A bundle-type algorithm for routing in telecommunication data networks

Claude Lemarechal, Adam Ouorou, Giorgios Petrou

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A bundle-type algorithm for routing in telecommunication data networks

Claude Lemaréchal — Adam Ouorou — Georgios Petrou

N° 6010

Novembre 2006

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6010--FR+ENG

A bundle-type algorithm for routing in telecommunication data networks

Claude Lemaréchal [*] , Adam Ouorou , Georgios Petrou [†]

Thème NUM —Systèmes numériques
Projet Bipop

Rapport de recherche n° 6010 — Novembre 2006 —14 pages

Abstract: To optimize the quality of service through a telecommunication network, we propose an algorithm based on Lagrangian relaxation. The bundle-type dual algorithm is adapted to the present situation, where the dual function is the sum of a polyhedral function (coming from shortest paths problems) and of a smooth function (coming from the congestion).

Key-words: Convex optimization, routing, multicommodity flows, Kleinrock delay function

[*] INRIA Rhônes-Alpes, 655 avenue de l'Europe, Montbonnot, 38334, Saint Ismier, claude.lemarechal@inria.fr
[†] France Telecom R&D, CORE/MCN, 38-40 rue du Général Leclerc, 92794 Issy-Les-Moulineaux cedex 9, {adam.ouorou,giorgios.petrou}@orange-ft.com

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 —Télécopie +33 4 76 61 52 52

Un algorithme de type faisceaux pour le routage dans une réseau de télécommunications

Résumé : Pour optimiser la qualité de service dans un réseau de télécommunications, nous proposons un algorithm à base de relaxation lagrangienne. L'algorithme dual adapte la méthode de faisceaux à la présente situation où la fonction duale est somme d'un terme polyédral (provenant des plus courts chemins) et d'un terme différentiable (provenant de la fonction de congestion).

Mots-clés : Optimisation convexe, routage, multiflot, fonction de retard de Kleinrock

1 Introduction

We consider the following problem

$$\begin{array}{ll} \min & f(y) := \displaystyle\sum_{j=1}^{n} f_j(y_j) \\ \text{s.t.} & Ax^k = b^k,\ x^k \geqslant 0, \quad k = 1, \ldots, K, \\ & \displaystyle\sum_{k=1}^{K} x^k = y, \\ & 0 \leqslant y < c\,, \end{array} \tag{1}$$

where

· $f_j(y_j)$ is proportional to Kleinrock average delay function:

$$f_j(y_j) = \frac{y_j}{c_j - y_j} \quad \text{if } 0 \leqslant y_j < c_j \text{ (and } +\infty \text{ otherwise)}\ , \tag{2}$$

· A is the node-arc incidence matrix of a graph $G = (V, E)$ (m nodes, n arcs),

· $b^k \in \mathbb{R}^m$ are vectors with two nonzero components (corresponding to an origin s_k and destination t_k) such that $\sum_{i=1}^{m} b_i^k = 0$,

· $c \in \mathbb{R}^n$ is the vector of arc capacities,

· x^k are flow vectors representing K commodities between source nodes s_k and sink nodes t_k, $k = 1, \ldots, K$,

· y is the total link flow vector.

Problem (1), often called (convex) *multicommodity flow*, occurs in data communication networks and plays an important role in the optimization of network performances. Various methods have been proposed in the literature to solve it, we refer to [15] for a review.

– Direct methods exploit the problem's block structure. Most popular is flow deviation [6] because of its simplicity; it is a special case of the Frank-Wolfe method [4], which works on a sequence of linearized problems. It has slow convergence and many authors have tried to improved it.

– Other classical mathematical programming algorithms (Newton, conjugate gradient) have been adapted to the structure of (1); see [5] for example.

– Some proposals adopt a dual point of view. They can be classified as cutting-plane methods and arise in connection with Lagrangian relaxation, resulting in a nonsmooth Lagrangian dual. There we find proximal, ACCPM or subgradient methods.

Our present proposal belongs to this latter class and solves the dual by a convex-optimization algorithm, comparable to the recent approach of [1]. We fully exploit the structure of the dual objective function, which is a sum of two pieces: the convex conjugate of the delay function f, and a polyhedral function which can be approximated through cutting planes. The smoothness of the first component provides useful second-order information, which we introduce as a quadratic regularization for the second component. With respect to standard bundle methods (which would apply cutting planes to the smooth component as well and append an artificial quadratic term), this necessitates new rules for the so-called ascent and null steps.

The paper is organized as follows. We propose in §2 a (classical) Lagrangian relaxation of (1) and in §3 our model of the Lagrangian dual function using a second-order oracle. Our adaptation of the bundling technique to cope with this special model is the subject of §4, while §5 recalls the aggregation mechanism, important for primal recovery. The algorithm is detailed in §6, its convergence established in §7, while §8 shows how to recover the primal optimal solution from the dual algorithm. Finally, §9 is devoted to numerical results.

2 Lagrangian relaxation

Associating with the coupling constraints $\sum_{k=1}^{K} x^k = y$ the dual variables $u \in \mathbb{R}^n$, we define the Lagrangian

$$L(x, y, u) = \sum_{j=1}^{n} f_j(y_j) + \sum_{j=1}^{n} u_j \Big(-y_j + \sum_{k=1}^{K} x_j^k \Big)$$

and we apply Lagrangian relaxation, as explained for example in [13]. We minimize $L(\cdot,\cdot,u)$ for fixed u; here, this amounts to computing

$$\Phi_j(u_j) \;:=\; \min_{0\leqslant y_j<c_j}\{f_j(y_j)-u_jy_j\} \quad [=-f_j^*(u_j)], \quad j=1,\dots,n, \tag{3}$$

$$\Pi^k(u) \;:=\; \min\{u^\top x^k:\ Ax^k=b^k,\ x^k\geqslant 0\}. \quad k=1,\dots,K. \tag{4}$$

It will be convenient for the sequel to use the notation

$$\Phi(u):=\sum_{j=1}^n \Phi_j(u_j)\,, \qquad \Pi(u):=\sum_{k=1}^K \Pi^k(u).$$

The dual problem is then to maximize with respect to u the so-called *dual function*, namely: solve

$$\max_{u\in\mathbb{R}^n}\theta(u), \quad \text{where } \theta(u):=\Phi(u)+\Pi(u). \tag{5}$$

