Constrained nonlinear network flow problems through projected Lagrangian methods

F. Javier Heredia Cervera
Dept. of Statistics and Operations Reseach, Univ. Politècnica de Catalunya
Pau Gargallo 5, 08028 Barcelona, Spain. heredia@eio.upc.es

Abstract: Recent numerical experiments show that the resolution of the Nonlinear Network flow problem with side Constraints **(NNC)** can be significantly sped up, when the side constraints are linear, by specialised codes based on a conjunction of basis partitioning techniques and active set methods. A natural extension of these methods is that of including them in a Projected Lagrangian Algorithm **PLA**. A specialised **PLA** will solve the general **(NNC)** problem through the optimization of a sequence of **(NNC)**s with linearized side constraints, taking advantage of the efficiency of the linear side constraint codes. The description of this methodology will be presented together with the numerical results obtained from the application of this technique to a set of hydrothermal scheduling problems.

Key-words: nonlinear network flows, side constraints, projected Lagrangian methods, basis partitioning, hydrothermal scheduling.

1 Introduction

1.1 Definition of the problem

The standard form of the *Nonlinear Network flow problem with side Constraints* **(NNC)** could be defined as the following optimization problem:

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{1}$$

s.t.:

$$Ax = r \tag{2}$$
$$c(x) + \mathrm{I}_n z_n = b_n \tag{3}$$
$$Tx + \mathrm{I}_l z_l = b_l \tag{4}$$
$$0 \leq x \leq u_x \tag{5}$$
$$0 \leq z_n \leq u_{zn} \tag{6}$$
$$0 \leq z_l \leq u_{zl} \tag{7}$$

where:

(1) $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is nonlinear and twice continuously differentiable over the feasible set defined by constraints (2-7).

(2) these are the network equations, where $A \in \mathbb{R}^{m\times n}$ is the full rank node-arc incidence matrix, and $r \in \mathbb{R}^m$ the injection/consumption vector.

(3) these equality *nonlinear side constraints* stand for any set of generally bounded nonlinear constraints $\underline{b_n} \leq c(x) \leq b_n$. The functions $c\colon \mathbb{R}^n \rightarrow \mathbb{R}^{t_n}$ are usually assumed to be twice continuously differentiable, although for practical purposes they only need to be differentiable and continuous. Vector $b_n \in \mathbb{R}^{t_n}$ is the right-hand side and $\mathrm{I}_n \in \mathbb{R}^{t_n\times\tilde{t}_n}$ is the coefficient matrix of the slacks $z_n \in \mathbb{R}^{\tilde{t}_n}$.

(4) these are the standard form of a set of t_l general *linear side constraints* $\underline{b_l} \leq Tx \leq b_l$, with $T \in \mathbb{R}^{t_l\times n}$ and the RHS vector $b_l \in \mathbb{R}^{t_l}$. $\mathrm{I}_l \in \mathbb{R}^{t_l\times\tilde{t}_l}$ is the coefficient matrix of the slacks $z_l \in \mathbb{R}^{\tilde{t}_l}$.

(5-7) $u_x \in \mathbb{R}^n$, $u_{zl} \in \mathbb{R}^{\tilde{t}_l}$ and $u_{zn} \in \mathbb{R}^{\tilde{t}_n}$ are, respectively, the upper bounds to the real variables $x \in \mathbb{R}^n$ and to the slack variables z_n and z_l used to transform the general inequality linear and nonlinear side constraint into the standard equality form.

1.2 Motivation: the Short-Term Hydrothermal Scheduling problem

One of the most relevant applications of the **(NNC)** problems is the *Short-Term Hydrothermal Scheduling* **(STHS)**problem. **(STHS)** is among the most important problems to be solved in the management of a power utility when hydroelectric plants are a part of the power system. The solution sought indicates how to distribute the hydroelectric generation (cost-free) in each reservoir of the reservoir system and how to allocate generation to thermal units committed to operating over a short period of time (e.g. two days) so that the fuel expenditure during the period is minimized. In **(STHS)** the predicted load at each hourly interval must be met, and a spinning reserve requirement to account for failures or load prediction errors must be satisfied. These load and spinning reserve constraints take in both hydro and thermal generation. As usual, the short-term period (of 24 to 168 hours) is subdivided into smaller time intervals (of 1 to 4 hours) for which data are determined and variables are optimized.

