19th IFIP TC7 Conference, Cambridge 1999

COMPUTATIONAL STUDY OF NOXCB:

AN OPTIMIZATION CODE FOR THE NONLINEAR NETWORK FLOW PROBLEM WITH LINEAR SIDE CONSTRAINTS

F. Javier Heredia

Department of Statistics and Operations Research
Politechnical University of Catalonia
Barcelona, Spain

FORMULATION OF THE (NNSC) PROBLEM

(NNSC)

$$
\begin{aligned}
\min \quad & f(x) && (1a) \\
\text{subj. to :} \quad & Ax = r && (1b) \\
& Tx + \mathbf{I}_z z = b && (1c) \\
& 0 \le x \le u_x && (1d) \\
& 0 \le z \le u_z && (1e) \\
& u_{x\,\tilde{n}} = 0 && (1f)
\end{aligned}
$$

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

(1d) : Arc capacity
$x, u_x \in \mathbb{R}^{\tilde{n}}$, $\tilde{n} = n+1$

(1e) : Bounds to the slacks
$z, u_z \in \mathbb{R}^{\tilde{t}}$, $\mathbf{I}_z \in \mathbb{R}^{t\times\tilde{t}}$
($\tilde{t}$: # of $\le$ side constraints)

(1f) : zero flow of the root arc

RELATION WITH OTHER OPTIMIZATION PROBLEMS

$$
\textbf{(NNSC)}\left\{
\begin{array}{rlr}
\min & f(x) & \text{(1a)} \\
\text{subj. to :} & Ax \qquad = r & \text{(1b)} \\
 & Tx + \mathbf{I}_z z = b & \text{(1c)} \\
 & 0 \le x \le u_x & \text{(1d)} \\
 & 0 \le z \le u_z & \text{(1e)} \\
 & u_{x\,\tilde{n}} = 0 & \text{(1f)}
\end{array}
\right.
$$

$$
\textbf{(LNSC)}\left\{
\begin{array}{rl}
\min & c'x \\
\text{s.t.:} & Ax \qquad = r \\
 & Tx + \mathbf{I}_z z = b \\
 & 0 \le x \le u_x \\
 & 0 \le z \le u_z \\
 & u_{x\,\tilde{n}} = 0
\end{array}
\right.
$$

Exploitation of the network structure. (Primal partitioning, Kennington & Helgason 1980)

$$
\textbf{(NN)}\left\{
\begin{array}{rl}
\min & f(x) \\
\text{s.t.:} & Ax \qquad = r \\
 & 0 \le x \le u_x \\
 & u_{x\,\tilde{n}} = 0
\end{array}
\right.
$$

General algorithmic structure (Active set algorithm with superbasic variables, Murtagh & Saunders 1978)

ALGORITHM FOR THE (NNSC) PROBLEM

Algorithm A3.1 : Active set algorithm.

0 **Initialization of the algorithm :**
Find a first feasible solution for **(NNSC)** $\rightarrow x^0$.
Define the sets $\mathcal{B}^0$, $\mathcal{S}^0$, $\mathcal{N}^0$ and the Active Set Reduced problem **(ASR)** associated to x^0:

$$(\textbf{ASR})^0 \left\{ \begin{array}{l} \min\limits_{x_\mathcal{S} \in \mathrm{IR}^s} f_z(x_\mathcal{S}) \\ \text{subj. to :} \\ \quad 0 \leq x_\mathcal{S} \leq u_\mathcal{S} \end{array} \right.$$

$k := 0$

1 **While** x^k **not optimal solution of (ASR)** ^{k} **:**

- **Find a feasible descent direction :**
  Resolution of $H_z{}^k p_z{}^k = -g_z{}^k$; $p^k := Z^k p_z{}^k$.
- **Maximum 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 iterate :** $x^{k+1} := x^k + \alpha^{*k} p^k$.
- If $\alpha^{*k} = \bar{\alpha}$ then: **Basis change**
  Update **(ASR)** ^{k} and $\mathcal{B}^k$, $\mathcal{S}^k$, $\mathcal{N}^k$
- $k := k + 1$.

2 x^k **optimal solution of (ASR)** ^{k}**, Pricing :**
If the current active set is optimal then:

STOP : x^k optimal solution of **(NNSC)** .

Otherwise, change the active set:

Update **(ASR)** ^{k} and $\mathcal{B}^k$, $\mathcal{S}^k$, $\mathcal{N}^k$.

Go to **1** .

STRUCTURE OF THE (NNSC) PROBLEM

- **Constraint matrix.**

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

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

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

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

PRIMAL PARTITIONING

- **The constraints matrix of problems (LNSC) i (NNSC) are the same ⇒ The primal partitioning method developped for the (LNSC) problem can be adapted to the (NNSC) problem:**

  ∗ Every basis of the **(NNSC)** could be expressed as :

$$B = \begin{array}{c|c|c|c} \multicolumn{1}{c}{m} & \multicolumn{1}{c}{c_x} & \multicolumn{1}{c}{c_z} & \\ \hline A_{\mathcal{A}} & A_c & \mathbf{0} & m \\ \hline T_{\mathcal{A}} & T_c & \mathbf{I}_c & t \\ \hline \multicolumn{1}{c}{\textit{Key columns}} & \multicolumn{2}{c}{\textit{Non key col.}} & \end{array} \qquad (2)$$

  where $A_{\mathcal{A}}$ is a basis of the matrix A (rooted tree) ⇒ **efficient network flow techniques could be applied**.

