Nonlinear Network flow Problems with Side Constraints Through Projected Lagrangian Methods.

F. Javier Heredia

Dept. of Statistics and Operations Research
Polytechnic University of Catalonia

EURO XV / INFORMS XXXIV
july 1997

FIRST PART : General overview of the method.

- The Nonlinear Network Flow problem with Side Constraints (NNC).
- The projected Lagrangian methodology.

SECOND PART : Optimization of the linearized subproblems (NNLC)^k: The NOXCB09 package.

- The primal partitioning active-set method.
- Computational results.

THIRD PART : The use of the NOXCB09 library into a projected Lagrangian methodology

- Structure of the prototype implementation.
- The MAPH4 program.
- Preliminary computational results.

CONCLUSIONS

FIRST PART : General overview of the method.

FORMULATION OF THE (NNC) PROBLEM

$$
\textbf{(NNC)}\left\{\begin{array}{lll}
\min & f(x) & (1a)\\
\text{subj. a :} & & \\
& Ax = r & (1b)\\
& c(x) + \mathbf{I}_n z_n = b_n & (1c)\\
& Tx + \mathbf{I}_l z_l = b_l & (1d)\\
& & \\
& 0 \le x \le u_x & (1e)\\
& 0 \le z_n \le u_{zl} & (1f)\\
& 0 \le z_l \le u_{zn} & (1g)
\end{array}\right.
$$

(1a) : Objective function: $f\colon \mathbb{R}^n \to \mathbb{R}$, $f \in \mathcal{C}^2$

(1b) : Network equations: $A \in \mathbb{R}^{m \times \tilde{n}}$, $r \in \mathbb{R}^m$

(1c,1d) : Nonlinear and linear side constraints:
$c\colon \mathbb{R}^{t_n} \to \mathbb{R}$, $c \in \mathcal{C}^2$, $b_n \in \mathbb{R}^{t_n}$
$T \in \mathbb{R}^{t_l \times \tilde{n}}$, $\text{rang}(T) = t_l$, $b \in \mathbb{R}^{t_l}$

(1e,1f,1g) : Arcs and slacks capacities:
$x, u_x \in \mathbb{R}^n$
$z_n, u_{zn} \in \mathbb{R}^{\tilde{t}}_n$, $\mathbf{I}_n \in \mathbb{R}^{t_n \times \tilde{t}_n}$
$z_l, u_{zl} \in \mathbb{R}^{\tilde{t}}_l$, $\mathbf{I}_n \in \mathbb{R}^{t_l \times \tilde{t}_l}$

THE PROJECTED LAGRANGIAN METHOD

Algorithm A1 : Projected Lagrangian Algorithm

- Select some starting values for x^0, λ^0 and ρ^0.

MAJOR ITERATION

Do While x^k not optimal for **(NNC)**

- Linearize the nonlinear side constraints at x^k:

$$c(x) \approx c^k(x) = c(x^k) + \nabla c(x^k)(x - x^k) \tag{2}$$

- **Minor iteration:** Solve the following *Nonlinear Network flow problem with Linear side Constraints* $\mathbf{(NNLC)}^k$:

$$\mathbf{(NNLC)}^k \begin{cases} \min \quad \Phi^k(x) & (3a) \\ \text{subj. a :} & \\ \quad Ax = r & (3b) \\ \quad \nabla c(x^k)x + \mathbf{I}_n z_n = \tilde{b}_n & (3c) \\ \quad Tx + \mathbf{I}_l z_l = b_l & (3d) \\ \quad 0 \le x \le u_x & (3e) \\ \quad 0 \le z_n \le u_{zl} & (3f) \\ \quad 0 \le z_l \le u_{zn} & (3g) \end{cases}$$

where :

$$\tilde{b}_n = b_n - c(x^k) + \nabla c(x^k)x^k$$

$$\Phi^k(x) = \Phi^k(x; \lambda^k, \ldots) \text{ (\textit{Merit function})}$$

- Update $\Phi^k(x)$.
- Set $k := k + 1$

End Do

RELEVANT FACTS.