A great variety of formulations of the **(STHS)** problems can be found in the literature of the power system field. In this paper we will focus our attention onn the so–called *Coupled Model* proposed by Heredia and Nabona in [1]. The most relevant characteristic of this model is that all the variables taking part in the **(STHS)** problem are flows on arcs of a single network such as that in Figure 1, called the *Hydro-Thermal-Transmission Extended Network (HTTEN)*. The HTTEN is build with the well–known hydro replicated network, through which the temporal evolution of the reservoir system is usually modelled. In Figure 1, variables d_k^i and s_k^i stand respectively for the discharge and spillage of reservoir k at time interval i, variable $v_k^{(i-1}$ is the volume of reservoir k at the beginning of the i^{th} interval and variable v_k^i represents the volume of the same reservoir at the end of the interval, after releasing the discharge d_k^i and the spill s_k^i.

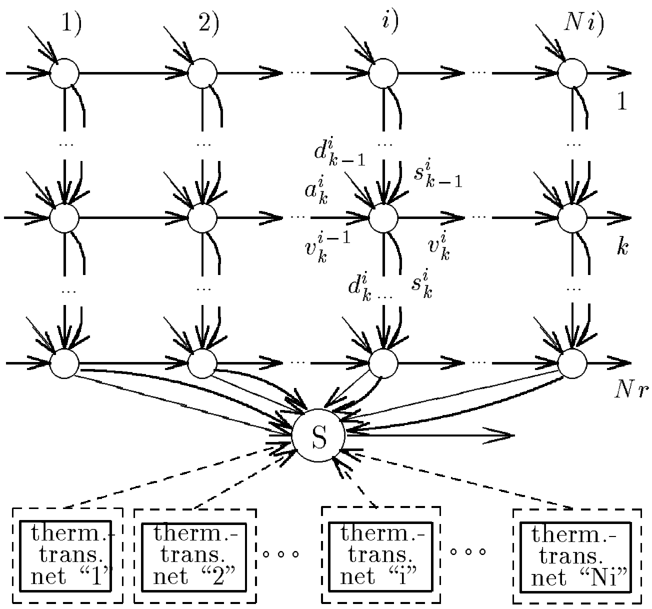

Figure 1: Hydro-Thermal-Transmission Extended Network (HTTEN).

The *thermal and transmission network* (*therm–trans net "i"* in Figure 1) is the graph which models the thermal units and the transmission network of the generation system. This graph is based on the concept of the *equivalent thermal network for thermal unit "j"* (Figure 2), a network with four arcs (r_{Ij}, r_{Dj}, g_{Ij} and g_{Dj}) and one node (A), which could be used to model the output generation and the incremental and decremental spinning reserve of thermal unit "j". Variables r_{Ij} and r_{Dj} in Figure 2 are the incremental and decremental spinning reserve of thermal unit "j" respectively. The auxiliary variables g_{Ij} and g_{Dj} are the so–called incremental and decremental "gaps", and are introduced into the model to transform the linear inequalities defining the spinning reserve into network equations. The set of equivalent thermal networks for each thermal unit at each interval are linked to the transmission network and to the sink node, through the bus node "B" and "S", respectively, in Figure 2. See reference [1] for a more detailed explanation of all the above.

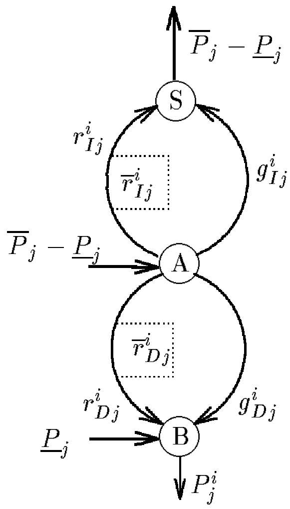

Figure 2: Equivalent thermal network of thermal unit "j".

