Global scaling of the Proximal Decomposition algorithm for convex multicommodity flow problems

Arnaud Lenoir* Philippe Mahey◇

*EDF R& D, Dept OSIRIS, Clamart, France

◇Laboratoire LIMOS, UMR 6158 CNRS, Clermont Université,
Clermont-Ferrand,France, email: philippe.mahey@isima.fr

Abstract

We analyze here the behaviour of the Proximal Decomposition algorithm with global and adaptive scaling. The algorithm is well adapted to solve convex multicommodity flow problems that arise in Network Routing problems with QoS requirements. Numerical tests performed on quadratic models confirm that adaptive global scaling subsumes former scaling strategies with one or many parameters.

Keywords: *Multicommodity flows, Decomposition algorithms*

1 Introduction

Let $G = (V, E)$ be a directed graph such that V is the set of nodes and $|V| = n$, and E is the set of arcs and $|E| = m$. Each arc is assigned a capacity $C_e, e \in E$, and T is a traffic requirement matrix such that, for each pair (i, j) of nodes, t_{ij} representing the amount of traffic required from node i to node j. Each pair of nodes such that $t_{ij} > 0$ will be referred to as a commodity and the index k will be associated with a commodity, i.e. a pair of origin and destination nodes (respectively o_k and d_k), and a demand of traffic $t_k = t_{o_k d_k}$. The flow residual on a given arc $e \in E$ will be denoted by x_e^0 and $x_e^0 = c_e - \sum_k x_e^k$ where x_e^k is the amount of commodity k routed on arc e. In the arc-node formulation of the multicommodity flow problem, we will need the node-arc incidence matrix A (where $a_{ie} = +1, a_{je} = -1$ if $e = (i, j) \in E$) to express the individual flow constraints as $F_k = \{x^k \in I\!R^m \mid Ax^k = b^k, x^k \geq 0\}$ (where $b_i^k = \begin{cases} +t_k & \text{if } i = o_k \\ -t_k & \text{if } i = d_k \\ 0 & \text{else} \end{cases}$).

We will consider the following multicommodity flow model with convex increasing costs on the arcs (without topology constraints) :

$$(MINDEL) \qquad \min \quad \sum_{e \in E} \phi_e(x_e^0)$$

$$s.t. \quad \begin{cases} x_e^0 + \sum_k x_e^k = c_e & e \in E \\ x^k \in F_k, & k = 1, \ldots, K \\ 0 \leq x^0 \leq c_e & e \in E \end{cases}$$

In practice, the congestion function can be either linear, or convex quadratic. In the classical Kleinrock's model, it approximates the average delay on the arc and $\phi_e(x_e^0) = \frac{c_e - x_e^0}{x_e^0}$, which is a smooth strictly convex and decreasing function defined on $(0, c_e]$.

Various methods have been proposed and tested in the literature to solve (MINDEL), from the Flow Deviation method to the Analytic Center Cutting-plane method, see Ouorou et al [OMV00] for a survey and numerical comparisons. Recent interesting contributions have been proposed by Ouorou in [Ouo07] and [LOP08]. We analyze here a generalized version of the Proximal Decomposition method [MOD95] with a global scaling, i.e. different scaling parameters in each subproblem. Numerical comparisons with the best known algorithms are performed on academic quadratic test problems and real-life multicommodity flow problems.

2 The Proximal Decomposition method for convex cost multicommodity flow problems

The Proximal decomposition method is an adaptation of the Alternating Direction Method of Multipliers originally proposed by [GM75] and further developed by Mahey et al [MOD95]. It can be viewed as a separable Augmented Lagrangian method and it is best understood when applied to the following generic separable convex program with coupling constraints :

$$\begin{array}{rl} \text{Minimize} & \Phi(x) = \sum_{i=1}^{p} \phi_i(x_i) \\ \text{s.t.} & \sum_{i=1}^{p} A_i x_i = b \\ & x_i \in S_i, i = 1, \ldots, p \end{array}$$

where the variables are partitioned in p blocks, all functions being convex on the compact convex sets S_i; suppose too that there exist $x_i \in S_i, i = 1, \ldots, p$ such that $\sum_{i=1}^{p} A_i x_i = b$ so that the problem has an optimal solution with value v^*.

