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

F. Javier Heredia

Dept. of Statistics and Operations Research
Polytechnic University of Catalonia

> ## 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

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

(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:
$$c \colon \mathbb{R}^{t_n} \to \mathbb{R} \ , \ c \in \mathcal{C}^2 \ , \ b_n \in \mathbb{R}^{t_n}$$
$$T \in \mathbb{R}^{t_l \times \tilde{n}} \ , \ \operatorname{rang}(T) = t_l \ , \ b \in \mathbb{R}^{t_l}$$

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

THE PROJECTED LAGRANGIAN METHOD

Algorithm A1 : Projected Lagrangian Algorithm

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

MAJOR ITERATION

<u>Do While</u> x^k not optimal for **(NNC)**

- Linearize the nonlinear side constraints at x^k:
$$c(x) \approx c^k(x) = c(x^k) + \nabla c(x^k)(x - x^k) \qquad (2)$$

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

$$(\text{NNLC})^k \begin{cases} & \min \quad \Phi^k(x) & (3a) \\[1mm] & \text{subj. a :} \\[1mm] & \qquad Ax \qquad\qquad\ = r & (3b) \\[1mm] & \qquad \nabla c(x^k)x + \mathbf{I}_n z_n = \tilde{b}_n & (3c) \\[1mm] & \qquad Tx \qquad + \mathbf{I}_l z_l\ = b_l & (3d) \\[3mm] & \qquad 0 \le x\ \le u_x & (3e) \\[1mm] & \qquad 0 \le z_n \le u_{zl} & (3f) \\[1mm] & \qquad 0 \le z_l\ \le u_{zn} & (3g) \end{cases}$$

where :

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

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

- Update $\Phi^k(x)$.

- Set $k := k + 1$

<u>End Do</u>

$$\boxed{\textbf{RELEVANT FACTS.}}$$

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

- **Computational implementations :**

  * MINOS (Murtagh & Saunders 1982).

  * Their merit function is a *modified augmented Lagrangian function* $\Phi^k(x)$:
  $$\Phi^k(x; x^k, \lambda^k, \rho^k) = f(x) - \lambda^{k'}\Big(c(x) - c^k(x)\Big) + \frac{\rho^k}{2}\|c(x) - c^k(x)\|_2$$

- **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 (NNLC)^k:
The `NOXCB09` package.

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

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

(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

$z, u_z \in \mathbb{R}^{\tilde{t}} \, , \, \mathbf{I}_z \in \mathbb{R}^{t \times \tilde{t}}$

$(\tilde{t} \, : \, \text{number of inequality side constraints})$

$$\boxed{\textbf{Solution procedure for the (NNLC) subproblem.}}$$

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, $g_z^0 = Z^{0'} \nabla f(x^0)$ and the Lagrange multipliers of the nonbasic variables σ_0^0 and σ_u^0.

Set $k := 0$

<u>Do</u>

    Relax some nonbasic variable from its active bound.

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

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

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

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

        <u>If</u> $\alpha^{*k} = \bar{\alpha}$ <u>then</u>

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

        <u>End If</u>

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

        $k := k + 1$

    <u>End Do</u>

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

<u>Until</u> $\sigma_0{}^k \ge -\epsilon_O$ and $\sigma_u{}^k \le \epsilon_O$

STRUCTURE OF THE PROBLEM (NNLC)

- **Equality constraint matrix.**

$$M = \begin{array}{c} m\{ \\ t\{ \end{array} \left[\begin{array}{cc} \overbrace{A}^{\tilde{n}} & \overbrace{\mathbf{0}}^{\tilde{t}} \\ T & \mathbf{I}_z \end{array} \right] = \left[\begin{array}{c|c|c} \overset{m}{B} & \overset{s}{S} & \overset{n-m-s}{N} \end{array} \right]$$

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

$$S = \left[\begin{array}{c|c} \overset{s_x}{A_{\mathcal{S}}} & \overset{s_z}{\mathbf{0}} \\ \hline T_{\mathcal{S}} & \mathbf{I}_{\mathcal{S}} \end{array} \right] \begin{array}{c} m \\ \\ t \end{array} \quad ; \quad N = \left[\begin{array}{c|c} \overset{|\mathcal{N}_x|}{A_{\mathcal{N}}} & \overset{|\mathcal{N}_z|}{\mathbf{0}} \\ \hline T_{\mathcal{N}} & \mathbf{I}_{\mathcal{N}} \end{array} \right] \begin{array}{c} m \\ \\ t \end{array}$$

$$B = \left[\begin{array}{c|c|c} \overset{m}{A_{\mathcal{A}}} & \overset{c_x}{A_c} & \overset{c_z}{\mathbf{0}} \\ \hline T_{\mathcal{A}} & T_c & \mathbf{I}_c \end{array} \right] \begin{array}{c} m \\ \\ t \end{array}$$