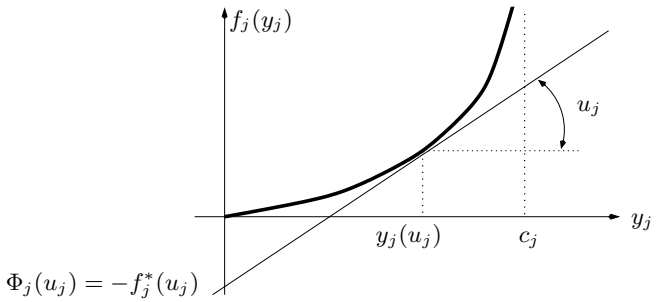

Figure 1: Conjugating Kleinrock function

In (3) (where f_j^* denotes the conjugate function of f_j), the optimal y_j is easy to compute (see Fig. 2): we obtain

$$y_j(u_j)=\begin{cases} c_j-\sqrt{\frac{c_j}{u_j}} & \text{if } u_j\geqslant\frac{1}{c_j}\\ 0 & \text{otherwise}\,,\end{cases} \tag{6}$$

so that Φ_j has the expression

$$\Phi_j(u_j)=\begin{cases} -\left(\sqrt{c_ju_j}-1\right)^2 & \text{if } u_j\geqslant\frac{1}{c_j}\\ 0 & \text{otherwise}\,.\end{cases} \tag{7}$$

On the other hand, computing Π from (4) amounts to solving K independent shortest path problems, each of which being posed between s_k and t_k and having arc lengths u_j. The next simple result says that this computation has to be done with *positive* arc lengths only.

Proposition 1 *Consider the set*

$$U:=\left\{u\in\mathbb{R}^n:\ u_j\geqslant\frac{1}{c_j},\ j=1,\dots,n\right\}. \tag{8}$$

For any $u\notin U$, there is $u'\in U$ such that $\theta(u')\geqslant\theta(u)$. As a result, (5) *is not changed if the constraint $u\in U$ is inserted (this can only eliminate some optimal u).*

Proof. Let $u\in\mathbb{R}^n$ be such that $u_{j_0}<1/c_{j_0}$ for some j_0 and increase u_{j_0} until the value $1/c_{j_0}$. Because $x^k\geqslant 0$ in (4), each Π^k can only increase. As for the Φ_j's, only Φ_{j_0} can change; but (7) shows that it remains constantly 0. □

Thus, to ease the computation of the Π^k's, the constraints $u_j\geqslant 1/c_j$ may be inserted into (5): this does not prevent the computation of a dual optimal solution and does not change the optimal dual value.

3 Model of the dual function

Taking advantage of Proposition 1, we reformulate (5) as

$$\max_{u\in U}\theta(u) := \Phi(u)+\Pi(u) := \sum_{j=1}^{n}\Phi_j(u_j)+\sum_{k=1}^{K}\Pi^k(u). \tag{9}$$

To solve it, we propose a hybrid method working as follows:

– Each smooth function Φ_j is approximated by its second-order development, as in Newton's method. This development is made at a point – call it $\hat u$ – controlled according to its (dual) objective value $\theta(\hat u)$.

– Each polyhedral function Π^k is approximated by cutting planes, as in Kelley's method [9, 2].

In a way, the above method can be viewed as a bundle variant [14], see also [1], in which

– the cutting-plane paradigm is applied to a part of the (dual) objective function, namely Π,

– stabilization around $\hat u$ is obtained by the Newtonian term $u^\top\nabla^2\Phi(\hat u)u$, instead of an artificial $|u|^2$ weighted by a hard-to-tune penalty coefficient.

Finally, since Lagrangian relaxation is column generation, our algorithm can be viewed as a Dantzig-Wolfe variant where the masters are suitably stabilized.

At each iteration s, (4) is solved with the iterate u^s, provides a shortest path $x^k(u^s)$, which in turn provides an upper linearization: by definition, $\Pi^k(u)\leqslant u^\top x^k(u^s)$ for all $u\in\mathbb{R}^n$. Accumulating these shortest paths, we form at the current iteration S the K polyhedral functions

$$\hat\Pi^k(u) := \min_{s=1,\dots,S} u^\top x^k(u^s) \geqslant \Pi^k(u), \quad \text{for all } u\in\mathbb{R}^n. \tag{10}$$

As for the Φ_j's, note that they have analytic derivatives over the feasible domain:

$$\Phi_j'(u_j)=\sqrt{\frac{c_j}{u_j}}-c_j, \quad \Phi_j''(u_j)=\frac{-1}{2u_j}\sqrt{\frac{c_j}{u_j}} \qquad \text{if } u_j\geqslant\frac{1}{c_j} \tag{11}$$

and that $-\Phi_j'(u_j)=y_j(u_j)$ is the optimal y_j of (6).

In addition to the $\hat\Pi^k$'s, suppose also that a *stability center* $\hat u\in U$ is available at the current iteration. Analogously to Π^k, each Φ_j will be replaced by its quadratic approximation near $\hat u$:

$$\tilde\Phi_j(u_j) := \Phi_j(\hat u_j)-y_j(\hat u_j)(u_j-\hat u_j)+\frac12 M_j(u_j-\hat u_j)^2 \quad [\simeq\Phi_j(u_j)], \tag{12}$$

where $y_j(\hat u_j)=-\Phi_j'(\hat u_j)$ is given by (6) and $M_j:=\Phi_j''(\hat u_j)$ by (11).

We will use the notation

$$\tilde\Phi(u) := \sum_{j=1}^{n}\tilde\Phi_j(u_j), \quad \hat\Pi(u) := \sum_{k=1}^{K}\hat\Pi^k(u), \quad \hat\theta(u) := \tilde\Phi(u)+\hat\Pi(u)$$

and the essential part of the algorithm will be to maximize $\hat\theta$, which can be viewed as a *model* of the true dual function θ in (9). We also find it convenient to use the change of variable $h=u-\hat u$: we solve

$$\max\big\{\tilde\Phi(\hat u+h)+\hat\Pi(\hat u+h) : \hat u+h\geqslant 1/c\big\}.$$

Introducing additional variables π^k (connoting $\hat\Pi^k$), this is the simple quadratic programming problem

$$\begin{array}{ll}\max\limits_{h,\pi} & \Big\{\tilde\Phi(\hat u+h)+\sum\limits_{k=1}^{K}\pi^k\Big\}\\ \text{s.t.} & \pi^k\leqslant(\hat u+h)^\top x^k(u^s), \quad \text{for } \left\{\begin{array}{l}k=1,\dots,K,\\ s=1,\dots,S,\end{array}\right.\\ & h_j\geqslant\dfrac{1}{c_j}-\hat u_j, \quad j=1,\dots,n.\end{array} \tag{13}$$

Note that the somewhat artificial constraint $\hat u+h=u\geqslant 1/c$ is useful for a fast computation of $\Pi^k(u)$ in (4); but it is even more useful for the dual algorithm: from (7), $\Phi_j''(u_j)=0$ if $u_j<1/c_j$. If such a u is a $\hat u$, then $\tilde\Phi_j$ will degenerate and (13) will perhaps be unbounded from above[1].