  ∗ The expression of B^{-1} is :

$$B^{-1} = \begin{bmatrix} A_{\mathcal{A}}^{-1} + A_{\mathcal{A}}^{-1}[A_c|\,\mathbf{0}]Q^{-1}T_{\mathcal{A}}A_{\mathcal{A}}^{-1} & -A_{\mathcal{A}}^{-1}[A_c|\,\mathbf{0}]Q^{-1} \\ -Q^{-1}T_{\mathcal{A}}A_{\mathcal{A}}^{-1} & Q^{-1} \end{bmatrix}$$

  with $Q = [T_c \;\;|\;\; \mathbf{I}_c] - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}[A_c|\,\mathbf{0}]$ being a $t \times t$ matrix (**working basis**).

EXPLOITATION OF THE PRIMAL PARTITIONING

- **Matrix computations involving B :**
  - **Resolution of the linear system $Bw = v$**

    Computation of the feasible descent direction associated with $x_{\mathcal{B}}$.
  - **Resolution of the linear system $w'B = v'$.**

    Computation of π and its update.
- **Operations with Z:**
  - **Product $w = Zv$:**

    Computation of the direction $p = Zp_z$.
    Application of the Truncated Newton Method.
    Bertsekas's linesearch.
  - **Product $w = Z'v$:**

    Computation of the reduced gradient $g_z = Z'g$.
    Application of the Truncated Newton Method.

EXPLOITATION OF THE PRIMAL PARTITIONING

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

$$w = \begin{bmatrix} w_{\mathcal{A}} \\ \text{—} \\ w_{\mathcal{C}} \end{bmatrix} = B^{-1} \begin{bmatrix} v_{\mathcal{A}} \\ \text{—} \\ v_{\mathcal{T}} \end{bmatrix} = \begin{bmatrix} A_{\mathcal{A}}^{-1}\Big(v_{\mathcal{A}} - [A_{\mathcal{C}}|\mathbf{0}]Q^{-1}(v_{\mathcal{T}} - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_{\mathcal{A}})\Big) \\ \hline Q^{-1}\left(v_{\mathcal{T}} - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_{\mathcal{A}}\right) \end{bmatrix} = \tag{3a}$$

$$= \left|\Theta_{\mathcal{C}} = A_{\mathcal{A}}^{-1}A_{\mathcal{C}}\right| = \begin{bmatrix} A_{\mathcal{A}}^{-1}v_{\mathcal{A}} - [\Theta_{\mathcal{C}}|\mathbf{0}]Q^{-1}\left(v_{\mathcal{T}} - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_{\mathcal{A}}\right) \\ \hline Q^{-1}\left(v_{\mathcal{T}} - T_{\mathcal{A}}A_{\mathcal{A}}^{-1}v_{\mathcal{A}}\right) \end{bmatrix} \tag{3b}$$

$\Theta_{\mathcal{C}} = A_{\mathcal{A}}^{-1}A_{\mathcal{C}}$, basic loops of the non-key arcs $\mathcal{C}_x$.

- **Product** $w = Zv = [\, w_{\mathcal{B}}' \mid w_{\mathcal{S}}' \mid w_{\mathcal{N}}' \,] = [\, -B^{-1}Sv \mid v \mid 0 \,]'$

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

$$Sv = \begin{bmatrix} A_{\mathcal{S}} & \mathbf{0} \\ T_{\mathcal{S}} & \mathbf{I}_{\mathcal{S}} \end{bmatrix} \begin{bmatrix} v_x \\ v_z \end{bmatrix} = \begin{bmatrix} A_{\mathcal{S}}v_x \\ \hline T_{\mathcal{S}}v_x + \mathbf{I}_{\mathcal{S}}v_z \end{bmatrix} \tag{4}$$

  * **Second option :** the product $B^{-1}S$ is developed.

$$w_{\mathcal{B}} = \begin{bmatrix} w_{\mathcal{A}} \\ \text{—} \\ w_{\mathcal{C}} \end{bmatrix} = -B^{-1}Sv = \left| \;\Theta_{\mathcal{S}} = A_{\mathcal{A}}^{-1}A_{\mathcal{S}}\,;\; \Theta_{\mathcal{C}} = A_{\mathcal{A}}^{-1}A_{\mathcal{C}} \;\right| =$$

$$= \begin{bmatrix} -\Theta_{\mathcal{S}}v_x - [\Theta_{\mathcal{C}}|\mathbf{0}]Q^{-1}\Big(T_{\mathcal{A}}\Theta_{\mathcal{S}}v_x - T_{\mathcal{S}}v_x - \mathbf{I}_{\mathcal{S}}v_z\Big) \\ \hline Q^{-1}\Big(T_{\mathcal{A}}\Theta_{\mathcal{S}}v_x - T_{\mathcal{S}}v_x - \mathbf{I}_{\mathcal{S}}v_z\Big) \end{bmatrix} \tag{5}$$

$\Theta_{\mathcal{S}} = A_{\mathcal{A}}^{-1}A_{\mathcal{S}}$, basic loops of the superbasic arcs $\mathcal{S}_x$.

WORKING BASIS

- **Computation of** $[Q^{-1}]^k$ **:** once after a given number of iterations
  - $*$ **Product form of the inverse (PFI) :**

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

    - ▷ Hellerman & Rarick's P^3 algorithm : $[Q^{-1}]^k = R\left(\prod_{t\geq i\geq 1} \tilde{E}_r^i\right) P$
    - ▷ Partial pivoting.
  - $*$ *LU* **Factorization**
    - ▷ Subroutines F01BRF and F04AXF of the NAG library.