The coupled network formulation of the short-term hydrothermal coordination problem considers an objective function which is the sum of the thermal generation costs plus an economic evaluation of the transportation losses. The generation costs are taken as a quadratic function of the thermal generation P_j^i which, in terms of the variables of the equivalent thermal network, could be expressed as $P_j^i = r_{Dj} + g_{Dj} + \underline{P}_j$ (balance equation of node "B" in Figure 2). The expression of the objective function is:

$$\min \sum_{i=1}^{Ni} \Big\{ \sum_{j=1}^{Nu} [c_{lj}(r_{Dj} + g_{Dj} + \underline{P}_j) + c_{qj}(r_{Dj} + g_{Dj} + \underline{P}_j)^2] + \pi^i \sum_{l=1}^{Nm} r_l (q_l^i)^2 \Big\} \quad (8)$$

where c_{lj} c_{qj} are the linear and quadratic coefficients of the thermal generation cost function, π^i is the cost per MWh of losses, and r_l and q_l^i are the reactance and the power flow, respectively, of the transmission line l.

The first constraints to be satisfied are the flow balance equations of the HTTEN, which could be expressed as:

$$A_{HTTEN} \begin{bmatrix} d \\ v \\ s \\ r_D \\ g_D \\ r_I \\ g_I \\ g \\ q \end{bmatrix} = r_{HTTEN} \quad (9)$$

where the only set of variables not yet defined is $g = [g_1^1, \ldots, g_{Nr}^{Ni}]'$. There is also a set of linear and nonlinear side constraints that must be imposed in order to couple the hydro and thermal systems. As the transmission network is included in the HTTEN, there is no need for a specific load constraint because the network balance equations of the thermal and transmission network ensure the satisfaction of the load at each interval. The following hydrogeneration nonlinear side constraints define variables g_j^i:

$$g_j^i = \sum_{k \in I_j} h_k(d_k^i, v_k^i, v_k^{i-1}) \quad j = 1, \ldots, Ng \; ; \; i = 1, \ldots, Ni \quad (10)$$

In this relation g_j^i is the arc which feeds the generation of the reservoir k at interval i into the corresponding generation bus of the transmission network. The nonlinear (and nonconvex) functions h_k on the right–hand side of (10) are the expression of the *hydrogeneration* as a function of the discharge ${d_k}^i$ and the initial and final volume v_k^i, v_k^{i-1}. More details of this expression can be obtained from the paper [1].

The satisfaction of the incremental and decremental spinning reserve requirements at each interval (R_I^i and R_D^i) are imposed through the following set of constraints:

$$-\sum_{k=1}^{Nr} g_j^i + \sum_{j=1}^{Nu} r_{Ij}^i \geq R_I^i - \sum_{k=1}^{Nr} \overline{H}_k^i \quad (11)$$

$$\sum_{k=1}^{Nr} g_j^i + \sum_{j=1}^{Nu} r_{Dj}^i \geq R_D^i \quad (12)$$

$$i = 1, \ldots, Ni$$

where $\overline{H}_k^i$ is the maximal hydrogeneration of reservoir k at interval i. Finally, to complete the d.c. approach to the transmission network, Kirchhoff's voltage law is imposed through the following set of linear side constraints:

$$\sum_{(k,l) \in \text{loop}j} x_{kl} p_{kl}^i = 0 \text{ for all basic loops} j \; ; \; i = 1, \ldots, Ni \quad (13)$$

The minimization of (8) subject to constraints (9–13), plus the necessary upper and lower bounds to the variables, constitutes the Coupled Model of the **(STHS)** problem. As the reader can verify, it is an **(NNC)** problem. The section below presents a method for solving this problem based on specialised network flow projected Lagrangian techniques.