[1]This diffi culty can be eliminated, though. Observe in (1) that the constraint $y\geqslant 0$ is redundant; each f_j of (2) can therefore be extended as we like on $\mathbb{R}_-$. A convenient extension is

$$f_j(y_j) := \frac{y_j^2}{c_j^2}+\frac{y_j}{c_j} \quad \text{for } y_j\leqslant 0,$$

Proposition 2 *Problem* (13) *has a unique optimal solution* $(\hat{h},\hat{\pi})$*, with* $\hat{\pi}^k=\hat{\Pi}^k(\hat{u}+\hat{h})$.

Proof. From standard convex analysis, each function $\hat{\Pi}^k$ is concave; and each $\tilde{\Phi}_j$ is a strictly concave quadratic function (from (11), $M_j<0$!): $\hat{\theta}$ has a unique maximum $\hat{u}+\hat{h}$, making up the h-part of the optimal solution in (13); and each π^k has to reach its maximal value, namely $\hat{\Pi}^k(\hat{u}+\hat{h})$. □

Note to conclude this section that our approach can of course be applied to the maximization of any concave function θ made up of two parts: a polyhedral one (given by an oracle) and a smooth one (whose Hessian is at hand). In (12), M_j is the jj^{th} entry of the Hessian (here diagonal) of the smooth part of θ.

4 Ascent steps, null steps and backtracking

The resolution of (13) *predicts* an increase

$$\delta:=\hat{\theta}(\hat{u}+\hat{h})-\theta(\hat{u}) \tag{14}$$

in the dual objective function. We will see below (Lemma 4) that δ is an optimality measure of $\hat{u}$: the algorithm can be stopped if δ is small. Note that $\delta\geqslant 0$ anyway, since $\theta(\hat{u})\leqslant\hat{\theta}(\hat{u})\leqslant\hat{\theta}(\hat{u}+\hat{h})$.

In our bundle variant, the next iterate u^+ will be set to $\hat{u}+t\hat{h}$, for some suitable stepsize $t\in]0,1]$. When doing so, it is desirable to increase "substantially" the dual function; this is quantified by

$$\theta(\hat{u}+t\hat{h})\geqslant\theta(\hat{u})+\kappa t\delta, \tag{15}$$

$\kappa\in]0,1[$ being a fixed tolerance. If (15) holds, the next iterate u^+ is set to $\hat{u}+t\hat{h}$ and $\hat{u}$ can safely be moved to the definitely better point u^+; iteration S is terminated, this is an *ascent step* in the bundle terminology. Theorem 1 will show that this update does imply convergence of the algorithm.

Another desirable feature is to improve the approximation $\hat{\Pi}$. If $\hat{u}$ is not moved (no ascent step is made), $(\hat{h},\hat{\pi})$ must not be feasible in the next QP (13) – see Lemma 4 below. We quantify the required improvement as

$$\Pi(\hat{u}+t\hat{h})\leqslant\Pi(\hat{u})+t[\hat{\Pi}(\hat{u}+\hat{h})-\Pi(\hat{u})]-\kappa' t\delta, \tag{16}$$

$\kappa'\in]0,\kappa]$ being another positive tolerance; by concavity of $\hat{\Pi}$, this implies that $\Pi(\hat{u}+t\hat{h})$ is "definitely" smaller than $\hat{\Pi}(\hat{u}+t\hat{h})$. If (16) holds, u^+ is set to $\hat{u}+t\hat{h}$ and $\hat{u}$ is kept as it is; again iteration S is terminated, this is a *null step*.

When neither (15) nor (16) holds, $\tilde{\Phi}$ is a bad approximation of Φ. To improve it, we decrease t and test (15), (16) again; we will make sure in Lemma 2 below that this *backtracking phase* cannot go forever.

Standard bundle has no backtracking: it merely uses $t=1$ and overlooks (16). Actually, if the whole of θ is approximated by cutting planes, "not (15)" implies that $(\hat{h},\hat{\pi})$ will be infeasible in the next instance of (13). In the present variant, the argument is destroyed by the $\tilde{\Phi}$-part of $\hat{\theta}$.

5 Toward primal recovery: the aggregate linearization

The necessary material to state the dual algorithm is now available. However, remember that our problem is rather (1) than (5). To establish the connexion between the two resolutions, we need some more sophisticated material from convex analysis. First we introduce some notation.

- The normal cone $\mathrm{N}_U(u)$ is the set of vectors $\nu\in\mathbb{R}^n$ such that $(v-u)^\top\nu\leqslant 0$ for all $v\in U$;
- $y(\hat{u})\in\mathbb{R}^n$ will be the vector whose components are $y_j(\hat{u}_j)$, see (6);
- $M:=\nabla^2\Phi(\hat{u})$ will be the diagonal (negative definite) $n\times n$ matrix whose jjth element is $M_j=\Phi_j''(\hat{u}_j)$ of (12);
- the unit simplex of $\mathbb{R}^S$ will be $\Delta^S:=\{\alpha\in\mathbb{R}^S:\sum_s\alpha^s=1,\ \alpha\geqslant 0\}$.

Now we recall some elementary subdifferential calculus. Denote by

$$\mathcal{G}\theta(u):=-\partial(-\theta)(u)=\{g\in\mathbb{R}^n:\theta(v)\leqslant\theta(u)+(v-u)^\top g\ \text{ for all } v\in\mathbb{R}^n\}$$

the "superdifferential" of a concave function such as θ.

- The superdifferential of the smooth concave function $\tilde{\Phi}$ is its gradient: $\mathcal{G}\tilde{\Phi}(u)=-y(\hat{u})+M(u-\hat{u})$;

which is C^2 and strongly convex; its conjugate enjoys the same properties and the algorithm can work.

· the superdifferential of $\hat\theta$ is the sum of superdifferentials:

$$\partial\hat\theta(u) = -y(\hat u) + M(u-\hat u) + \Big\{\sum_{k=1}^K \hat x^k\Big\} \quad \text{with } \hat x^k \text{ describing } \partial\hat\Pi^k(u);$$

· the superdifferential of $\hat\Pi^k$ is the convex hull of the active slopes in (10):

$$\partial\hat\Pi^k(u) = \Big\{\hat x^k = \textstyle\sum_{s=1}^S \alpha^s x^k(u^s) : \alpha\in\Delta^S, \\ \alpha^s = 0 \text{ if } u^\top x^k(u^s) > \hat\Pi^k(u)\Big\}. \tag{17}$$

This allows us to describe the solution of (13):

Proposition 3 *The unique optimal solution $\hat h$ of* (13) *is characterized as follows: for some $\hat x^k \in \partial\hat\Pi^k(\hat u+\hat h)$, $k=1,\dots,K$ and $\nu\in \mathrm{N}_U(\hat u+\hat h)$,*

$$\hat h = M^{-1}\hat g\,, \quad \textit{where} \quad \hat g := y(\hat u) - \hat x + \nu,\ \hat x := \sum_{k=1}^K \hat x^k \in \partial\Pi(\hat u+\hat h). \tag{18}$$

Proof. Watching for various changes of sign, apply the optimality condition [7, Thm. VII.1.1.1(iii)]: there is some supergradient in $\partial\hat\theta(\hat u+\hat h)$ lying in $\mathrm{N}_U(\hat u+\hat h)$. In view of the above-mentioned calculus rules, this writes

$$-y(\hat u) + M\hat h + \sum_{k=1}^K \hat x^k = \nu,$$

which is just (18). □

With the particular form of U, the property $\nu\in\mathrm{N}_U(\hat u+\hat h)$ means that $\nu\leqslant 0$ is in complementarity with $\hat u+\hat h-1/c\geqslant 0$.