OPTIMIZATION OF (ASR) PROBLEM.

$$\textbf{(ASR)}^k \left\{ \begin{array}{l} \quad \min\limits_{x_S \in \mathbb{R}^s} \; f_z(x_S) \\ \text{subj. a :} \\ \qquad\qquad 0 \leq x_S \leq u_S \end{array} \right.$$

- **Linesearch :**
  - $*$ Bertsekas's method (Bertsekas (1982), Toint & Tuyttens (1991)).
  - $*$ Linesearch through curbe fitting and backtracking (GETPTC subroutine, the same used in MINOS).
- **Computation of the descent directions:** $\tilde{H}_z p_z = -g_z(x)$
  - $*$ (TNM) Truncated Newton Method (Dembo & Steihaug (1983), Toint & Tuyttens (1991)).
  - $*$ (QNM) Quasi-Newton Method (BFGS update, as proposed by Murtagh & Saunders (1978)).

FINDING AN INITIAL FEASIBLE SOLUTION

Algorithm A5.1 : Computation of a feasible solution for **(NNSC)**

1 **Phase 0** : $\hat{x}$, a feasible solution w.r.t. the network equations and bounds is computed (*pseudo-feasible solution*):

$$A\hat{x} = r \quad , \quad 0 \leq \hat{x} \leq u_x$$

2 **Phase 1** : the infeasibilities of the side constraints are eliminated by solving the following **(LNSC)** problem :

$$\begin{aligned} \min \quad & z_a = \sum_{i=1}^{|\mathrm{I}|} f_i \\ \text{subj. to :} \quad & Ax = r \\ & Tx + \mathbf{I}_z z + \tilde{\mathbf{I}} f = b \\ & 0 \leq x \leq u_x \; ; \; 0 \leq z \leq u_z \; ; \; f \geq 0 \end{aligned}$$

with $\hat{z}$ being the slacks of the side constraints not violated at $\hat{x}$, and $\hat{f}$ the artificial variables associated to the side constraints violated at $\hat{x}$.

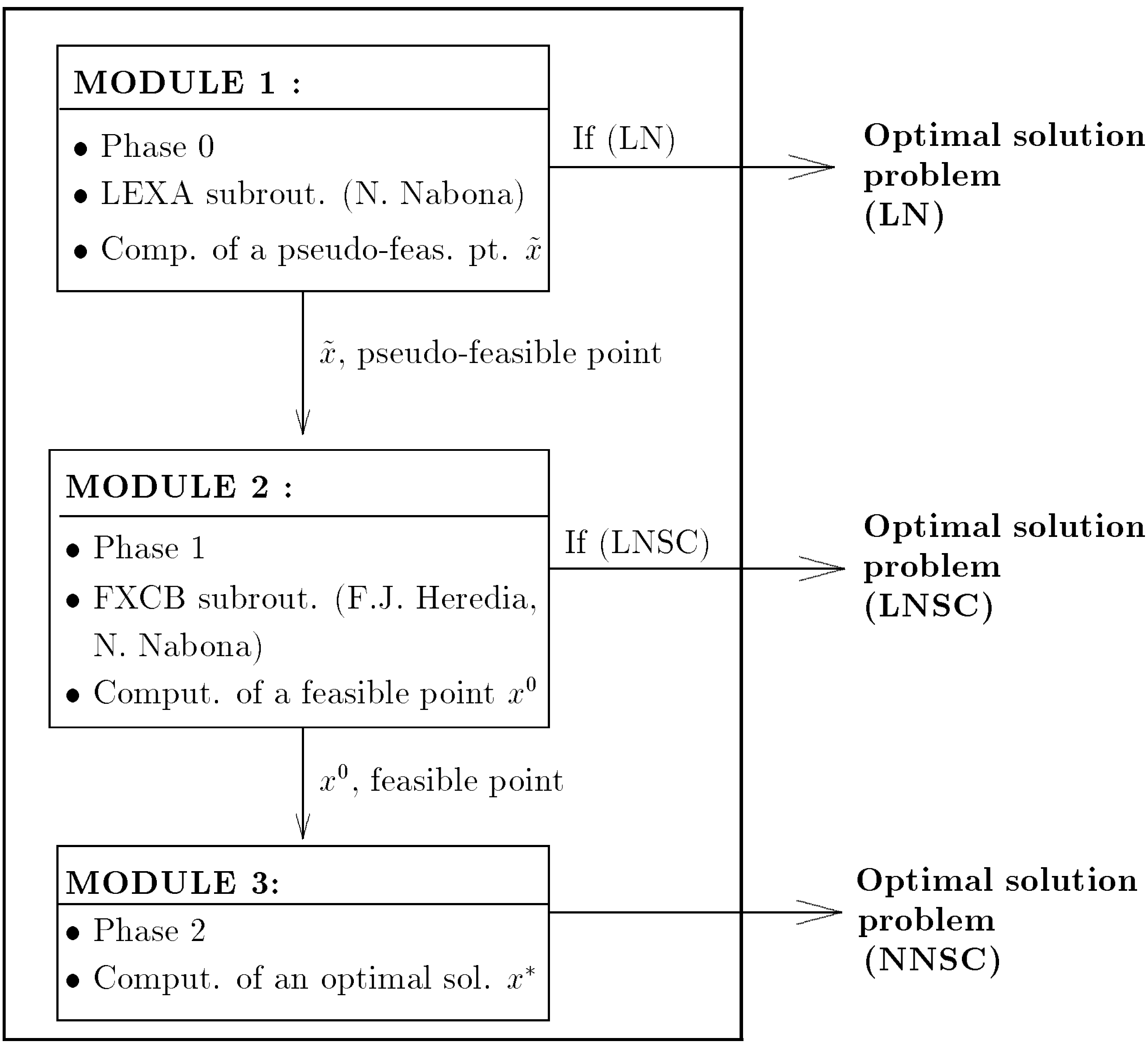
3 If $z_a^* = 0$, then $y^{*\prime} = [\, x^{*\prime} \;\; | \;\; z^{*\prime} \,]$ is a feasible solution of **(NNSC)** .