- **Local convergence:**
  - Robinson (1972) proved the q-quadratically local convergence of the method, if the merit function and the iterates $\{x^{k+1}, \lambda^{k+1}\}$ are taken as:

$$\Phi^k(x) = f(x) - \lambda^{k\prime}\Big(c(x) - c^k(x)\Big)$$
$$\{x^{k+1}, \lambda^{k+1}\} := \{[x^*]^k, [\lambda^*]^k\}$$

- **Computational implementations :**
  - MINOS (Murtagh & Saunders 1982).
  - Their merit function is a *modified augmented Lagrangian function* $\Phi^k(x)$:

$$\Phi^k(x; x^k, \lambda^k, \rho^k) = f(x) - \lambda^{k\prime}\Big(c(x) - c^k(x)\Big) + \frac{\rho^k}{2}\|c(x) - c^k(x)\|_2$$

- **Efficient solution of the (NNLC)^{k}subproblems:**
  - NOXCB09 (Heredia and Nabona 1995).
  - Active set method specialized trough primal partitioning techniques.

SECOND PART : Optimization of the linearized subproblems $(\mathrm{NNLC})^k$: The NOXCB09 package.

FORMULATION OF THE PROBLEM $(\mathbf{NNLC})^k$

$$(\mathbf{NNLC}) \begin{cases} \min & f(x) & (4a) \\ \text{subj. a :} & Ax = r & (4b) \\ & Tx + \mathbf{I}_z z = b & (4c) \\ & 0 \leq x \leq u_x & (4d) \\ & 0 \leq z \leq u_z & (4e) \end{cases}$$

(4a) : Objective function.
$f : \mathbb{R}^n \to \mathbb{R}$, $f \in \mathcal{C}^2$

(4b) : Network equations
$A \in \mathbb{R}^{m \times \tilde{n}}$, $r \in \mathbb{R}^m$

(4c) : Linear side constraints
$T \in \mathbb{R}^{t \times \tilde{n}}$, $\text{rang}(T) = t$, $b \in \mathbb{R}^t$

(4d) : Arcs capacities
$x, u_x \in \mathbb{R}^n$

(4e) : Bounds of the slacks
$z, u_z \in \mathbb{R}^{\tilde{t}}$, $\mathbf{I}_z \in \mathbb{R}^{t \times \tilde{t}}$
($\tilde{t}$: number of inequality side constraints)

Solution procedure for the (NNLC) subproblem.

Algorithm A2 : Active set method

Find an initial feasible solution x^0.

Define the *basic* $\mathcal{B}^0$, *superbasic* $\mathcal{S}^0$ and *nonbasic* $\mathcal{N}_0^0$, $\mathcal{N}_u^0$ variables associated to the current active set at x^0.

Compute Z^0, $g_z^0 = Z^{0\prime}\nabla f(x^0)$ and the Lagrange multipliers of the nonbasic variables ${\sigma_0}^0$ and ${\sigma_u}^0$.

Set $k := 0$

Do

Relax some nonbasic variable from its active bound.

Update $\mathcal{B}^k$, $\mathcal{S}^k$, $\mathcal{N}^k$, Z^k and ${g_z}^k$.

Do While $\|g_z\| > {\epsilon_G}^k$

Compute a feasible descent direction :

$${H_z}^k {p_z}^k = -{g_z}^k \quad ; \quad p^k := Z^k {p_z}^k$$

Find the maximal step length: $\bar{\alpha} = \min\{\bar{\alpha}_{\mathcal{B}}, \bar{\alpha}_{\mathcal{S}}\}$.

Linesearch: $\alpha^{*k} = \text{argmin}_{0<\alpha\leq\bar{\alpha}}\{f(x^k + \alpha p^k)\}$.

Update the current solution: $x^{k+1} := x^k + \alpha^{*k} p^k$.

If $\alpha^{*k} = \bar{\alpha}$ then

Update $\mathcal{B}^k$, $\mathcal{S}^k$, $\mathcal{N}^k$

End If

Update Z^k and ${g_z}^k$.

$k := k + 1$

End Do

Compute the Lagrange multipliers estimates ${\sigma_0}^k$, ${\sigma_u}^k$.