Note that each $\hat x^k$ is a convex combination as described in (17). To make up $\hat x$, one needs K sets of convex multipliers $\alpha^k\in\Delta^S$, which indeed are the KKT multipliers (not necessarily unique) associated with the constraint involving π^k in (13); any reasonable QP solver computes them, in addition to the optimal $(\hat h,\hat\pi)$. In the next statement, this remark could also be used for an alternative proof, based on complementarity slackness:

Lemma 1 *With the notation of Proposition 3, $\hat\Pi^k(u)\leqslant u^\top\hat x^k$ for all $u\in\mathbb{R}^n$. Equality holds for $u=\hat u+\hat h$. In particular, $\hat\Pi(\hat u+\hat h) = (\hat u+\hat h)^\top\hat x$.*

Proof. Apply (17) with $u=\hat u+\hat h$: $\hat x^k = \sum_s \alpha^s x^k(u^s)$ for some $\alpha\in\Delta^S$. The required inequality is therefore clear from the definition (10) of $\hat\Pi^k$. Besides, this convex combination involves only indices s such that $(\hat u+\hat h)^\top x^k(s) = \hat\Pi^k(\hat u+\hat h)$, so the stated equality holds as well; and the last statement follows by summation over k. □

The whole business of dual convergence will be to drive δ to 0, and this has interesting consequences:

Proposition 4 *With the notation of Proposition 3, $\delta=\delta_h+\delta_x+\delta_\nu$, where*

$$\delta_h := -\frac12\hat h^\top M\hat h \geqslant 0\,, \quad \delta_x := \hat u^\top\hat x - \Pi(\hat u)\geqslant 0\,, \quad \delta_\nu := \hat h^\top\nu\geqslant 0. \tag{19}$$

Besides, for all $u\in U$,

$$\theta(u)\leqslant\theta(\hat u)+\delta_x+\delta_\nu-(u-\hat u)^\top\hat g. \tag{20}$$

Proof. Write the definition (14) of δ, using (12) and Lemma 1:

$$\begin{array}{rcl}\delta &=& \Phi(\hat u) - \hat h^\top y(\hat u) + \frac12\hat h^\top M\hat h + (\hat u+\hat h)^\top\hat x - \Phi(\hat u) - \Pi(\hat u)\\ &=& \frac12\hat h^\top M\hat h + \big[\hat u^\top\hat x - \Pi(\hat u)\big] - \hat h^\top(y(\hat u)-\hat x)\end{array}$$

and (19) follows because $y(\hat u)-\hat x = M\hat h-\nu$ from (18).

Then remember from (11) that M is negative semi-definite: $\delta_h\geqslant 0$. The property $\delta_x\geqslant 0$ comes from Lemma 1; and $\delta_\nu = (\hat u+\hat h-\hat u)^\top\nu$ is nonnegative because $\nu\in\mathrm{N}_U(\hat u+\hat h)$.

Now take an arbitrary $u \in U$. Using feasibility of $\hat{x}^k$ in (4), concavity of Φ and definition of normal cones,

$$\begin{array}{rcl} \Pi(u) & \leqslant & u^\top \hat{x} = \hat{u}^\top \hat{x} + (u-\hat{u})^\top \hat{x} \\ \Phi(u) & \leqslant & \Phi(\hat{u}) - (u-\hat{u})^\top y(\hat{u}) \\ 0 & \leqslant & (\hat{u}+\hat{h}-u)^\top \nu = (\hat{u}-u)^\top \nu + \hat{h}^\top \nu. \end{array}$$

Summing up and disclosing appropriate δ-values:

$$\theta(u) \leqslant \Pi(\hat{u}) + \delta_x + \Phi(\hat{u}) + (u-\hat{u})^\top(-\hat{g}) + \delta_\nu,$$

which is just (20). □

Thus, when δ is small, δ_h, δ_x and δ_ν are small. If M behaves itself, $|\hat{g}|$ is also small and (20) shows that $\hat{u}$ is approximately optimal in (9).

6 The algorithm

We are now in a position to state the algorithm. Knowing the expression of Kleinrock function, it works with the help of the "oracle" solving (4) for given $u \geqslant 1/c$. It uses the improvement parameters κ and κ' satisfying $0 < \kappa' \leqslant \kappa < 1$, and the stopping tolerance $\underline{\delta} \geqslant 0$. The starting point $u^1 \in U$ is given, as well as the initial shortest paths $x^k(u^1)$ forming the initial bundle, and the initial quadratic model $\tilde{\Phi}$.

Algorithm 1 (Combined Newton-cutting-plane algorithm (`Ncp`)) Initialize $S=1$, $\hat{u} = \hat{u}^1 = u^1$.
STEP 1 (*Trial point finding*). Find $\hat{h}$, $\hat{x} = \hat{x}^S$ and $\hat{g} = \hat{g}^S$ as described by Proposition 3. Compute $\delta = \delta^S$ by (14)
STEP 2 (*Stopping test*). If $\delta^S \leqslant \underline{\delta}$ stop, returning $\hat{u}$ and $\hat{x}$.
STEP 3 (*Line-search*). Set $t=1$.
  STEP 3.1 (*Oracle call*). Set $u^+ := \hat{u} + t\hat{h}$. Compute $x^k(u^+)$ from (4) and the resulting values $\Pi(u^+)$, $\theta(u^+)$.
  STEP 3.2 (*Ascent test*). If (15) holds, set $\hat{u}^{S+1} = u^+$; update the quadratic approximation $\tilde{\Phi}$.
    Go to Step 4.
  STEP 3.3 (*Null-test*). If (16) holds, set $\hat{u}^{S+1} = \hat{u}^S$.
    Go to Step 4.
  STEP 3.4 (*Interpolation*). Select a new t "well inside" the segment $]0,t[$.
    Go to Step 3.1.
STEP 4 (*Bundle updating and loop*). For $k=1,\dots,K$, append $x^k(u^+)$ obtained in Step 3.1 to the bundle. Increase S by 1 and go to Step 1. □

In Step 3.4, the simplest is to divide t by 2. More sophisticated interpolation formulae can be designed, in the spirit of cubic fitting in NLP. The expression "well inside" can mean for example "in the segment $[0.1t,\ 0.9t]$": the new t should be not too close to the old t for obvious reasons, but also not too close to 0 for convergence of the overal algorithm. Of course, the bundle updating of Step 4 is beneficial for storage requirement and complexity of (13). However this should be done cautiously, as it might substantially deteriorate the algorithm's efficiency.