If $z_a^* > 0$, then problem **(NNSC)** is infeasible.

IMPLEMENTATION OF THE ALGORITHM: THE NOXCB 9.0 CODE

- **Developed in FORTRAN 77.**
- **NOXCB 9.0 can solve efficienly (LN), (LNSC) and (NNSC) problems.**

NOXCB



IMPLEMENTATION OF THE ALGORITHM: THE NOXCB 9.0 CODE

- **Composition of the package: fourteen source files:**

1.- `noxcb09.f` : main program necessary to use NOXCB 9.0 as an standalone program.

2.- `snoxcb09.f` : subroutine to solve the **(NNSC)** problem.

3.- `noeres05.f` : output information subroutines.

4.- `noexli11.f` : linesearch and variables update.

5.- `noline11.f` : Phase 1 (pivoting subroutines excluded).

6.- `noqnew01.f` : Quasi-Newton updates.

7.- `noauxi02.f` : subroutines to read/write data and basis.

8.- `noesnu11.f` : optimization of the **(ASR)** problem.

9.- `noetes05.f` : Computation/update of the working basis inverse/factorization.

10.- `nofas007.f` : Phase 0.

11.- `nopivo09.f` : pivoting subroutines.

12.- `NOXCB.PAR` : maximum dimensions of the data structure.

13.- `TOLCIA09.CMN` : COMMON which stores numerical tolerances.

14.- `ETES.CMN` : COMMON which stores the data structure of the working basis inverse/factorization.

COMPUTATIONAL RESULTS

- **Objectives of the computational study :**

  1.- To evaluate the computational efficiency of the NOXCB 9.0 code againts a reputed non specialized implementation of the same algorithm, MINOS 5.3.

  2.- To study the dependency of the efficiency of NOXCB 9.0 on the characteristics of the (NNSC) problems.

  3.- To evaluate the behaviour several algorithmic options.

- **Hardware :**

  * Workstation Sun Sparc 10/41 with one processor at 40MHz (≈100Mips, 20Mflops).
  * 64Mb of central memory (32Mb real, 32Mb swapped).

EIO/UPC (NNSC) test problems

- **110 (NNSC) problems.**
  - **DIMACS problems:**
    - ▷ Random generators of Network flow problems :

      Rmfgen (Goldfarb, Grigoriadis): problems *rmfa* and *rmfb*.
      Gridgen (Lee & Orlin): problems *ggeb* and *gged*.
      Grid-on-Torus (Goldberg): problems *gotd*.
    - ▷ Addition of the side constraints : *Di2no*.
    - ▷ Objective functions :

      Namur (Toint & Tuyttens).
      EIO1.
  - **Hydrothermal Coordination problems:**
- **Characteristics of the EIO/UPC test problems:**
  - 1524 arcs, 360 nodes → 23832 arcs and 6589 nodes.
  - Ratio (# side constraints / # nodes) : 1% → 100%
  - # of nonzero elements of the side constraints : 0.02% → 10%
  - # of superbasic variables at the optimal solution : 0 → 3352 (80% non basic variables).

- **DIMACS problems:**

	Di2no parameters ($seed_2$ =46533591)							Dimensions				
	$\%s.c.$	$spar$	$\%act$	t_{ij}^{max}	t_{ij}^{min}	δ_E	δ_L	n	m	t	t_E	nel
rmfa1111	1	0.010	1	1.0	0.1	1.0	1.5	1524	360	4	1	4
rmfa2111	5	0.010	1	1.0	0.1	1.0	1.5	1524	360	18	1	18
rmfa3111	10	0.010	1	1.0	0.1	1.0	1.5	1524	360	36	1	36
rmfa4111	50	0.010	1	1.0	0.1	1.0	1.5	1524	360	180	1	180
rmfa5111	100	0.010	1	1.0	0.1	1.0	1.5	1524	360	360	3	363
rmfa3211	10	1.000	1	1.0	0.1	1.0	1.5	1524	360	36	1	538
rmfa3311	10	10.000	1	1.0	0.1	1.0	1.5	1524	360	36	1	5387
rmfa3221	10	1.000	10	1.0	0.1	1.0	1.5	1524	360	36	3	538
rmfa3241	10	1.000	100	1.0	0.1	1.0	1.5	1524	360	36	36	538
rmfa3231	10	1.000	50	1.0	0.1	1.0	1.5	1524	360	36	18	538
rmfa2112	5	0.010	1	10^5	0.1	1.0	1.5	1524	360	18	1	18
rmfa3132	10	0.010	50	10^5	0.1	1.0	1.5	1524	360	36	18	36
rmfa3232	10	1.000	50	10^5	0.1	1.0	1.5	1524	360	36	18	538
rmfb1111	1	0.010	1	1.0	0.1	1.0	1.5	5420	1200	12	1	13
rmfb3111	10	0.010	1	1.0	0.1	1.0	1.5	5420	1200	120	1	138
rmfb4111	50	0.010	1	1.0	0.1	1.0	1.5	5420	1200	600	6	658
rmfb5111	100	0.010	1	1.0	0.1	1.0	1.5	5420	1200	1200	12	1338
rmfb3211	10	1.000	1	1.0	0.1	1.0	1.5	5420	1200	120	1	6492
rmfb3311	10	5.000	1	1.0	0.1	1.0	1.5	5420	1200	120	1	32360
rmfb3221	10	1.000	10	1.0	0.1	1.0	1.5	5420	1200	120	12	138
rmfb3132	10	0.010	50	10^6	0.1	1.0	1.5	5420	1200	120	60	6491
ggeb1121	1	0.100	10	1.0	0.1	100.0	100.0	4008	501	5	1	20
ggeb3121	10	0.100	10	1.0	0.1	10.0	100.0	4008	501	50	5	198
ggeb4121	50	0.100	10	1.0	0.1	10.0	10.0	4008	501	251	25	1013
ggeb5121	100	0.100	10	1.0	0.1	1.5	1.5	4008	501	501	50	2040
ggeb4221	50	1.000	10	1.0	0.1	10.0	10.0	4008	501	251	25	9973
ggeb4321	50	5.000	10	1.0	0.1	10.0	10.0	4008	501	251	25	50203
ggeb4131	50	0.100	50	1.0	0.1	1.0	1.0	4008	501	251	125	1013
ggeb4141	50	0.100	100	1.0	0.1	1.0	1.0	4008	501	251	251	1013
ggeb4132	50	0.100	50	10^5	0.1	1.0	1.0	4008	501	251	125	1013