The key idea resumes in adding primal allocation variables $y_i, i = 1, \ldots, p$ to get the equivalent formulation (where $\sum_i b_i = b$ are given initial allocations) :

$$v^* = \min_{x,y} \{\Phi(x) \,|A_i x_i + y_i = b_i, x_i \in S_i, i = 1, \ldots, p, \sum_i y_i = 0\}$$

which is itself equivalent for any $\lambda > 0$ to :

$$v^* = \min_{x,y} \{\Phi(x) + \frac{\lambda}{2} \sum_i \|A_i x_i + y_i - b_i\|^2 \,|A_i x_i + y_i = b_i, x_i \in S_i, i = 1, \ldots, p, \sum_i y_i = 0\}$$

The Lagrangian dual with respect to the local constraints in (x_i, y_i) will induce the following subproblem :

$$v(u_1, \ldots, u_p) = \min_{x \in S, y \in Y} \{\Phi(x) + \frac{\lambda}{2} \sum_i \|A_i x_i + y_i - b_i\|^2 + \sum_i \langle u_i, A_i x_i + y_i - b_i \rangle\}$$

where $Y = \{(y_1, \ldots, y_p) | \sum_i y_i = 0\}$ is the primal subspace. To decompose that subproblem, the trick is to apply some kind of Gauss-Seidel scheme to alternate minimizations with respect to x and y. Indeed, the minimization w.r.t. y can be solved explicitly.

Resuming the steps, we solve first the inner subproblem with respect to x for a fixed u^t and fixed allocations $y^{t-1} \in Y$; observe that the subproblem decomposes in p subproblems, where the i-th subproblem is :

$$\text{Minimize}_{x_i \in S_i} \; [\phi_i(x_i) + \frac{\lambda}{2} \|A_i x_i + y_i^{t-1} - b_i\|^2 + \langle u_i^t, A_i x_i + y_i^{t-1} - b_i \rangle]$$

Let x^t be the optimal solution; we must then solve the primal allocation subproblem, i.e. the following quadratic program with linear equality constraints :

$$\inf_{y \in Y} \; \sum_i [\frac{1}{2} \|A_i x_i^t + y_i - b_i\|^2 + \langle u^t, A_i x_i^t + y_i - b_i \rangle]$$

y^t is an optimal solution of that quadratic program if and only if there exists $v^t \in I\!R^m$ such that :

$$u^t + A_i x_i^t + y_i^t - b_i = v^t, \sum_i y_i^t = 0$$

We can solve easily the linear system in $I\!R^m$ to get $v^t = u^t + \frac{1}{p} r^t$ where $r^t = \sum_i (A_i x_i^t - b_i)$ is the residual of the coupling constraint, and substitute above to get y^t. Finally, the update of u^t will be simply $u^{t+1} = v^t$.

Observe that, even if the method may be interpreted as a separable Augmented Lagrangian technique, the parameter λ is a scaling parameter and not a penalty one, which must be estimated to drive the two primal and dual sequences at the same pace towards the optimal fixed point.

When applied to the convex multicommodity flow problem (MINDEL), the coupling constraints are the multicommodity constraints to aim at decomposing into K single commodity flow problems, plus m congestion subproblems on each arc. It is shown in [OMV00] how quadratic flow problems can be avoided by including the K individual demand satisfaction constraints

$$\sum_p x_{kp} = d_k, k = 1, \ldots, K \tag{1}$$

in the coupling constraints to yield a completely distributed decomposition algorithm. Moreover, the arc-path formulation is used to induce the generation of supporting paths by successive shortest-paths calculations.

3 Global scaling of the proximal decomposition method

Global scaling is a generalization of the Proximal Decomposition algorithm where multiple scaling parameters may be used in each coupling constraint and in each subproblem.

3.1 Scaled allocations

Instead of allocating y_i to each subsystem, we can introduce positive definite matrices M_i different from each other for every i and force the scaled allocations $z_i = M_i(b_i - A_i x_i)$. The concatenated scaled allocation vector z has now to live in a subspace depending on the M_i:

$$\mathcal{A}_M = \{z = (z_1, \ldots, z_p) \in (I\!R^m)^p \mid \sum_{i=1}^{p} M_i^{-1} z_i = 0\} \tag{2}$$

and we get a new equivalent formulation of the problem :

$$\begin{cases} \text{Minimize} & \sum_{i=1}^{p} f_i(x_i) \\ \forall i = 1, \ldots, p & z_i = M_i(b_i - A_i x_i) \\ & z \in \mathcal{A}_M \end{cases} \tag{3}$$