Remark 1 (Sum of max vs. max of sum) Our approximation of Π uses K *individual* approximations $\hat{\Pi}^k$ of (10). Traditional bundle methods actually ignore the summation property $\Pi = \sum_k \Pi^k$: they use just one supergradient, say $\xi^s \in \partial\Pi(u^s)$ for each u^s, corresponding to the *compound* linearization $u^\top \xi^s$ of $\Pi(u)$. Here, ξ^s is of course the sum of the shortest paths $x^k(u^s)$.

Storing S linearizations needs Sn elements (as opposed to the KSn elements needed here). Besides, the S^{th} quadratic problem (13) simplifies to

$$\begin{array}{ll} \max\limits_{u,\pi} & \{\tilde{\Phi}(u) + \pi\} \\ \text{s.t.} & \pi \leqslant u^\top \sum\limits_{k=1}^{K} x^k(u^s), \quad \text{for } s=1,\dots,S, \\ & u_j \geqslant \frac{1}{c_j}, \quad j=1,\dots,n, \end{array}$$

which has just S linking constraints. It corresponds to a less accurate description of Π, therefore its solution is probably of lesser quality.

Even with the above simplification, Algorithm 1 needs potentially infinite memory. However, traditional bundle methods make use of the aggregate linearization $\hat{x}$ revealed by Proposition 3: a "minimal" variant would approximate Π at iteration $S+1$ by a polyhedral function made up of two pieces only, namely

$$\min\{u^\top \hat{x}^S, u^\top \xi^{S+1}\}.$$ □

7 Dual convergence

In this section, we pretend that $\underline{\delta} = 0$ in Algorithm 1. Then we prove that $\liminf \delta^S = 0$; this implies that the algorithm will eventually stop if $\underline{\delta} > 0$. We use the terminology introduced in §5. First we make sure that each backtracking phase terminates.

Lemma 2 *Assume $\kappa + \kappa' \leqslant 1$; let $-L \leqslant -\ell < 0$ be lower and upper bounds on the eigenvalues of M over the segment $[\hat{u}, \hat{u} + \hat{h}]$. Then* (15) *or* (16) *holds (or both) whenever $t \leqslant \ell/L$.*

Proof. Suppose that neither (16) nor (15) holds. Subtracting and using definitions:

$$\begin{array}{rcl} \Phi(\hat{u} + t\hat{h}) & < & \Phi(\hat{u}) - t[\hat{\Pi}(\hat{u} + \hat{h}) - \Pi(\hat{u})] + (\kappa + \kappa')t\delta \\ & = & \Phi(\hat{u}) + t[\tilde{\Phi}(\hat{u} + \hat{h}) - \Phi(\hat{u})] + (\kappa + \kappa' - 1)t\delta \\ & \leqslant & \Phi(\hat{u}) + t[\tilde{\Phi}(\hat{u} + \hat{h}) - \Phi(\hat{u})] \\ & = & \Phi(\hat{u}) + t\big[-y(\hat{u})^\top \hat{h} + \frac{1}{2}\hat{h}^\top M \hat{h}\big]. \end{array}$$

Apply some mean-value theorem to Φ: for example, denoting by $\tilde{M}$ the Hessian of Φ at some point between $\hat{u}$ and $\hat{u} + t\hat{h}$

$$\Phi(\hat{u} + t\hat{h}) = \Phi(\hat{u}) - ty(\hat{u})^\top \hat{h} + \frac{t^2}{2}\, \hat{h}^\top \tilde{M} \hat{h},$$

so that

$$-ty(\hat{u})^\top \hat{h} + \frac{t^2}{2}\, \hat{h}^\top \tilde{M} \hat{h} < t\Big[-y(\hat{u})^\top \hat{h} + \frac{1}{2}\hat{h}^\top M \hat{h}\Big].$$

Divide by $t > 0$ and simplify to obtain

$$-tL|\hat{h}|^2 \leqslant t\hat{h}^\top \tilde{M} \hat{h} < \hat{h}^\top M \hat{h} \leqslant -\ell|\hat{h}|^2 < 0. \qquad \square$$

As usual with a bundle method, our convergence proof considers two cases.

7.1 Case of infinitely many ascent steps

To simplify our analysis, we will assume here that (1) is feasible. This guarantees the Slater property, which is a key for an appropriate primal-dual behaviour:

Lemma 3 *If* (1) *has a feasible point, then θ is sup-compact on U: the sets of $u \in U$ such that $\theta(u) \geqslant b$ are (closed and) bounded. As a result,* (9) *has a unique solution.*

Proof. Closedness classically follows from upper semi-continuity of a dual function. Let $\mathring{x}$ and $\mathring{y} = \sum_k \mathring{x}^k$ make a feasible point. Because each $\mathring{y}_j < c_j$, we can find $\varepsilon > 0$ and $B > 0$ such that

$$\text{for } j = 1, \ldots, n, \quad \mathring{y}_j \leqslant y_j \leqslant \mathring{y}_j + \varepsilon \quad \Longrightarrow \quad f_j(y_j) \leqslant B.$$

Take an arbitrary $u \in U \subset \mathbb{R}^n_+$ and set $y^u := \mathring{y} + \varepsilon \frac{u}{|u|}$. By definition of the dual function,

$$\theta(u) \leqslant L(\mathring{x}, y^u, u) = f(y^u) + u^\top \Big(-\mathring{y} - \varepsilon \frac{u}{|u|} + \sum_{k=1}^K \mathring{x}^k\Big) = f(y^u) - \varepsilon|u|;$$

but $0 \leqslant y^u_j \leqslant \mathring{y}_j + \varepsilon$ for each j ($0 \leqslant u_j \leqslant |u|$!), hence $f(y^u) \leqslant nB$. We have proved that $b \leqslant \theta(u)$ implies $b \leqslant nB - \varepsilon|u|$.

Then (9) has at least one optimal solution, which is unique because Π is concave and (11) shows that Φ is strictly concave. $\square$

Sup-compactness classically eliminates the duality gap and allows the characterization of primal-dual solutions via the superdifferential of the dual function. We will recover these results in a constructive way, by establishing appropriate convergence properties of the sequences $\hat{u}$ and $(y(\hat{u}), \hat{x})$ (the latter being established in §8 below).

Theorem 1 *Assume that* (1) *has a feasible point and let Algorithm 1 generate an infinite sequence $\mathcal{S}$ of ascent steps. Then the subsequences $(\delta^s)_\mathcal{S}$, $(\hat{h}^s)_\mathcal{S}$ and $(\hat{g}^s)_\mathcal{S}$ tend to 0; and the sequence $\hat{u}^s$ tends to the optimal solution of* (9)[2].