2 The projected Lagrangian approach

Projected Lagrangian methods are well known in the optimization literature. One of the most successful general purpose nonlinear optimization packages, MINOS [3], is based on this methodology. Roughly speaking, the projected Lagrangian method solves nonlinear constrained problems by successively solving a set of subproblems where the objective function is somehow related with the Lagrangian function and the constraints are formulated as a linearization of the original nonlinear constraints. A projected Lagrangian based algorithm applied to the solution of **(NNC)** problem (1-7) goes through the following steps:

PLA : Projected Lagrangian algorithm for problem (NNC)

1. Select some starting values for x^0 and λ^0 (*Lagrange multipliers estimate*). Set $k := 0$.
2. **Major iteration:** If x^k is the optimal solution of **(NNC)** then STOP. Otherwise, proceed.
   (a) Linearize the nonlinear side constraints at x^k:
   $$c(x) \approx c^k(x) = c(x^k) + \nabla c(x^k)(x - x^k) \quad (14)$$
   (b) **Minor iterations:** solve the linearized subproblem $\mathbf{(NNLC)}^k$:
   $$\min_{x \in \mathbb{R}^n} \quad \Phi^k(x) \quad (15)$$
   $$\text{s.t.:} \quad (16)$$
   $$Ax = r \quad (17)$$
   $$\nabla c(x^k)x + \mathrm{I}_n z_n = b_n^k \quad (18)$$
   $$Tx + \mathrm{I}_l z_l = b_l \quad (19)$$
   $$0 \leq x \leq u_x \quad (20)$$
   $$0 \leq z_n \leq u_{zl} \quad (21)$$
   $$0 \leq z_l \leq u_{zn} \quad (22)$$
   where $b_n^k = b_n - c(x^k) + \nabla c(x^k)x^k$ and $\Phi^k(x)$ is the *merit function*, which should be equal to, or at least related with, the *Lagrangian function* $\mathcal{L}(x, \lambda^k, x^k) = f(x) - \lambda^{k'}\left(c(x) - c^k(x)\right)$. Let $[x^*]^k$ be the optimal solution of problem $\mathbf{(NNLC)}^k$ and $[\lambda^*]^k$ the Lagrange multipliers of constraints (18) at the optimal solution $[x^*]^k$.

(c) Update x^k and λ^k somehow from the solution $[x^*]^k$ and $[\lambda^*]^k$. Set $k := k + 1$ and go to step 2.

Robinson [4] proved the q-quadratically local convergence of this method if the merit function is defined as

$$\Phi^k(x) = f(x) - \lambda^{k'}\left(c(x) - c^k(x)\right) \quad (23)$$

and the vectors x^{k+1} and λ^{k+1} for the new major iteration are taken respectively as $[x^*]^k$ and $[\lambda^*]^k$. Moreover, the Robinson's method lacks global convergence. To overcome this problem, Murtagh and Saunders proposed in [5] a projected Lagrangian method based on Robinson's procedure where the merit function is defined as a *modified augmented Lagrangian function*:

$$\Phi^k(x; x^k, \lambda^k, \rho^k) = f(x) - \lambda^{k'}\left(c(x) - c^k(x)\right) + \\ + \frac{\rho^k}{2}\|c(x) - c^k(x)\|_2 \quad (24)$$

Murtagh and Saunders's package MINOS [3], which implements this methodology, is probably considered as the best of the general purpose nonlinear optimization packages. It must be emphasised that problem $\mathbf{(NNLC)}^k$ defined by equations (15-22) is nothing but a nonlinear network flow problem with *linear* side constraints, which could be efficiently solved through the specialised nonlinear network flow techniques described in the section below.

2.1 Optimization of the linearized subproblems $(\mathrm{NNLC})^k$

Package `noxcb09` [2] is an optimization code for solving the nonlinear network flow problem with linear side constraints **(NNLC)** defined as:

$$\min_{x \in \mathbb{R}^n} \quad f(x) \quad (25)$$

$$\text{s.t.:} \quad (26)$$

$$Ax \quad = r \quad (27)$$

$$Tx \quad + \mathrm{I}_l z_l \quad = b_l \quad (28)$$

$$0 \le x \quad \le u_x \quad (29)$$

$$0 \le z_l \quad \le u_{zl} \quad (30)$$

This program consists of a specialized implementation of the active set strategy proposed by Murtagh and Saunders in [6]. This algorithm could be expressed as follows:

ASA: active set algorithm.

1. **Initialization.**

   (a) Find an initial feasible solution x^0. Initialize the numerical tolerances $\epsilon_o > 0$ and $\epsilon_G > 0$.