- **DIMACS problems (continuation):**

	Di2no parameters ($seed_2$ =46533591)							Dimensions				
	$\%s.c.$	$spar$	$\%act$	t_{ij}^{max}	t_{ij}^{min}	δ_E	δ_L	n	m	t	t_E	nel
gged1121	1	0.100	10	1.0	0.1	100.0	100.0	12008	1501	15	1	182
gged3121	10	0.100	10	1.0	0.1	10.0	100.0	12008	1501	150	15	1834
gged4121	50	0.100	10	1.0	0.1	1.0	1.0	12008	1501	751	75	9054
gged4131	50	0.100	50	1.0	0.1	1.0	1.0	12008	1501	751	375	9054
gged4141	50	0.100	100	1.0	0.1	1.0	1.0	12008	1501	751	751	9054
gged4122	50	0.100	10	100.	0.1	1.0	1.0	12008	1501	751	75	9054
gotd1121	1	0.100	10	1.0	0.1	1.0	1.0	18000	3000	30	3	527
gotd2121	10	0.100	10	1.0	0.1	1.0	1.0	18000	3000	300	30	5390
gotd3121	25	0.100	10	1.0	0.1	1.0	1.0	18000	3000	750	75	13537
gotd2131	10	0.100	50	1.0	0.1	1.0	1.0	18000	3000	300	150	5390

- **Hydrothermal problems:**

	n	m	t	nel	$\%s.c.$	$spar$
xh48	1152	289	144	3154	49.8	1.901
xha48	1824	433	144	4732	33.3	1.802
xhs40	4176	961	144	9187	15.0	1.528
xh168	4032	1009	504	11074	50.0	0.545
xha168	6384	1513	504	16612	33.3	0.516
xhs50	14616	3361	504	32227	15.0	0.437

	n	m	t	nel	$\%s.c.$	$spar$
xa48	2256	697	240	1915	34.4	0.354
xa168	8064	2479	840	6799	33.9	0.100
xaa48	4416	1345	528	4601	39.3	0.197
xaa168	15600	4741	1848	16193	39.0	0.056
xas48	6768	1873	528	6086	28.2	0.170
xas168	23832	6589	1848	21398	28.0	0.049

- **Objective functions.**

  * **Namur** (Toint & Tuyttens, (MP) 1990):

$$f(x) = \frac{1}{c_1}\sum_{i=1}^{n} x_i^2 + \\ +\frac{1}{c_2}\left(\sum_{i=1}^{n-1}\sqrt{1+x_i^2+(x_i-x_{i+1})^2}+\frac{1}{c_3}\left(10+\sum_{i=1}^{n}(-1)^i x_i\right)^4\right)$$

NAME	c_1	c_2	c_3	Problems
10	10^3	10^3	$1.2\cdot10^3$	rmfa, rmfb
13	1.	10^5	$1.2\cdot10^5$	ggeb

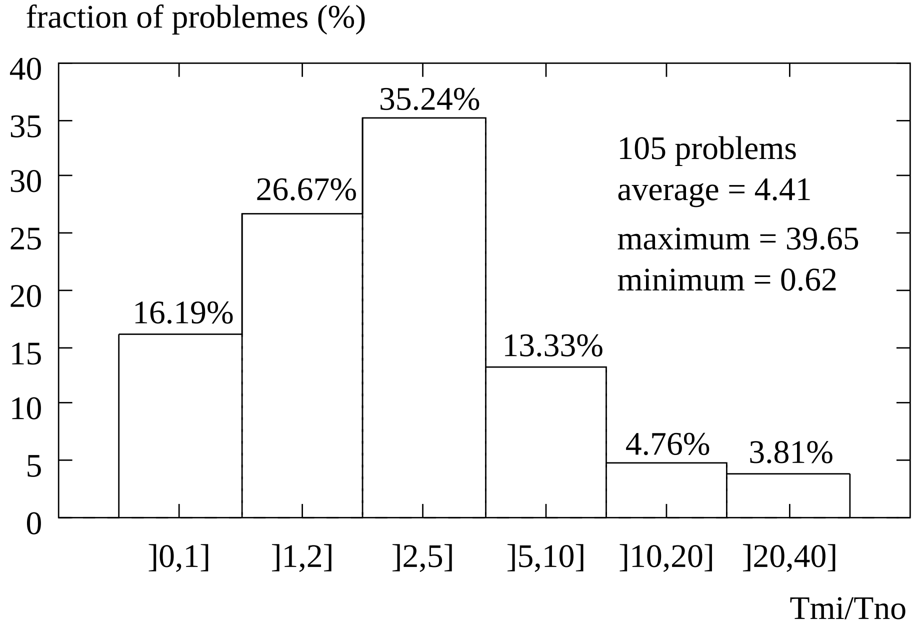
  * **EIO1:**

$$f(x) = k_1\left\{\sum_{i=1}^{n} c_i\left(x_i+k_2x_i^2\right)+ \\ +k_3\left[\sum_{i=1}^{n-2} c_i\left(x_ix_{i+1}x_{i+2}\right)^2+c_{n-1}\left(x_{n-1}x_n\right)^2\right]\right\}$$

NAME	k_1	k_2	k_3	Problems
20	1.	0.	0.	rmfa, rmfb
21	0.01	0.01	0.001	ggeb, gotd
22	0.01	0.01	0.	rmfa, rmfb, ggeb, gged

GENERAL RESULTS

- **Hystogram of the efficiency of NO vs. MI, total execution time :**



- **Efficiency index :**

  t_{MI}/t_{NO} =execution time of MINOS / execution time for NOXCB

  (p.ex.: $t_{MI}/t_{NO} = 5 \Rightarrow$ NOXCB 9.0 is five times faster than MINOS 5.3, or NOXCB 9.0 reduces the execution time in a 80%).