Until ${\sigma_0}^k \geq -\epsilon_O$ and ${\sigma_u}^k \leq \epsilon_O$

STRUCTURE OF THE PROBLEM (NNLC)

- **Equality constraint matrix.**

$$M = \begin{matrix} \\ m \\ t \end{matrix} \overbrace{\begin{matrix} \\ \left\{ \\ \left\{ \end{matrix} \begin{bmatrix} A & \mathbf{0} \\ T & \mathbf{I}_z \end{bmatrix}}^{\begin{matrix} \tilde{n} & \tilde{t} \end{matrix}} = \begin{array}{|c|c|c|} \multicolumn{1}{c}{m} & \multicolumn{1}{c}{s} & \multicolumn{1}{c}{n-m-s} \\ \hline B & S & N \\ \hline \end{array}$$

- **Superbasic S, non-basic N and basic B matrices.**

$$S = \begin{array}{|c|c|} \multicolumn{1}{c}{s_x} & \multicolumn{1}{c}{s_z} \\ \hline A_{\mathcal{S}} & \mathbf{0} \\ \hline T_{\mathcal{S}} & \mathbf{I}_{\mathcal{S}} \\ \hline \end{array} \begin{matrix} m \\ t \end{matrix} \quad ; \quad N = \begin{array}{|c|c|} \multicolumn{1}{c}{|\mathcal{N}_x|} & \multicolumn{1}{c}{|\mathcal{N}_z|} \\ \hline A_{\mathcal{N}} & \mathbf{0} \\ \hline T_{\mathcal{N}} & \mathbf{I}_{\mathcal{N}} \\ \hline \end{array} \begin{matrix} m \\ t \end{matrix}$$

$$B = \begin{array}{|c|c|c|} \multicolumn{1}{c}{m} & \multicolumn{1}{c}{c_x} & \multicolumn{1}{c}{c_z} \\ \hline A_{\mathcal{A}} & A_{\mathcal{C}} & \mathbf{0} \\ \hline T_{\mathcal{A}} & T_{\mathcal{C}} & \mathbf{I}_{\mathcal{C}} \\ \hline \end{array} \begin{matrix} m \\ t \end{matrix}$$

PRIMAL PARTITIONING

- **Every basis of the (NNLC) problem may be placed in the following form (Kennington and Helgason 1980):**

$$B = \begin{array}{c|cc|c} m & c_x & c_z & \\ \hline A_{\mathcal{A}} & A_c & \mathbf{0} & m \\ \hline T_{\mathcal{A}} & T_c & \mathbf{I}_c & t \\ \hline \textit{Key} & \textit{Non Key} & & \end{array} \qquad (5)$$

  where $A_{\mathcal{A}}$ is a basis (spanning tree) of the network matrix A

- **The systems of equations $Bv = w$ and $v'B = w$ can be computed solving:**

  * Systems of equations with the matrix $A_{\mathcal{A}}$, which can be computed efficiently through network flow techniques.

  * Systems of equations with the general (dense) matrix Q .

$$Q = [\, T_c \quad | \quad \mathbf{I}_c \,] - T_{\mathcal{A}} A_{\mathcal{A}}^{-1} [A_c | \mathbf{0}] \in \mathbb{R}^{t \times t}$$

(working basis)

COMPUTATIONS WITH THE BASIC MATRIX B

- **Resolution of the linear system $Bw = v$:**

$$w = \begin{bmatrix} w_{\mathcal{A}} \\ \hline w_{\mathcal{C}} \end{bmatrix} = B^{-1} \begin{bmatrix} v_A \\ \hline v_T \end{bmatrix} = \left[\frac{A_{\mathcal{A}}^{-1}\left(v_A - [A_{\mathcal{C}}|\mathbf{0}]Q^{-1}(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A)\right)}{Q^{-1}\left(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A\right)} \right] \tag{6a}$$

$$= \left[\frac{A_{\mathcal{A}}^{-1}v_A - [\Theta_{\mathcal{C}}|\mathbf{0}]Q^{-1}\left(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A\right)}{Q^{-1}\left(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A\right)} \right] \tag{6b}$$

Where $A_{\mathcal{A}}^{-1}A_{\mathcal{C}} = \Theta_{\mathcal{C}}$, is the matrix of the basic cycles of the non-key arcs.