[2]Note that the whole sequence $\hat{u}^s$ coincides with $(\hat{u}^s)_\mathcal{S}$, since $\hat{u}^{s+1} = \hat{u}^s$ if $s \notin \mathcal{S}$.

Proof. The increasing sequence $\theta(\hat{u}^s)$ has a limit, not larger than the optimal value $\bar{\theta}$ of (9). From Lemma 3, the sequence $\hat{u}^s$ is bounded; $L > 0$ [resp. $\ell > 0$] of Lemma 2 is bounded from above [resp. away from 0], t is bounded away from 0: say $t \geqslant \underline{t} > 0$. Then we have from (15)

$$\begin{aligned} \theta(\hat{u}^{s+1}) &\geqslant \theta(\hat{u}^s) + \kappa \underline{t} \delta^s \quad && \text{if } s \in \mathcal{S} \\ \theta(\hat{u}^{s+1}) &= \theta(\hat{u}^s) && \text{otherwise} \end{aligned}$$

and we obtain by summation $\sum_{s\in\mathcal{S}} \delta^s \leqslant [\bar{\theta} - \theta(u^1)]/\kappa\underline{t}$: $(\delta^s)_{\mathcal{S}}$ tends to 0. The three components of δ^s in (19) tend to 0 as well, and this is also true of the subsequences $(\hat{g}^s)_{\mathcal{S}}$ and $(\hat{h}^s)_{\mathcal{S}}$.

Then take a cluster point of $\hat{u}^s$ and pass to the limit in (20): this cluster point has to be the unique optimal solution of (9). □

Note that, if (1) has no feasible point, then $\theta(\hat{u})$ will typically tend to $+\infty$; δ has no reason to tend to 0, the stop in Algorithm 1 will never occur. To prevent this situation, it is wise to insert an "emergency stop" when $\theta(\hat{u})$ is unduly large.

7.2 Case of finitely many ascent steps

First we show that the next QP will modify the present $\hat{h}$ (remember from Proposition 2 that $\hat{\Pi}(\hat{u}+\hat{h}) = \hat{\pi}$):

Lemma 4 *If* (16) *holds, the new linearization* $x(\hat{u}+t\hat{h})$ *satisfies*

$$(\hat{u}+\hat{h})^\top x(\hat{u}+t\hat{h}) \leqslant \hat{\Pi}(\hat{u}+\hat{h}) - \kappa'\delta. \tag{21}$$

Proof. Use simplified notation: with $u^+ = \hat{u} + t\hat{h}$, set $x^+ := x(u^+)$ and $z := (\hat{u}+\hat{h})^\top x^+$.

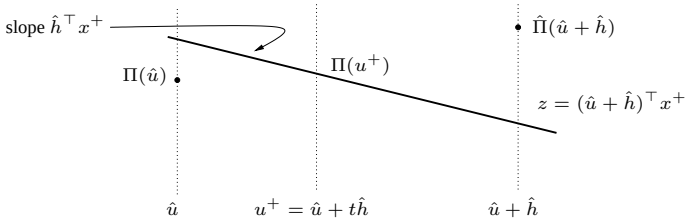

Figure 2: The new linearization passes under $\hat{\Pi}(\hat{u}+\hat{h})$

Because $x^+ \in \partial\Pi(u^+)$, we have by definition $\Pi(\hat{u}) \leqslant \Pi(u^+) - t\hat{h}x^+$; hence $\hat{h}^\top x^+ \leqslant [\Pi(u^+) - \Pi(\hat{u})]/t$, so that

$$\begin{aligned} z = \Pi(u^+) + (1-t)\hat{h}^\top x^+ \leqslant \Pi(u^+) + \tfrac{1-t}{t}[\Pi(u^+) - \Pi(\hat{u})] = \\ \tfrac{1}{t}\Pi(u^+) - \tfrac{1-t}{t}\Pi(\hat{u}). \end{aligned}$$

Now use (16) to bound $\Pi(u^+)$:

$$z \leqslant \frac{1}{t}\Pi(\hat{u}) + \hat{\Pi}(\hat{u}+\hat{h}) - \Pi(\hat{u}) - \kappa'\delta - \frac{1-t}{t}\Pi(\hat{u}),$$

which is just (21). □

The proof of the next result uses explicitly the fact that *all* linearizations are stored in the bundle: to accommodate the bundle compression alluded to at the end of Remark 1, a more sophisticated proof would be required, along the lines of [7, Thm. XV.3.2.4].

Theorem 2 *Suppose the stability center stops at some iteration* S*:* $\hat{u}^{s+1} = \hat{u}^s$ *for all* $s \geqslant S$*. Then* $\delta^s \to 0$ *and* $\hat{u}^S$ *is the optimal solution of* (9)*.*

Proof. The situation is as follows: at all iterations s following S, $\hat{u} = \hat{u}^S$, M and $y(\hat{u})$ are fixed; $\delta = \delta^s$ forms a nonincreasing sequence since (13) has more and more constraints; Lemma 4 then guarantees that $\hat{h} = \hat{h}^s$ is bounded. It follows that $u^{s+1} = \hat{u} + t^s h^s$ is also bounded, as lying in the segment $[\hat{u}, \hat{u}+\hat{h}^s]$; hence $x(u^s) \in \partial\Pi(u^s)$ is bounded ([7, Prop. VI.6.2.2]).

Write (21) and the definition (10) of $\hat{\Pi}$ at the s^{th} iteration: for all $s > S$ and all $r \leqslant s$,

$$(\hat{u} + \hat{h}^s)^\top x(u^{s+1}) + \kappa' \delta^s \leqslant \hat{\Pi}(\hat{u} + \hat{h}^s) \leqslant (\hat{u} + \hat{h}^s)^\top x(u^r),$$

so that

$$\kappa' \delta^s \leqslant (\hat{u} + \hat{h}^s)^\top [x(u^r) - x(u^{s+1})] \leqslant B|x(u^r) - x(u^{s+1})|,$$

where we have used the Cauchy-Schwarz inequality and B is a bound for $|\hat{u} + \hat{h}^s|$. Now assume $\delta^s \geqslant \varepsilon > 0$ for all s. Then

$$|x(u^r) - x(u^{s+1})| \geqslant \frac{\kappa' \varepsilon}{B} \quad \text{for all } s > S \text{ and all } r \leqslant s.$$

In words: around each $x(u^r)$, there is a ball of fixed radius $\kappa' \varepsilon / B$ which cannot contain any other x; because the x's are confined in a bounded set, this is impossible.

It follows that the monotone sequence δ^s tends to 0, pass to the limit in (20) to establish optimality of $\hat{u}$. □

8 Primal recovery

It is known that convergence of the dual algorithm has its counterpart concerning the primal problem. However, we solve here (9), while the dual of (1) is rather (5). The issue is therefore more delicate, especially when infinitely many ascent steps are performed; we analyze this case first.

