	-5.059×10^4	0	0	2	1.4×10^{-3}	5.1×10^{-4}	6.1×10^{-1}
EBRE78	17.807	230	335	-2.572×10^4	-5.057×10^4	0	1	2	4.0×10^{-3}	1.2×10^{-2}	6.1×10^{-1}
EBRE77	13.349	207	309	3.616×10^7	-3.870×10^4	2	4	2	4.0×10^{-3}	1.3×10^{-2}	4.2×10^{-1}
EBRE76	24.913	376	521	6.030×10^8	-3.554×10^4	3	7	12	9.2×10^{-6}	1.1×10^{-7}	3.8×10^{-1}
EBRE71	31.964	317	636	5.142×10^8	-2.348×10^4	8	15	24	6.6×10^{-3}	6.2×10^{-4}	2.3×10^{-1}
SUMA75	1880.2	2942	6041	6.947×10^8	-2.343×10^5	5	4	29	1.5	0.1	8.2×10^5

Table 5 : Phase 2 of NOXCB versus phase 2 of MINOS.

	iterations		OFE		OFE/iter.		# SBV		ASC*		$f(x^*)$		seconds		
	NO	MI	NO	MI	NO	MI	NO	MI	NO	MI	NO	MI	NO	MI	ratio
EBRE01	210	315	254	1321	1.2	4.2	0	0	0	0	-5.059×10^4	-5.059×10^4	12.08	43.30	0.28
EBRE75	225	315	384	1319	1.7	4.2	2	0	0	0	-5.059×10^4	-5.059×10^4	18.23	44.50	0.41
EBRE78	230	263	335	982	1.4	3.7	2	3	1	1	-5.057×10^4	-5.056×10^4	17.81	33.59	0.53
EBRE77	207	246	309	921	1.5	3.7	2	3	4	4	-3.870×10^4	-3.887×10^4	13.35	32.01	0.42
EBRE76	376	385	521	1110	1.4	2.9	12	27	7	8	-3.554×10^4	-3.540×10^4	24.91	38.86	0.64
EBRE71	317	313	636	649	2.0	2.1	24	28	15	17	-2.348×10^4	-2.337×10^4	31.96	24.22	1.32
SUMA75	2952	12376	6041	23590	2.0	1.9	29	57	4	7	-2.343×10^5	-2.270×10^5	1880.	6474.	0.29

Figure 2 : Objective function value versus execution time starting from the same initial feasible point.