   (b) Define the *basic* $\mathcal{B}^0$, *superbasic* $\mathcal{S}^0$ and *nonbasic* $\mathcal{N}^0$ sets of variables associated with the current active set at x^0.

   (c) Compute the *null space basis* Z^0, the *reduced gradient* ${g_z}^0 = Z^{0'}\nabla f(x^0)$ and the Lagrange multipliers $\sigma^{0'} = \nabla_{\mathcal{N}} f(x^0) - \nabla_{\mathcal{B}} f(x^0) B^{-1} N$ of the nonbasic variables. Set $k := 0$.

2. **Do While** $\sigma^k < -\epsilon_o$.

   (a) Select a nonbasic variable $q \in \mathcal{N}_0^k \cup \mathcal{N}_u^k$ such that $\sigma_q^k < 0$ and relax its active bound.

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

   (c) **Do While** $\|{g_z}^k\| > \epsilon_G$

      i. Compute a feasible descent direction:

         A. Compute the positive definite approximation $\tilde{H}_z^k \approx Z^{k'}\nabla^2 f(x^k) Z^k$

         B. Find ${p_z}^k$ by solving $\tilde{H}_z^k {p_z}^k = -{g_z}^k$

         C. Compute $p^k := Z^k {p_z}^k$

      ii. Find the maximal step length:

      $$\bar{\alpha} = \min\{\bar{\alpha}_{\mathcal{B}}, \bar{\alpha}_{\mathcal{S}}\}$$

      iii. Linesearch:

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

      iv. Update the current solution:

      $$x^{k+1} := x^k + {\alpha^*}^k p^k$$

      v. If ${\alpha^*}^k = \bar{\alpha}$, then update $\mathcal{B}^k$, $\mathcal{S}^k$, $\mathcal{N}^k$

      vi. Update Z^k and ${g_z}^k$. $k := k + 1$

      End Do

   (d) Compute the Lagrange multiplier estimates ${\sigma_0}^k$, ${\sigma_u}^k$.

   End Do

The efficiency of this algorithm depends both on the cost of each iteration and on the number of iterations. The cost per iteration is usually dominated by the cost of solving systems of equations with the basic matrix B, while the number of iterations depends mainly on the quality of the descent direction, the efficiency of the linesearch and the convergence criteria, which, in the algorithm above, has been stated in the simplifying form $\|{g_z}^k\| \le {\epsilon_G}^k$, ${\sigma_0}^k \ge -\epsilon_o$ and ${\sigma_u}^k \le \epsilon_o$, but is actually far more complex.

The network structure of problem **(NNLC)** could be exploited through the so-called *basis partitioning* method to reduce the cost of each iteration, speeding up the solution of systems of equations involving the basic matrix B. The basis partitioning technique [7] is based on the fact that each basis of problem **(NNLC)** could be expressed as:

$$B = \left[\begin{array}{c|c} A_{\mathcal{A}} & A_{\mathcal{C}} \vdots 0 \\ \hline T_{\mathcal{A}} & T_{\mathcal{C}} \vdots \mathrm{I}_{\mathcal{C}} \end{array} \right] \tag{31}$$

where $A_{\mathcal{A}} \in \mathbb{R}^{m\times m}$ is a basis of the network matrix A (*spanning tree*), $A_{\mathcal{C}} \in \mathbb{R}^{m\times c_x}$ are the columns of matrix A related with the basic arcs not belonging to the spanning tree (*non-key arcs*), and the last columns of B correspond to the basic slacks z_l. Matrices $T_{\mathcal{A}} \in \mathbb{R}^{t_l\times m}$ and $T_{\mathcal{C}} \in \mathbb{R}^{t_l\times c_x}$ are the basic columns of matrix T, while matrix $\mathrm{I}_{\mathcal{C}}$ is built with the basic columns of matrix I_l. The key idea for the exploitation of the network structure of problem **(NNLC)** is that, thanks to expression (31), it can be shown [7] that the systems of equations $Bv = w$ and $v'B = w'$ can be computed by solving *(a)* systems of equations with the matrix $A_{\mathcal{A}}$, whose solution could be efficiently found through network flow techniques, and *(b)* systems of equations with the general (dense) matrix $Q = [T_{\mathcal{C}}|\mathrm{I}_{\mathcal{C}}] - T_{\mathcal{A}} A_{\mathcal{A}}^{-1}[A_{\mathcal{C}}|\mathbf{0}] \in \mathbb{R}^{t_l\times t_l}$. Matrix Q is called the *working basis* and is simply the Schurz complement of the matrix $A_{\mathcal{A}}$ in B.

