`PLANNC`: a projected Lagrangian based implementation for constrained nonlinear network flow problems

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

31st January 2001

Let us define the standard form of the *Nonlinear Network flow problem with side Constraints* **(NNC)** as:

$$
\begin{array}{lll}
\min\limits_{x\in\mathbb{R}^n} & f(x) & (1)\\
\text{s.t.:} & & \\
 & Ax \qquad\quad = r & (2)\\
 & c(x) + \mathrm{I}_n z_n = b_n & (3)\\
 & Tx \quad + \mathrm{I}_l z_l \;= b_l & (4)\\
 & 0 \le x \;\; \le u_x & (5)\\
 & 0 \le z_n \le u_{zn} & (6)\\
 & 0 \le z_l \;\; \le u_{zl} & (7)
\end{array}
$$

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 $\underline{b_n} \le c(x) \le b_n$. 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.

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

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.

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** [4]. 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. 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:

$$c(x) \approx c^k(x) = c(x^k) + \nabla c(x^k)(x - x^k) \tag{8}$$

   (b) **Minor iterations:** solve the linearized subproblem $\mathbf{(NNLC)}^k$:

$$\min_{x \in \mathbb{R}^n} \quad \Phi^k(x) \tag{9}$$

$$\text{s.t.:} \tag{10}$$

$$Ax = r \tag{11}$$

$$\nabla c(x^k)x + \mathrm{I}_n z_n = b_n^k \tag{12}$$

$$Tx + \mathrm{I}_l z_l = b_l \tag{13}$$

$$0 \leq x \leq u_x \tag{14}$$

$$0 \leq z_n \leq u_{zl} \tag{15}$$

$$0 \leq z_l \leq u_{zn} \tag{16}$$

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 (12) 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.

It must be emphasised that problem $\mathbf{(NNLC)}^k$ defined by equations (9-16) 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 [1]. Package `noxcb09` [1, 2] is an optimization code for solving the nonlinear network flow problem with linear side constraints **(NNLC)**. In [2] the efficiency of code `noxcb09` was tested against the general purpose package `MINOS` [3] over a set of 110 **(NNLC)** problems, some of them randomly generated and others coming from real-world applications.

`plannc` (`p` rojected `l` agrangian `a` lgorithm for `n` onlinear `n` etworks with `c` onstraints) is an implementation of the projected Lagrangian algorithm **PLA**. This implementation uses the package `noxcb09` to solve the linearized subproblems $\mathbf{(NNLC)}^k$ in step 2b of algorithm **PLA**. This code has been tested againts `MINOS`, showing that the efficiency or speed–up ratio goes from 2.08 for the smaller case to 5.65 for the larger one. 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.

References

[1] 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.

[2] Heredia, F.J., "*Computational study of NOXCB: an optimization code for the nonlinear network flow problem with linear side constraints*", in *19th IFIP Conference on System Modelling and Optimization*. Cambridge, England. July 1999.

[3] Murtagh, B.A and M.A. Saunders, "*MINOS 5.4 User's Guide*", Technical report SOL 83-20R. Systems Optimization Laboratory. Dept. of Operations Research, Stanford University, California 94302-4022. Feb. 1995.

[4] 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.