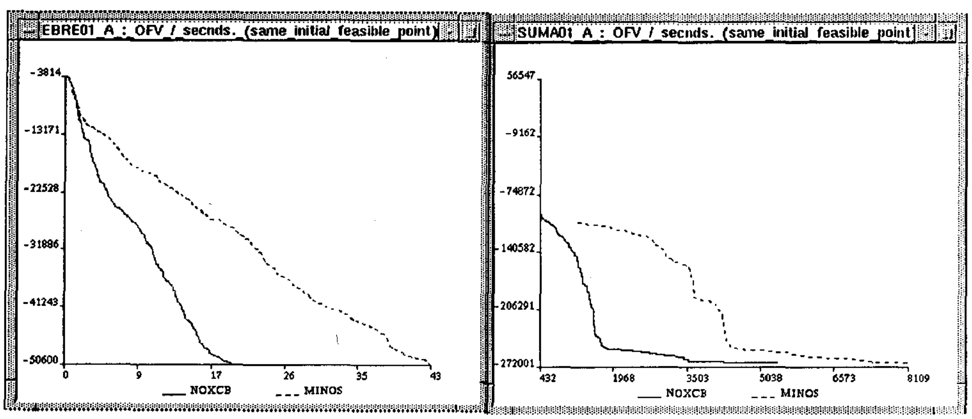
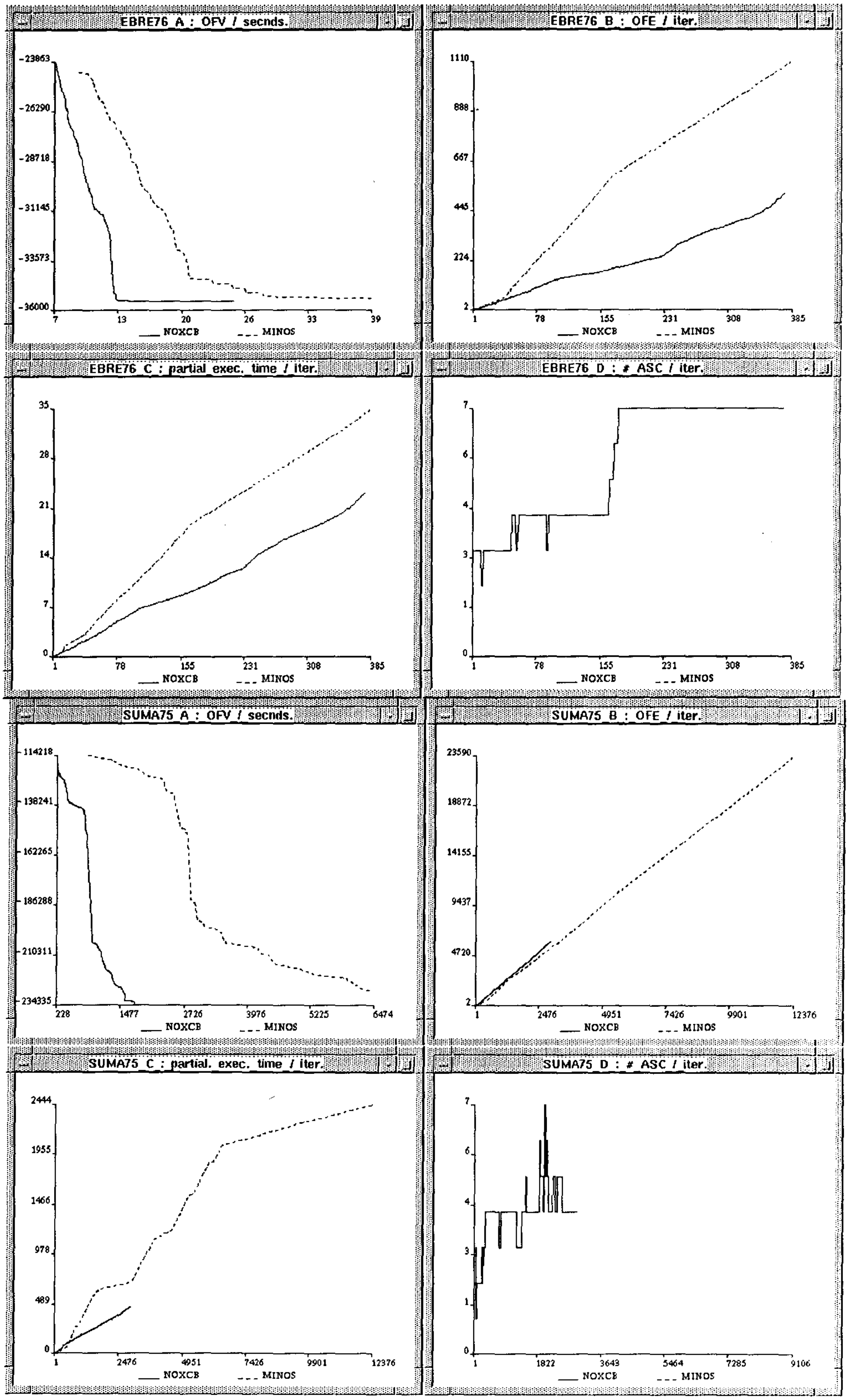


Figure 1 :
Comparative computational performance of MINOS and NOXCB with problems EBRE76 and SUMA75



4.4. Comments on the nonconvexity of the objective function.

Due to the nonconvexity of the objective function, any local optimization technique applied to the HTC problem will possibly stop at a local minimum. The solutions found either by NOXCB or MINOS could not be the global optimum. Thus the user may wish to check the results against the value of the objective function for well known operating conditions of the reservoir system, may have lower objective function values than some local optima found by the program. Thus it is useful to be able to introduce feasible operating points in the program. This is not easy because variables must be separated into basic, superbasic and nonbasic, and the set of basic variables must lead to a non-singular basis. From these points the programs can try to reach a better solution.