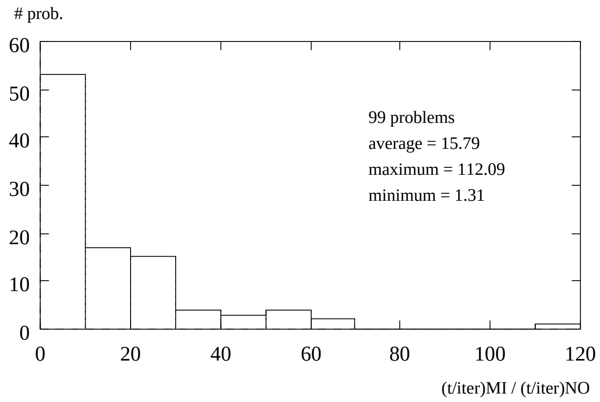
- **61.91% of the test results with** $t_{MI}/t_{NO} \in [1,5]$.

- **17 problems with** $t_{MI}/t_{NO} < 1$ **:**

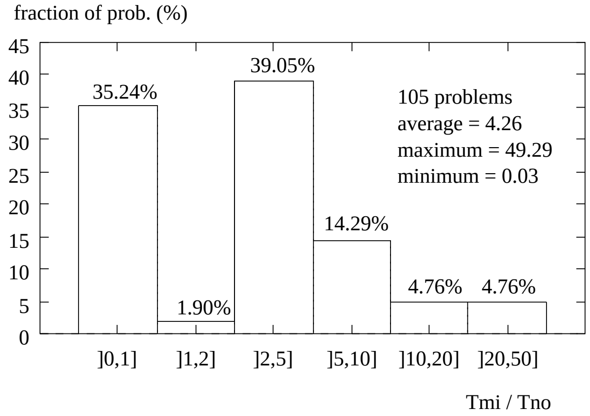
  * Usually small scale models (13 problems *rmfa*, 1524 arcs, 360 nodes).
  * The reduction in the execution time (aprox. 20%) is not so important than in the cases where $t_{MI}/t_{NO} > 1$ (aprox. 80%).

Results for the different Phases

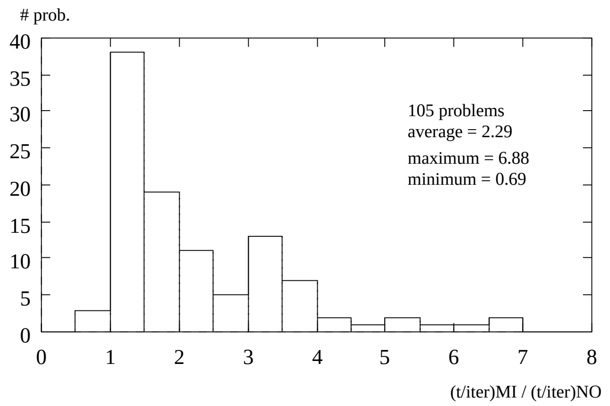
- **Comparative MI vs. NO Phases 0+1 and Phase 2 :**



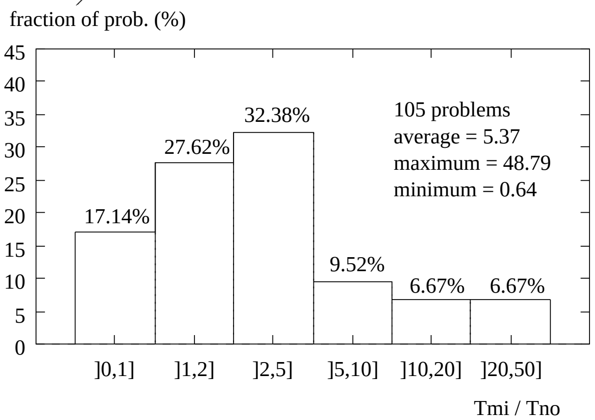
a) ratio for the iteration time, Phase 0+1



b) ratio for the execution time Phase 0+1



c) ratio for the iteration time, Phase 2



d) ratio for the execution time, Phase 2

- **Comparation MI vs. NO Phases 0+1 and Phase 2 :**

  - $*$ Average efficiency for Phase 0+1 :

$$t_{MI}/t_{NO} = 4.26 < \frac{(t/iter)_{MI}}{(t/iter)_{NO}} = 15.79$$

  - $*$ Average efficiency for Phase 2 :

$$t_{MI}/t_{NO} = 5.37 > \frac{(t/iter)_{MI}}{(t/iter)_{NO}} = 2.29$$

Results for the different problems.