- **Algorithm to compute $Bw = v$:**

Algorithm A3 :

$w = B^{-1}v$, without $\overset{\circ}{\mathcal{C}}$.

1	$\gamma_1 \overset{F.X.}{\longleftarrow} A_{\mathcal{A}}^{-1}v_A$
2	$\gamma_2 \longleftarrow v_T - T_{\mathcal{A}}\gamma_1$
3	$w_{\mathcal{C}} \longleftarrow Q^{-1}\gamma_2$
4	$\gamma_2 \longleftarrow v_A - [A_{\mathcal{C}}|\mathbf{0}]w_{\mathcal{C}_x}$
5	$w_{\mathcal{A}} \overset{F.X.}{\longleftarrow} A_{\mathcal{A}}^{-1}\gamma_2$

Algorithm A4 :

$w = B^{-1}v$, with $\overset{\circ}{\mathcal{C}}$.

1	$\gamma_1 \overset{F.X.}{\longleftarrow} A_{\mathcal{A}}^{-1}v_A$
2	$\gamma_2 \longleftarrow v_T - T_{\mathcal{A}}\gamma_1$
3	$w_{\mathcal{C}} \longleftarrow Q^{-1}\gamma_2$
4	$w_{\mathcal{A}} \overset{A4.1}{\longleftarrow} \gamma_1 - \Theta_{\mathcal{C}}w_{\mathcal{C}_x}$

$$\textbf{Product } w = Zv = \left[\begin{array}{c} -B^{-1}S \\ \hline \mathbf{I} \\ \hline \mathbf{0} \end{array}\right] v$$

- **Option A:** first $\gamma_{\mathcal{A}} = -Sv$ and then $w = B^{-1}\gamma$

$$Sv = \begin{bmatrix} A_{\mathcal{S}} & \mathbf{0} \\ T_{\mathcal{S}} & \mathbf{I}_{\mathcal{S}} \end{bmatrix} \begin{bmatrix} v_x \\ v_z \end{bmatrix} = \left[\begin{array}{c} A_{\mathcal{S}} v_x \\ \hline T_{\mathcal{S}} v_x + \mathbf{I}_{\mathcal{S}} v_z \end{array}\right] \tag{7}$$

Algorithm A5 : $-B^{-1}(Sv)$

1 $\gamma_{\mathcal{A}} \longleftarrow -A_{\mathcal{S}} v_x$

2 $\gamma_{\mathcal{T}} \longleftarrow -T_{\mathcal{S}} v_x - \mathbf{I}_{\mathcal{S}} v_z$

3 $w_{\mathcal{B}} \overset{\text{A3}}{\longleftarrow} B^{-1}\gamma$

- **Option B:** to develop $B^{-1}S$.

$$w_{\mathcal{B}} = \left[\begin{array}{c} w_{\mathcal{A}} \\ \hline w_{\mathcal{C}} \end{array}\right] = -B^{-1}Sv = \left[\begin{array}{c} -\Theta_{\mathcal{S}} v_x - [\Theta_{\mathcal{C}} | \mathbf{0}] Q^{-1}\left(T_{\mathcal{A}}\Theta_{\mathcal{S}} v_x - T_{\mathcal{S}} v_x - \mathbf{I}_{\mathcal{S}} v_z\right) \\ \hline Q^{-1}\left(T_{\mathcal{A}}\Theta_{\mathcal{S}} v_x - T_{\mathcal{S}} v_x - \mathbf{I}_{\mathcal{S}} v_z\right) \end{array}\right] \tag{8}$$

Algorithm A6 : $w_{\mathcal{B}} = Zv$.