Theorem 3 *The assumptions are those of Theorem 1. The subsequence $\{y(u^s)\}_{s \in \mathcal{S}}$ tends to the unique y-optimal solution of* (1)*. For $k = 1, \ldots, K$, the subsequences $\{\hat{x}^{k,s}\}_{s \in \mathcal{S}}$ are bounded and any of their cluster points makes up an x-optimal solution of* (1)*.*

Proof. We already know from Theorem 1 that $(\delta^s)_{\mathcal{S}}$, $(\hat{h}^s)_{\mathcal{S}}$ and $(\hat{g}^s)_{\mathcal{S}}$ tend to 0. We also know that $\hat{u}^s$ has a limit $\bar{u}$; therefore $y(\hat{u}^s) \to y(\bar{u})$ and we proceed to prove that $(\hat{x}^s)_{\mathcal{S}} \to y(\bar{u})$. Note that $(\hat{u}^s + \hat{h}^s)_{\mathcal{S}} \to \bar{u}$.

Define the set $J^* := \{j = 1, \ldots, n : \bar{u}_j = 1/c_j\}$ of artificial constraints that are active at $\bar{u}$.

– For $j \notin J^*$, $\bar{u}_j > 1/c_j$ so that $\hat{u}^s_j + \hat{h}^j_j > 1/c_j$ for $s \in \mathcal{S}$ large enough, The property $\nu^s \in \mathrm{N}_U(\hat{u}^s + \hat{h}^s)$ therefore implies $\nu^s_j = 0$, hence $\hat{x}^s_j = y_j(\hat{u}^s) - \hat{g}^s_j$ tends to $y_j(\bar{u})$.

– For $j \in J^*$, $y_j(\bar{u}) = 0$; hence $y_j(\hat{u}^s) \to 0$ and $\hat{x}^s_j \to 0$ because, from (18),

$$0 \leqslant \hat{x}^s_j = y_j(\hat{u}^s) - \hat{g}^s_j + \nu^s_j \leqslant y_j(\hat{u}^s) - \hat{g}^s_j \to 0.$$

Piecing together, we see that

$$(\hat{x}^s - y(\hat{u}^s))_{\mathcal{S}} \to 0. \tag{22}$$

Now write

$$\theta(\hat{u}^s) = \Phi(\hat{u}^s) + \Pi(\hat{u}^s) = f(y(\hat{u}^s)) - (\hat{u}^s)^\top y(\hat{u}^s) + \Pi(\hat{u}^s)$$

and pass to the limit for $u \in \mathcal{S}$:

$$\theta(\bar{u}) = f(y(\bar{u})) - \bar{u}^\top y(\bar{u}) + \Pi(\bar{u}).$$

But observe from (22) that

$$-\bar{u}^\top y(\bar{u}) + \Pi(\bar{u}) = \lim_{s \in \mathcal{S}} \big[-(\hat{u}^s)^\top \hat{x}^s + \Pi(\hat{u}^s)\big] = \lim_{s \in \mathcal{S}} \delta^s_x$$

where we have used the notation of Lemma 4. Since $(\delta^s_x \delta^s_x)_{\mathcal{S}} \to 0$,

$$\theta(\bar{u}) = f(y(\bar{u})). \tag{23}$$

Finally, the convergent sequence $(\hat{x}^s)_{\mathcal{S}}$ is bounded. Being nonnegative and summing up to $(\hat{x}^s)_{\mathcal{S}}$, the subsequences $(\hat{x}^{k,s})_{\mathcal{S}}$ are also bounded. Consider a cluster point: say, with $\mathcal{S}' \subset \mathcal{S}$, $(\hat{x}^{k,s})_{\mathcal{S}'} \to \bar{x}^k$ for $k = 1, \ldots, K$. The $\bar{x}^k$'s are feasible in (1) and they sum up to $y(\bar{u})$: $(\bar{x}, y(\bar{u}))$ makes a feasible point in (1). In view of (23), weak duality tells us that this point is primal optimal. □

The case of finitely many ascent steps is just easier, as $\hat{u}^s$ reaches its limit $\bar{u}$ for some finite s.

Theorem 4 *Suppose that the stability center stops at some iteration S. Then the conclusions of Theorem 3 hold, with $\mathcal{S}$ replaced by the whole sequence $S + 1, \ldots$ In fact, $\hat{u}^S$ is the optimal solution of* (9)*.*

Proof. Invoke Theorem 2: the whole sequences δ^s, $\hat{h}^s$ and $\hat{g}^s$ converge to 0. Then proceed exactly as for Theorem 3, with the simplifying property that $\hat{u}^s = \bar{u}$ for all $s > S$. □

Note that this result makes no assumption about primal feasibility... and yet proves primal existence. This has an interesting consequence:

Corollary 1 *Suppose that* (1) *has no feasible point. Then the dual function θ does not reach its maximum.*

Proof. Suppose for contradiction that (9) has an optimal solution $\bar{u}$. Initialize Algorithm 1 with $u^1 = \bar{u}$. There can be no descent step and Theorem 4 establishes the existence of an optimal primal solution. □

9 Numerical illustrations

To get a feeling of the numerical merits of our approach, we benchmark it on 16 test-problems against two implementations of its direct concurrent: the standard bundle method, which we briefly recall now.

If no attention is paid to its smoothness, Φ can be approximated via the linearizations $\bar{\Phi}_j^s(u_j) := \Phi_j(u_j) - (u_j - u_j^s)^\top y_j(u_j^s)$, instead of the quadratic functions $\tilde{\Phi}_j(u_j)$ of (12). Then Φ can be approximated just as Π by a polyhedral function (call it $\hat{\Phi}$) instead of the quadratic function $\tilde{\Phi}$ of (12); then a bundle method maximizes the resulting polyhedral approximation $\hat{\Phi} + \hat{\Pi}$ of θ, stabilized by a quadratic term $\frac{1}{2t}|u - \hat{u}|^2$; $t > 0$ is a parameter. Standard bundle methods maximize this approximation, and then manage the stability center $\hat{u}$ just as in Algorithm 1 (except that no backtracking is necessary).