3.2 Algorithm and convergence

Following the same approach as in the original Proximal Decomposition algorithm, we can write the augmented lagrangian function (of parameter $\lambda = 1$) obtained by associating a multiplier w_i to the scaled allocation constraint $z_i = M_i(b_i - A_i x_i)$:

$$L_M(x, z; w) = \sum_{i=1}^{p} L_{i,M_i}(x_i, z_i; w_i) \tag{4}$$

with:

$$L_{i,M_i}(x_i, z_i; w_i) = f_i(x_i) - \langle w_i, M_i(b_i - A_i x_i) - z_i \rangle + \frac{1}{2}\|M_i(b_i - A_i x_i) - z_i\|^2 \tag{5}$$

The minimization in x still consists in solving independent subproblems:

$$\min_{x_i} L_{i,M_i}(x_i, z_i^k; w_i^k) \tag{6}$$

To project on $\mathcal{A}_M$ we use the fact that $\mathcal{A}_M = ker(N)$ and $\mathcal{A}_M^\perp = Im(N^\top)$ with:

$$N = \left(M_1^{-1} | \dots | M_p^{-1}\right)$$

Consequently, the projectors onto $\mathcal{A}_M$ and $\mathcal{A}_M^\perp$ respectively are:

$$P_{\mathcal{A}_M} = I - N^\top (NN^\top)^{-1} N \tag{7}$$

$$P_{\mathcal{A}_M^\perp} = N^\top (NN^\top)^{-1} N \tag{8}$$

The inner matrix $(N^\top N)^{-1}$ is given by $(\sum_{i=1}^p M_i^{-1} M_i^{-\top})^{-1}$ so we can compute the projection v of a vector $\tilde{v}$ by the following procedure:

$$r = \sum_{i=1}^p M_i^{-1} \tilde{v}_i \tag{9a}$$

$$\forall i \quad v_i = M_i^{-T} (\sum_{i'=1}^p M_{i'}^{-1} M_{i'}^{-\top})^{-1} r \tag{9b}$$

v is the projection of $\tilde{v}$ onto $\mathcal{A}_M^\perp$ and $(v - \tilde{v})$ is the projection onto $\mathcal{A}_M$.

Convergence of the new algorithm with global scaling has been analyzed in [LM07].

Proposition 1 *If f is convex proper l.s.c, then the sequence $\{(z^k, w^k)\}_k$ converges to some $(\overline{z}, \overline{w}) \in Z^* \times W^*$.*

Proposition 2 *If each A_i is full-rank, or if the level-sets of f are bounded then the sequence $\{x^k\}_k$ converges to $\overline{x}$, optimal solution.*

In practical situations, we need to introduce an adaptive strategy to update the scaling matrices M_i in order to accelerate convergence. It is shown in [LM07] that the global convergence results are not weakened by using variable scaling with very slight hypotheses. Moreover, the choice of a good strategy is induced by the study of the quadratic case where the authors showed that the optimal choice of the global scaling matrix is exactly the gradient of the primal-dual operator. Thus, an empirical strategy to update the scaling matrices in the general case is to set $M_i^{k+1} = (1 - \alpha_k) M_i^k + \alpha_k D_i^k$ where $0 < \alpha_k < 1$ is a relaxation factor and D_i^k is a diagonal positive definite matrix with the following components :

$$D_{i,j}^k = \frac{(\tilde{u}_i^{k+1})_j - (\tilde{u}_i^k)_j}{(\tilde{y}_i^{k+1})_j - (\tilde{y}_i^k)_j}$$

where $u_i = M_i^T w_i, y_i = M_i(b_i - A_i x_i)$ (omitting the iteration indexes) and where $(\tilde{u}, \tilde{y})$ means the dual and primal allocations just before the projection operation in the algorithm.

4 Numerical study

We present numerical experiments on quadratic examples where each block function $f_(x_i) = \frac{1}{2}\langle x_i, Q x_i \rangle +_i, x_i \rangle$ is strongly convex with positive definite Hessian matrix for different numbers of blocks (parameter p) and

different numbers of coupling constraints (parameter m). Coefficients of b_i and c_i were drawn log-uniformly in $[-100, -0.01] \cup [0.01, 100]$ and the matrices Q_i were built by generating matrices P_i and vectors p_i, with coefficients ranging respectively log-uniformly in $[-10, -0.1] \cup [0.1, 10]$ and $[0.1, 1]$, by setting $Q_i = P_i^T P_i + \mathrm{diag}(p_i)$.

We stopped the algorithm when $\|\tilde{y}^{k+1} - y^k\|^2 + \|\tilde{u}^{k+1} - u^k\|^2$ was lower than $p.10^{-5}$ or when the number of iterations exceeded 5000.