Different optimizers have sometimes been obtained with MINOS and NOXCB for some problems, which correspond to different local minimizers.

5. CONCLUSIONS.

The structure of a specialized code of nonlinear network flows with linear side constraints has been described and analyzed. The application of such a code to real Short Term Hydrothermal Coordination problems of many sizes has been presented and has been compared to the performance of the general purpose MINOS package when used to solve the same problems. A first comparative analysis shows that the specialized code performs better though not by an order of magnitude as it is the case for network problems without side constraints.

REFERENCES

1. T.Y. Lee and N. Chen, "The effect of pumped storage and battery energy storage systems on hydrothermal generation coordination". IEEE Transactions on Power Systems, to appear.
2. G.H. Bradley, G.G. Brown and G.W. Graves, "Design and implementation of large scale transshipment algorithms" Management Science 24 (1977) 1–34.
3. B.A. Murtagh and M.A. Saunders, "Large–scale linearly constrained optimization", Mathematical Programming 14 (1978) 41–72.
4. R.S. Dembo, "A primal truncated newton algorithm with application to large-scale nonlinear network optimization", Mathematical Programming Studies 31 (1987) 43–71.
5. Ph.L. Toint and D.Tuyttens, "On large scale nonlinear network optimization", Mathematical Programming 48 (1990) 125–159.
6. R.E. Rosenthal " A nonlinear network flow algorithm for maximization of benefits in a hydroelectric power system". Operations research, 29 (1981) 763–786.
7. L.F. Escudero "Performance evaluation of independent superbasics set on nonlinear replicated networks". European Journal of Operations Research, 23 (1986) 343–355.
8. C. Lyra and H. Tavares, "A contribution to the midterm scheduling of large scale hydrothermal power systems". IEEE Transactions on Power Systems, Vol. 3, No. 3, August 1988.
9. H. Brännlund, D. Sjelvgren and J.A. Bubenko, "Short term generation scheduling with security constraints". IEEE Transactions on Power Systems, Vol. 3, No. 1, February 1988.
10. J.L. Kennington and R.V. Helgason , "Algorithm for network programming" , (John Wiley & Sons, New York, 1980).
11. F.J. Heredia, N. Nabona, "Large scale nonlinear network optimization with linear side constraints", EURO XI, European Congress on Operational Research, Aachen, Germany, July 16–19, 1991.
12. F.J. Heredia, N. Nabona, "Numerical implementation and computational results of nonlinear network optimization with linear side constraints", 15th IFIP Conference on system modelling and optimization, Zurich, Switzerland, September 2–6, 1991, Springer–Verlag (to appear).
13. R.S. Dembo and T. Steihaug, "Truncated–Newton algorithms for large–scale unconstrained optimization", Mathematical Programming 26 (1983) 190–212.
14. D.P. Bertsekas, "Constrained optimization and Lagrange multiplier methods" , (Academic Press, London, 1982).
15. B.A. Murtagh and M.A. Saunders, "MINOS 5.1 User's Guide." Technical Report SOL 83-20R (1987). Dept. of Operations Research, Standford University, CA 9430, USA.
16. N. Nabona, " Descripció del programa LEXA", research report of the Facultat d'Informàtica de Barcelona (to appear).
17. F.J. Heredia and N. Nabona, " Programa FXCB de fluxos lineals en xarxes amb constriccions a banda lineals". research report RR 90/06 Facultad d'Informàtica de Barce-lona, 1990.