- **Efficiency of the total execution time for the different problems and o.f. :**

Problems	t_{MI}/t_{NO}	Objective functions 10	13	20	21	22	Results for problems
rmfa	Average	1.49		9.01		1.01	**3.70**
	Maximum	2.23		20.41		2.16	**20.41**
	Minimum	0.72		4.72		0.62	**0.62**
rmfb	Average	3.12		15.92		2.97	**8.18**
	Maximum	7.25		39.65		4.97	**39.65**
	Minimum	0.92		3.11		0.95	**0.92**
ggeb	Average		4.99		1.85	2.03	**2.96**
	Maximum		10.01		2.08	3.34	**10.01**
	Minimum		1.60		1.50	1.56	**1.50**
gged	Average					1.68	**1.68**
	Maximum					2.44	**2.44**
	Minimum					0.84	**0.84**
gotd	Average				10.43		**10.43**
	Maximum				17.96		**17.96**
	Minimum				6.56		**6.56**
xh	Average						**4.25**
	Maximum						**12.81**
	Minimum						**1.31**
xa	Average						**2.36**
	Maximum						**3.77**
	Minimum						**1.53**
Results for objective functions	Average	**1.87**	**4.99**	**11.78**	**4.00**	**1.81**	
	Maximum	**7.25**	**10.01**	**39.65**	**17.96**	**4.97**	
	Minimum	**0.72**	**1.60**	**3.11**	**1.50**	**0.62**	

- **Higest degree of efficiency: problems** *gotd*.
  - ∗ 18000 arcs, 3000 nodes, up to 750 side constraints
  - ∗ f.o. EIO1.
  - ∗ High degree of degeneration.
- **Higest efficiency for nonlinear obj. func. : 13**
  - ∗ *Namur* (high number of superbasic variables).
- **Lowest efficiency : problems** *rmfa* **(small scale problems).**

Results for the different problems: conclusions.

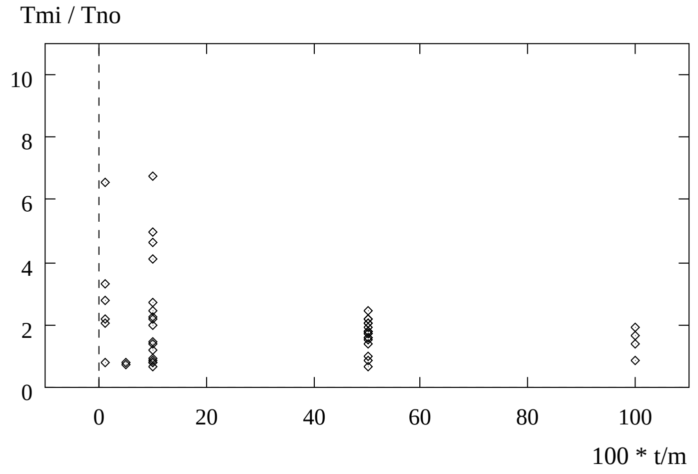
- **Results depending on the characteristics of the problems:**
  - ∗ *The greatest efficiency is reached with large escale models.*

Scale	**Maximal dimensions** arcs	nodes	s.c.	t_{MI}/t_{NO} 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

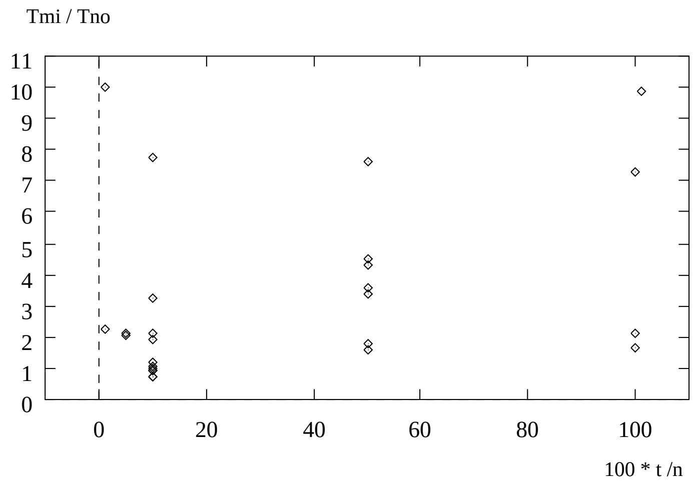
- **Results depending on the objective function:**
  - ∗ *The greatest efficiency is reached with the Namur objective function family (high number of superbasic variables).*

Number of side constraints

- **Relation between efficiency and the ratio (# of s. c. /# network equations)=(t/m) :**



a) o.f. EIO1



b) o.f. Namur

∗ *Conclusions :*

▷ *Objective function EIO1 :* $t/m \uparrow \Rightarrow t_{MI}/t_{NO} \downarrow$

▷ *Objective function Namur : no correlation.*

∗ *High efficiency even with problems with great ratio* t/m*:*

▷ *Problems with* $t/m = 1.$ *:* $t_{MI}/t_{NO} = 2.79$ *in average.*

Side constraints structure

- **Relation between the minimum (t_{ij}^{min}) and maximum (t_{ij}^{max}) coeficient of the side constraints:**
  - ∗ Reference problems: $t_{ij}^{min} = 0.1$, $t_{ij}^{max} = 1$.
  - ∗ Modified problems: $t_{ij}^{min} = 0.1$, $t_{ij}^{max} \in \{10^5, 10^6, 100\}$
  - ∗ Efficiency according to the value of the elements of the side constraints :

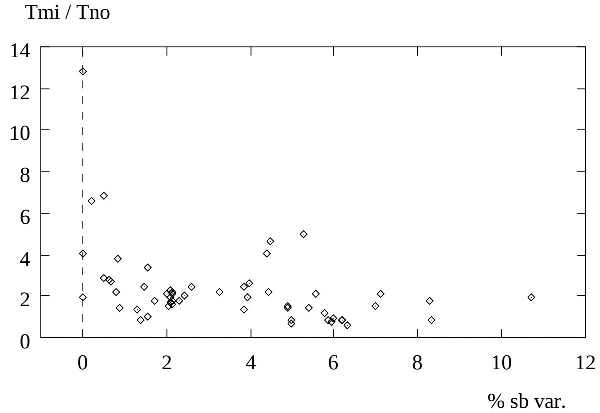
problem	Average for large t_{ij}^{max}	Plain average	# cases/total
rmfa	3.17	3.84	8/30
rmfb	3.94	8.65	2/18
ggeb	3.80	2.8	3/24
gged	0.84	1.85	1/5