1 $\gamma_1 \overset{\text{A4.2}}{\longleftarrow} \Theta_{\mathcal{S}} v_x$

2 $\gamma_2 \longleftarrow T_{\mathcal{A}}\gamma_1 - T_{\mathcal{S}} v_x - \mathbf{I}_{\mathcal{S}} v_z$

3 $[\, w'_{\mathcal{C}_x} \;|\; w'_{\mathcal{C}_z} \,] \longleftarrow -Q^{-1}\gamma_2$

4 $w_{\mathcal{A}} \overset{\text{A4.1}}{\longleftarrow} \gamma_1 - \Theta_{\mathcal{C}} w_{\mathcal{C}_x}$

OTHER IMPLEMENTATION ISSUES

- **Linesearch :**
  - ∗ Bertsekas method (Bertsekas (1982), Toint & Tuyttens (1991)).
  - ∗ Bactracking linesearch with quadratic and cubic fit.
- **Descent direction computation:** $\tilde{H}_z p_z = -g_z(x)$
  - ∗ (TNM) Truncated Newton Method (Dembo & Steihaug (1983), Toint & Tuyttens (1991)).
  - ∗ (QNM) Quasi-Newton method (Murtagh & Saunders (1978)).
- **Working basis update** $[Q^{-1}]^k$ **:** cada cert nombre d'iteracions
  - ∗ **Product form of the inverse :**

$$[Q^{-1}]^k = \prod_{t \geq i \geq 1} E_r^i$$

    - ▷ P^3 algorithm of Hellerman & Rarick : $[Q^{-1}]^k = R\left(\prod_{t \geq i \geq 1} \tilde{E}_r^i\right) P$
    - ▷ Partial pivoting.
  - ∗ **LU factorization.**
  - ▷ Subroutines F01BRF and F04AXF (NAG library).

EIO/UPC problem tests

- **110 problemes (NNLC).**
  - ∗ **DIMACS problems:**
    - ▷ Random generators of linear network flow problems (FL) :

      Rmfgen (Goldfarb, Grigoriadis).
      Grid-on-Torus (Goldberg).
      Gridgen (Lee & Orlin).

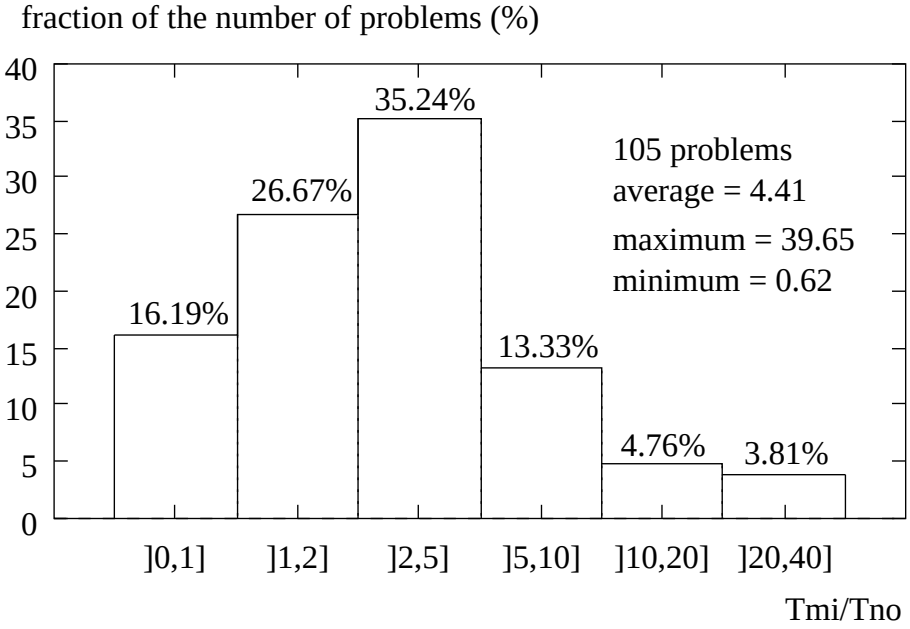
    - ▷ Addition of side constraints to the DIMACS problems: *Di2no*.
    - ▷ Objective functions :

      Namur (Toint & Tuyttens).
      EIO1.