To improve the convergence of this algorithm (number of iteration to achieve the optimal solution), special care was paid to the computation of the descent direction (step 2(c)i), the linesearch (step 2(c)iii), the working basis update (step 2(c)vi), and the convergence criteria (steps 2 and 2c). `noxcb09` can find the descent direction either through a *truncated Newton method* [8] or a quasi-Newton method [6]. The linesearch procedure uses a backtracking search with quadratic and cubic fit. As for the working basis update, the user can choose between the product form of the inverse, with the Hellerman and Rarick's P^3 reordering algorithm, and an LU factorization. A detailed explanation of the convergence criteria implemented in steps 2 and 2c) could be found in [9].

Table 1: Summarized EIO/UPC computational results

Problem scale	Max. size n	m	t_l	$\mathrm{t_{MI}}/\mathrm{t_{NO}}$ Average	Max.
Large	18000	3000	750	5.10	17.96
Medium	8064	2479	840	2.87	10.00
Small	2256	697	240	1.34	2.47

The efficiency of code `noxcb09` was tested against the general purpose package MINOS 5.3 over the EIO/UPC test problems described in [9]. This is a set of 110 **(NNLC)** problems, some of them randomly generated and others coming from real-world applications. The numerical experiments were done on a Sun Sparc 10/41 workstation with a single processor at 40MHz (≈100Mips, 20Mflops) and 64Mb of central memory (32Mb real memory, 32Mb swapping memory). The results of the computational experiments are summarized in Table 1, where the 110 **(NNLC)** problems of the EIO/UPC collection have been divided into small, medium and large scale. The efficiency of `noxcb09` against `MINOS 5.3` is measured through the ratio between $\mathrm{t_{MI}}$, the execution time for `MINOS 5.3` and $\mathrm{t_{NO}}$, the execution time for `noxcb09` (for instance, $\mathrm{t_{MI}}/\mathrm{t_{NO}} = 2$ means that `noxcb09` is twice as fast as `MINOS 5.3`). From Table 1 it seems clear that, in general, the specialised code `noxcb09` performs better, and that its efficiency improves with the size of the **(NNLC)** problem.

3 `MAPH4` : a specialized projected Lagrangian code for the (STHS) problem

`MAPH4` is an implementation of the projected Lagrangian algorithm **PLA** to solve the **(STHS)** problem described in Section 1.2. This implementation uses the package `noxcb09` to solve the linearized subproblems $\mathbf{(NNLC)}^k$ in step 2b of algorithm **PLA**. Both `MAPH4` and `MINOS 5.3` were applied to solve the **(STHS)** problems shown in Table 2.

Table 2: Characteristics of test cases

case	Nr	Nu	Nm	Nb	Ni	n	m	t_n	t_l
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

In this table, Nr is the number of reservoirs, Nu the number of thermal units, Nm the number of transmission lines, Nb the number of busses, and n, m, t_n and t_l are, respectively, the number of arcs and nodes of the HTTEN and the number of nonlinear and linear side constraints. These problems were solved with `MINOS 5.3` and with `MAPH4`. The computational results are summarized in Table 3. All the runs were performed on the same Sun Sparc 10/41 workstation as was used to solve the EIO/UPC collection. In this table, the column $|\frac{f^*_{MA}-f^*_{MI}}{f^*_{MA}}|$, which expresses the discrepancy in the objective function value at the optimal solution, shows that both programs find the same local optima. It can be observed, looking at the column $\frac{t_{MI}}{t_{MA}}$, that the efficiency or speed–up ratio goes from 2.08 for the smaller case to 5.65 for the larger one. These values are greater than those obtained with the problems in the EIO/UPC collection with similar size (last row of Table 1). Thus, we conclude that, on the limited basis of current computational experience, the efficiency of code MAPH4 is even better than in the linearly constrained case.