We used $D^0 = \lambda^0 I$ as a starting value, with λ^0 ranging in $[10^{-4}, 100]$. We always chose $\alpha_k = (k+1)^{-10/9}$ for the relaxation strategy. The updating strategy implemented is the one described above with 3 variants :

- **Single** : Single parameter update;
- **Subproblem** : Different parameter update in each subproblem (Global scaling);
- **Component** : Componentwise parameter update (Multidimensional scaling).

Column 'none' means that no adaptive scaling is used (single parameter with a fixed value equal to one).

		Updating rules			
p	m	none	**Single**	**Subproblem**	**Component**
2	5	45	64	63	55
	10	130	123	145	146
	20	71	72	72	82
5	5	53	52	57	64
	10	69	72	69	67
	20	69	71	72	119
10	5	155	139	130	78
	10	94	98	86	84
	20	128	108	119	133
20	5	66	63	70	69
	10	75	74	98	96
	20	100	100	96	141

Table 1: Best number of iterations obtained for $\lambda^0 \in [10^{-4}, 100]$

We have reported in Table 1 the minimum number of iterations obtained and we can observe that the best case is comparable whenever we use an updating strategy or not. However, the need to choose a good initial value disappears. Table 2 show the standard deviation of the number of iteration with respect to the initial value λ^0. Observe that we have skipped all runs which terminated with the maximum iteration count (some appear in column 'none').

We can conclude from these preliminary testbeds that Global Scaling is always better than single scaling but with a rather limited impact. On the other hand, multidimensional scaling is less efficient for these quadratic models. The different graphics on Figure 1 show clearly how efficient is the updating strategy compared with fixed parameter strategies, only superior with the optimal value of the parameter.

References

[GM75] Roland Glowinski and A. Marocco, *Sur l'approximation par éléments finis d'ordre un et la résolution par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires*, RAIRO **9** (1975), 41–76.

[LM07] Arnaud Lenoir and Philippe Mahey, *Accelerating convergence of a separable augmented lagrangian algorithm*, Tech. report, LIMOS, RR 07-14 2007.

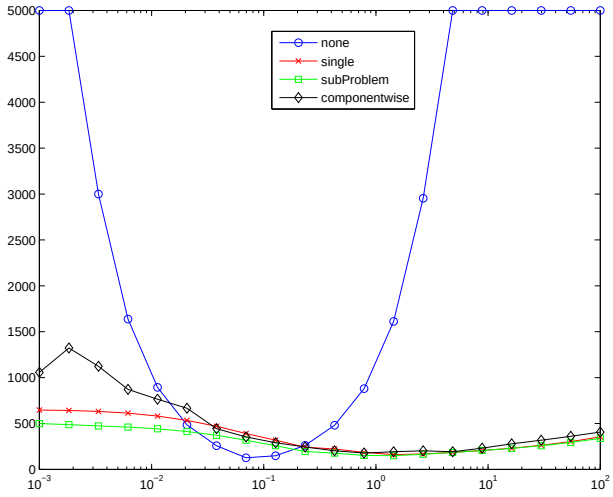

Figure 1: Number of iterations for different updating rules and different initial parameter values. Case $p = 20$, $m = 10$.

		Updating rules			
p	m	none	**Single**	**Subproblem**	**Component**
2	5	415	17	9	21
	10	2076	56	62	93
	20	1776	60	56	58
5	5	2038	38	39	54
	10	2129	37	31	58
	20	2158	59	51	55
10	5	2104	72	39	54
	10	2112	79	55	112
	20	2129	180	123	354
20	5	2108	119	67	430
	10	2081	251	133	320
	20	2092	220	131	321

Table 2: Standard deviation of the number of iteration

[LOP08] Claude Lemarechal, Adam Ouorou, and Giorgiou Petrou, *A bundle-type algorithm for routing in telecommunications networks*, Computational Optimization and Applications (2008).

[MOD95] Philippe Mahey, Said Oualibouch, and Pham Din Tao, *Proximal decomposition on the graph of a maximal monotone operator*, SIAM J. Optimization **5** (1995), 454–466.

[OMV00] Adam Ouorou, Philippe Mahey, and Jean-Philippe Vial, *A survey of algorithms for convex multicommodity flow problems*, Management Science **46** (2000), 126–147.

[Ouo07] Adam Ouorou, *Implementing a proximal algorithm for some nonlinear multicommodity flow problem*, Networks **49** (2007), 18–27.