An implementation in the spirit of the present paper uses the *individual* approximations $\Phi_j(u_j) \leqslant \hat{\Phi}_j(u_j) := \min_s \bar{\Phi}_j^s(u_j)$, thus replacing (13) by

$$\max_{h,\phi,\pi}\left\{\sum_{j=1}^n \phi_j + \sum_{k=1}^K \pi^k - \frac{1}{2t^S}|h|^2\right\} \quad \text{s.t.}$$
$$\left.\begin{array}{l} \phi_j \leqslant \bar{\Phi}_j^s(\hat{u}_j + h_j),\ j = 1,\dots,n\,, \\ \pi^k \leqslant (\hat{u}+h)^\top x^k(u^s),\ k = 1,\dots,K\,, \end{array}\right\} \quad s = 1,\dots,S,$$
$$h_j \geqslant \tfrac{1}{c_j} - \hat{u}_j,\ j = 1,\dots,n.$$

We will refer to this implementation as `bfull`, as it fully splits the approximation of θ. Now remember Remark 5: ignoring the summation in Φ, we can also use the *compound* (less accurate) polyhedral approximation $\Phi(u) \leqslant \min_s \sum_j \bar{\Phi}_j^s(u_j)$. In compensation, the resulting quadratic program simplifies to

$$\max_{h,\phi,\pi}\left\{\phi + \sum_{k=1}^K \pi^k - \frac{1}{2t^S}|h|^2\right\} \quad \text{s.t.}$$
$$\left.\begin{array}{l} \phi \leqslant \displaystyle\sum_{j=1}^n \bar{\Phi}_j^s(\hat{u}_j + h_j)\,, \\ \pi^k \leqslant (\hat{u}+h)^\top x^k(u^s),\ k = 1,\dots,K\,, \end{array}\right\} \quad s = 1,\dots,S,$$
$$h_j \geqslant \tfrac{1}{c_j} - \hat{u}_j,\ j = 1,\dots,n.$$

We also compare our method to this implementation, referred to as `bhalf`: it uses only a half of the splitting possibilities in θ.

With Algorithm 1 (referred to as `Ncp`), this makes three solvers, which have been implemented in C on a bi-processor Intel Xeon (30.06GHZ, 1.5GB RAM) under Linux operating system. Both standard bundle variants are home-made implementations of [11] (in particular for the t-management); Dijkstra's algorithm is used for the shortest path problems and the various QP are solved by Cplex 10.0. We use $\kappa = \kappa' = 0.1$ in Algorithm 1; the stopping criterion is $\underline{\delta} = 10^{-6}$ for `Ncp` and $\varepsilon = 10^{-6}$ for `bm`. We group the commodities by source nodes so that each computation of Π calls at most m times Dijkstra's algorithm.

Table 1 gives the result.

– The first group of 4 columns describes the 16 test problems, ranked by number K of commodities (recall that the number of dual variables is n). Problems 3,7,10 and 13 are the random networks already used in [15]. Problems 1 and 4 are nso22 and nso148, well known in the convex optimization community: see [5, 8]. The remaining test problems are based on actual networks.

Pb	m	n	K	iterations			total CPU			final θ		
				Ncp	bfull	bhalf	Ncp	bfull	bhalf	Ncp	bfull	bhalf
1	14	22	23	12 (88)	19	645	0.03	0.20	286.2	103.412019	103.412019	103.4119
2	19	68	30	5 (6)	11	16	0.02	0.14	0.08	8.994992	8.994991	8.994992
3	60	280	100	7 (95)	14	93	0.20	1.47	8.01	53.080767	53.080767	53.080767
4	61	148	122	7 (8)	24	167	1.07	8.84	61.09	151.926869	151.926869	151.926866
5	20	64	133	7 (8)	16	156	0.35	2.67	70.59	39.635462	39.635462	39.635460
6	122	332	162	9 (90)	21	309	0.61	5.23	495.4	276.321357	276.321357	276.321354
7	100	600	200	7 (96)	17	190	0.78	6.91	250.6	84.967483	84.967482	84.967481
8	30	72	335	7 (104)	24	1000*	0.13	3.56	5982.1	36.451716	36.451712	36.4516
9	21	68	420	7 (8)	42	151	5.62	88.66	856.5	68.838958	68.838958	68.83891
10	100	800	500	10 (304)	16	274	9.60	26.32	2061	139.096514	139.096509	139.0964
11	67	170	761	8 (133)	32	158	4.84	64.26	409.	109.895582	109.895579	109.895577
12	34	160	946	5 (6)	14	285	1.03	12.20	1650.2	19.566682	19.566681	19.566681
13	300	2000	1000	11 (319)	22	569	73.37	322.51	30564.	304.389462	304.389460	304.38940
14	48	198	1583	9 (80)	22	803	13.09	68.45	5h45	135.463182	135.463168	135.463124
15	81	188	2310	3 (4)	19	1000*	2.44	308.51	28h	41.791840	41.791826	41.75
16	122	342	2881	9 (73)	30	1000*	311.85	764.5	42h	242.714771	242.714733	238.7

Table 1: Comparison of Algorithm 1 (`Ncp`) with standard bundle (`bhalf` uses a compound approximation of Φ).

– The next group of columns gives the number of QP solved; column `Ncp` gives also the number of oracle calls (which depends on the number of backtracking steps; for standard bundle, one iteration is one QP resolution and one oracle call).
– The remaining columns are self-understanding. Computing times are in seconds; they are mainly indicative and would probably change substantially with the use of a specialized QP solver such as [10, 12, 3].

Note the difficulties of `bhalf` on Problems 8, 15 and 16, which could not reach the stopping criterion within 1000 iterations.

References

[1] F. Babonneau and J.P. Vial. Proximal-accpm with a nonlinear constraint and active set strategy to solve nonlinear multicommodity flow problems. *Mathematical Programming, forthcoming*, 2006.

[2] E. Cheney and A. Goldstein. Newton's method for convex programming and Tchebycheff approximations. *Numerische Mathematik*, 1:253–268, 1959.

[3] A. Frangioni. Solving semidefinite quadratic problems within nonsmooth optimization algorithms. *Computational Operational Research*, 23(11):1099–1118, 1996.

[4] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistic Quarterly*, 3:95–110, 1956.

[5] E.M. Gafni and D. Bertsekas. Two-metric projection methods for constrained optimization. *SIAM J. Control and Optimization*, 22:936–964, 1983.

[6] M. Gerla, L. Fratta, and L. Kleinrock. The flow deviation method: an approach to store-and-forward communication network design. *Networks*, 3:97–133, 1984.

[7] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1993. Two volumes.

[8] J.-L. Gofin J. Gondzio, R. Sarkissian and J.-Ph Vial. Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. *Mathematical Programming*, 76:131–154, 1996.

[9] J. E. Kelley. The cutting plane method for solving convex programs. *J. of the SIAM*, 8:703–712, 1960.

[10] K. C. Kiwiel. A dual method for certain positive semidefinite quadratic programming problems. *SIAM J. on Scientific and Statistical Computing*, 10(1):175–186, 1989.

[11] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1):105–122, 1990.

[12] K.C. Kiwiel. A Cholesky dual method for proximal piecewise linear programming. *Numerische Mathematik*, 68:325–340, 1994.

[13] C. Lemaréchal. Lagrangian relaxation. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization*, pages 112–156. Springer Verlag, Heidelberg, 2001.

[14] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Mathematical Programming*, 76(3):393–410, 1997.

[15] A. Ouorou, Ph. Mahey, and J-Ph. Vial. A survey of algorithms for convex multicommodity flow problems. *Management Science*, 47(1):126–147, 2000.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifi que
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399