Finally, it is interesting to study the speed–up achieved in the cost per iteration by the network flow

Table 3: General efficiency of MAPH4

case	$\lvert \frac{f^*_{MA} - f^*_{MI}}{f^*_{MA}} \rvert$	CPU(sec.) MA	MI	Ratio $\frac{t_{MI}}{t_{MA}}$
A24x	5.0×10^{-4}	2.3	4.8	2.08
B24x	4.7×10^{-2}	3.1	15.3	4.93
B48x	1.5×10^{-3}	6.4	36.2	5.65

specialised implementation of the **PLA** algorithm. Table 4 shows the number of major and minor iterations, the time per minor iteration and the time per iteration ratio, which is a measure of the speed–up of the iteration cost.

Table 4: Efficiency in the iteration time

	Iter. (maj/min)		**millisec./iter.**		**Ratio**
case	MAPH4	MINOS	MAPH4	MINOS	MI/MA
A24x	3/2384	53/2030	0.96	2.3	2.4
B24x	3/3947	99/3855	0.78	3.9	5.0
B48x	3/7988	119/4656	0.80	7.7	9.6

Analysing the last column in Table 4, there is an evident speed–up with the network flow techniques. It is also clear that the speed–up or efficiency factor increases with the size of the problem, but also that the speed–up in the cost per iteration is greater than the general speed–up (last column in Table 3). This degradation of the convergence speed–up factor in comparison with the iteration speed–up is due to the greater number of minor iterations needed to achieve the convergence criteria. Therefore, it seems that, although the network flow techniques significantly improve the efficiency of the solution of the subproblems $\mathbf{(NNLC)}^k$, other parts of the algorithm contributing to the rate of convergence, such as the update rules for the Lagrange multipliers λ^k and the penalty parameter ρ^k, need to be improved.

4 Conclusions

A new projected Lagrangian methodology for the nonlinear network flow problem with side constraints has been proposed and described. This methodology is based on the well–known projected Lagrangian methods with the modified augmented Lagrangian merit function together with basis partitioning techniques, which have proved to be suitable for exploiting the network structure of the problem. The proposed projected Lagrangian algorithm has been implemented and used to solve a set of real power system problems. The computational results shows the efficiency of the proposed method, but also reveal the posibility of increasing the convergence rate with a more careful implementation of several algorithmic issues.

5 Acknowledgements

This work has been partially supported by CICYT grant TAP1999-1075-C02-01.

References

[1] Heredia, F.J. and N. Nabona, "*Optimum short-term hydrothermal scheduling with spinning reserve through network flows*", *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1642–1651, 1995.

[2] Heredia, F.J. and N. Nabona, "*Numerical implementation and computational results of nonlinear network optimization with linear side constraints*", in *Proceedings of the 15th IFIP Conference on System Modelling and Optimization.* P. Kall editor. Springer–Verlag, 1992, 301–310.

[3] Murtagh, B.A and M.A. Saunders, "*MINOS 5.0 User's Guide*", Dept. of Operations Research, Stanford University, CA 9430, USA. 1983

[4] Robinson, S.M., "*A quadratically–convergent algorithm for general nonlinear programming problems*", Mathematical Programming 3 (1972) 145–156.

[5] Murtagh, B.A. and M.A. Saunders, "*A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints*", Mathematical Programming Study, 16 (1982) 84–117.

[6] Murtagh, B.A. and M.A. Saunders, "*Large–Scale Linearly Constrained Optimization*", Mathematical Programming 14 (1978) 41–72.

[7] Kennington, J.L. and R.V. Helgason, "*Algorithm for network programming*", John Wiley and Sons, New York, 1980.

[8] Dembo, R.S. and T. Steihaug, "*Truncated-Newton Algorithms for Large-Scale Unconstrained Optimization*", Mathematical Programming 26 (1983) 190–212.

[9] Heredia, F. J., "*Optimització de Fluxos No Lineals en Xarxes amb Constriccions a Banda No Lineals. Aplicació a Models Acoblats de Coordinació Hidro-Tèrmica a Curt Termini*", Ph.D. Thesis, Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya (1996).