  - ∗ **Real Short term Hydro-Thermal Coordination problems.**
- **Characteristics of the EIO/UPC test collection:**
  - ∗ 1524 arcs, 360 nodes → 23832 arcs and 6589 nodes.
  - ∗ Ratio (# of side const./ # network equations) : 1% → 100%
  - ∗ # of nonzero elements of the S.C. : 0.02% → 10%
  - ∗ # of superbasic variables at the optimal solution: 0 → 3352
- **Hardware configuration :**
  - ∗ Workstation Sun Sparc 10/41 with a single processor at 40MHz (≈100Mips, 20Mflops).
  - ∗ 64Mb of central memory (32Mb real memory, 32Mb swapping memory).

COMPUTATIONAL RESULTS

- **Efficiency of NOXCB09 vs. MINOS 5.3:**



- **Dependency of the efficiency on the size of the problem:**

Scale	**Maximal dimensions**			t_{MI}/t_{NO}		
	arcs	nodes	# S.C.	Average	Maximum	Minimum
Large	18000	3000	750	5.10	17.96	0.84
Medium	8064	2479	840	2.87	10.00	0.92
Small	2256	697	240	1.34	2.47	0.62

∗ *The efficiency of NOXCB09 seems to increase with the dimension of the* **(NNLC)** *problem.*

THIRD PART : The use of the NOXCB09 library into a projected Lagrangian methodology

STRUCTURE OF THE IMPLEMENTATION

Algorithm A7 : a NOXCB09 based Projected Lagrangian method

Select some starting values for x^0, λ^0 and ρ^0.

Set $k := 0$

Do

Linearize the nonlinear side constraints at x^k.

Do

Solve $(\mathbf{NNLC})^k$ with `NOXCB09` and the merit function $\Phi^k(x; x^k, \lambda^k, \rho^k)$. This optimization provides $[x^*]^k$ and $[\lambda^*]^k$.

If $(\mathbf{NNLC})^k$ infeasible then

Relax the RHS of the linearized n.s.c. $\tilde{b}_n$.

End If

Until $(\mathbf{NNLC})^k$ feasible

Find new values of x^k and λ^k from $[x^*]^k$ and $[\lambda^*]^k$.

Update de penalty parameter ρ^k, if necessary.

Set $k := k + 1$

Until x^k optimal for **(NNC)**

- **Updates rules** the update rules follows those suggested by Murtagh & Saunders (1982) and implemented in MINOS.

PROTOTYPE IMPLEMENTATION

- **MAPH4** a projected Lagrangian NOXCB09 based package to solve the Short-Term Hydrothermal Scheduling problem

PRELIMINARY COMPUTATIONAL RESULTS

Table I : characteristics of the **(NNC)** problems.

Problem ident.	Power system size					**(NNC)** problem size			
	Nr	Nu	Nm	Nb	Ni	arcs	nodes	n.s.c.	l.s.c.
A24x	3	4	6	5	24	888	290	24	96
B24x	6	4	6	5	24	1152	355	24	96
B48x	6	4	6	5	48	2256	697	48	192

Table II : Computational results.

Problem ident.	Major/Minor iter		CPU (s) [1]		$f(x^*)$	
	MAPH4	MINOS	MAPH4	MINOS	MAPH4	MINOS
A24x	3/2384	53/2030	2.3	4.8	0.07879	0.07883
B24x	3/3947	99/3855	3.1	15.3	0.0667	0.0635
B48x	3/7988	119/4656	6.4	36.2	0.1317	0.1319

[1] SUN UltraSparc 2

CONCLUSIONS

- A projected Lagrangian methodology for the **(NNC)** has been proposed.
- This methodology explotes the network structure of the problem applying the `NOXCB09` package to solve the linearized subproblems $\mathbf{(NNLC)}^k$.
- The efficiency of the `NOXCB09` code has been proved through extensive computational tests.
- A prototype implementation, (`MAPH4`) of the proposed projected Lagrangian method has been developped to solve the (STHTC) problem.
- The preliminary computational results from a limited set of **(NNC)** problems seems to corroborate the efficiency of the method, although more computational evidence is needed.