- **Sparsity :**
  - ∗ Density : $nel/(n \times t)$:
  - ∗ Dependency between the efficiency and the sparsity of the side constraints:

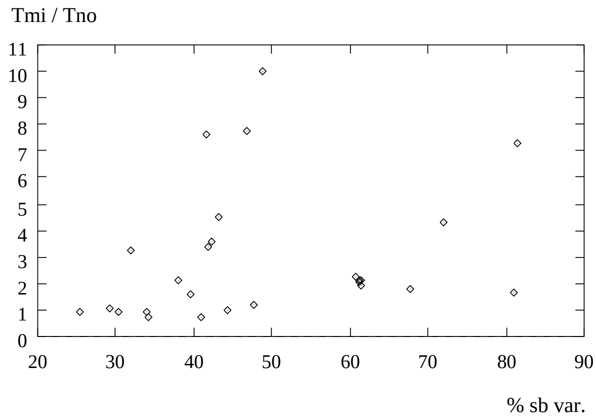
$100 \times nel/(n \times t)$	Average	# instances
≤ 0.1	3.24	30
$\approx 1.$	2.32	45
$\approx 5.$	2.89	8
$\approx 10.$	1.17	2

Number of superbasic variables

- **Fraction of the s.v.b. at the optimal solution :**
  $$\%s^* = 100 \times s^*/(n-m-t).$$
- **Relation $\%s^*$ - efficiency :**



a) o.f. EIO1, xh i xa



b) f.o. Namur

$*$ *Behaviour depending on the value of $\%s^*$:*

- ▷ *Objective function EIO1 :* $\%s^* \uparrow \Rightarrow t_{MI}/t_{NO} \downarrow$
- ▷ *Objective function Namur : no apparent correlation.*

- **Comparative between problems of the same family with different $\%s^*$:**

	Average $\%s^*$	Average $(t_2/it_2)\frac{\text{MI}}{\text{NO}}$	Average $t_{2_{\text{MI}}}/t_{2_{\text{NO}}}$
10rmfa	52.36	1.71	1.48
22rmfa	5.70	1.45	0.98
10rmfb	43.29	2.90	3.08
22rmfb	2.88	1.99	2.79
13ggeb	46.02	4.42	5.00
21ggeb	3.48	1.61	1.78
22ggeb	3.06	1.64	1.96

$*$ *The efficiency increases with problems with large $\%s^*$.*

EVALUATION OF THE ALGORITHMIC VARIANTS

- **Algorithmic variants studied:**
  - ∗ Bertsekas's linesearch.
  - ∗ Blocking control.
  - ∗ Key and superbasic cicles.
  - ∗ Truncated Newton method vs. quasi Newton.
  - ∗ Working basis factorization.
- **EIO/UPC10 test problems:**

		n	m	t	nel	$\%s.c.$	$spar.$	$\%c_x^*$	$\%s^*$
$\mathbf{P}_1$	**10 rmfa2111**	1524	360	18	18	5.0	0.066	0.56	61.7
$\mathbf{P}_2$	**22 rmfa3241**	1524	360	36	538	10.0	0.981	10.00	4.43
$\mathbf{P}_3$	**22 rmfb3132**	5420	1200	120	138	10.0	0.021	1.25	6.00
$\mathbf{P}_4$	**10 rmfb5111**	5420	1200	1200	1338	100.0	0.021	0.50	79.00
$\mathbf{P}_5$	**21 ggeb3121**	4008	501	50	198	10.0	0.099	2.79	2.43
$\mathbf{P}_6$	**13 ggeb4141**	4008	501	251	1013	50.1	0.101	47.11	41.89
$\mathbf{P}_7$	**22 gged3121**	12008	1501	150	1834	10.0	0.100	2.86	2.10
$\mathbf{P}_8$	**21 gotd2121**	18000	3000	300	5390	10.0	0.100	7.10	0.54
$\mathbf{P}_9$	**11 xha168**	6384	1513	504	16612	33.3	0.516	0.00	3.85
$\mathbf{P}_{10}$	**12 xaa168**	15600	4741	1848	16193	39.0	0.056	32.67	3.92

EVALUATION OF THE ALGORITHMIC VARIANTS Outstanding results.

- **Bertsekas's linesearch.**
  - ∗ *Both the total execution time and the iteration time is worsened because of the increase in the number of iteration needed to solve the subproblems* $(\mathbf{ASR})^{k}$.
- **Blocking control.**
  - ∗ *There are two opposite effects that mutually cancels: an increase in the iteration time and a decrease in the number of iterations.*
- **Key and superbasic cicles (Θ_c , Θ_s).**
  - ∗ *Problems with a large number of superbasic variables (Namur o.f.) worsen clearly when the cicle structures are used explicitly.*
- **Truncated Newton method (TNM)vs. quasi Newton method (QNM).**
  - ∗ *The number of iterations needed to solve subproblems* $(\mathbf{ASR})^{k}$ *is decreased when the (TNM) is used.*
  - ∗ *The number of objective function and gradient evaluations experiments a very significant increase with the truncated Newton method. This fact produces an increase in the total execution time.*
- **Working basis factorization.**
  - ∗ *The product form of the inverse reduces a 6% in average the total execution time, but the LU factorization is more robust.*