PRIMAL PARTITIONING

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

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

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 $Bw = v$:**

$$w = \begin{bmatrix} w_{\mathcal{A}} \\ - \\ w_{c} \end{bmatrix} = B^{-1} \begin{bmatrix} v_A \\ - \\ v_T \end{bmatrix} = \begin{bmatrix} A_{\mathcal{A}}^{-1}\left(v_A - [A_c|\mathbf{0}]Q^{-1}(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A)\right) \\ \overline{} \\ Q^{-1}\left(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A\right) \end{bmatrix}$$

$$(6a)$$

$$= \begin{bmatrix} A_{\mathcal{A}}^{-1}v_A - [\Theta_c|\mathbf{0}]Q^{-1}\left(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A\right) \\ \overline{} \\ Q^{-1}\left(v_T - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_A\right) \end{bmatrix}$$

$$(6b)$$

Where $A_{\mathcal{A}}^{-1}A_c = \Theta_c$, 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	$w_c \longleftarrow Q^{-1}\gamma_2$
4	$\gamma_2 \longleftarrow v_A - [A_c|\mathbf{0}]w_{c_x}$
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	$w_c \longleftarrow Q^{-1}\gamma_2$
4	$w_{\mathcal{A}} \overset{A4.1}{\longleftarrow} \gamma_1 - \Theta_c w_{c_x}$

$$\boxed{\textbf{Product } w = Zv = \begin{bmatrix} -B^{-1}S \\ \hline \mathbf{I} \\ \hline \mathbf{0} \end{bmatrix} v}$$

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

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

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

1	$\gamma_A \longleftarrow -A_{\mathcal{S}}v_x$
2	$\gamma_T \longleftarrow -T_{\mathcal{S}}v_x - \mathbf{I}_{\mathcal{S}}v_z$
3	$w_{\mathcal{B}} \overset{\text{A3}}{\longleftarrow} B^{-1}\gamma$

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

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

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	$[\, w'_{c_x} \mid w'_{c_z} \,] \longleftarrow -Q^{-1}\gamma_2$
4	$w_{\mathcal{A}} \overset{\text{A4.1}}{\longleftarrow} \gamma_1 - \Theta_c w_{c_x}$

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 : $[Q^{-1}]^k = R \left(\prod_{t \geq i \geq 1} \tilde{E}_r^i \right) P$
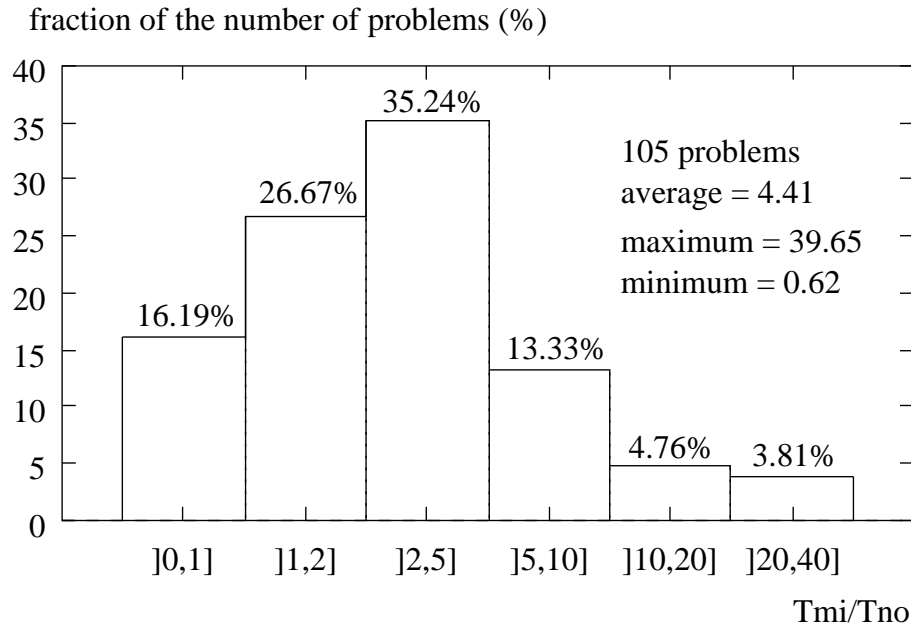
    ▷ Partial pivoting.

  * **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 ($\approx$100Mips, 20Mflops).

  - ∗ 64Mb of central memory (32Mb real memory, 32Mb swapping memory).

$$\boxed{\textbf{COMPUTATIONAL RESULTS}}$$

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

fraction of the number of problems (%)

105 problems
average = 4.41
maximum = 39.65
minimum = 0.62

16.19%   26.67%   35.24%   13.33%   4.76%   3.81%

]0,1]   ]1,2]   ]2,5]   ]5,10]   ]10,20]]20,40]

Tmi/Tno

- **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 $k := 0$

<u>Do</u>

Linearize the nonlinear side constraints at x^k.

<u>Do</u>

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 $[\lambda^*]^k$.

<u>If</u> $(\mathbf{NNLC})^k$ infeasible <u>then</u>

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

<u>End If</u>

<u>Until</u> $(\mathbf{NNLC})^k$ 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 $k := k + 1$

<u>Until</u> x^k optimal for $(\mathbf{NNC})$

- **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	Power system size					(NNC) problem size			
ident.	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	Major/Minor iter		CPU (s) [1]		$f(x^*)$	
ident.	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

$$\boxed{\textbf{CONCLUSIONS}}$$

- A projected Lagrangian methodology for the $(\textbf{NNC})$ has been proposed.

- This methodology explotes the network structure of the problem applying the `NOXCB09` package to solve the linearized subproblems $(\textbf{NNLC})^k$.

- 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 $(\textbf{NNC})$ problems seems to corroborate the efficiency of the method, although more computational evidence is needed.