# Warm start and $\varepsilon$-subgradients in cutting plane scheme for block-angular linear programs*

J. Gondzio† and J.-P. Vial

Logilab, HEC, Section of Management Studies, University of Geneva,

102 Bd Carl Vogt, CH-1211 Genève 4, Switzerland

Logilab Technical Report 1997.1
June 6, 1997

**Abstract**

This paper addresses the issues involved with an interior point-based decomposition applied to the solution of linear programs with a block-angular structure. Unlike classical decomposition schemes that use the simplex method to solve subproblems, the approach presented in this paper employs a primal-dual infeasible interior point method. The above-mentioned algorithm offers perfect measure of the distance to optimality, which is exploited to terminate the algorithm earlier (with a rather loose optimality tolerance) and to generate $\epsilon$-subgradients. In the decomposition scheme, subproblems are sequentially solved for varying objective functions. It is essential to be able to exploit the optimal solution of the previous problem when solving a subsequent one (with a modified objective). Warm start routine is described that deals with this problem.

The approach proposed has been implemented within the context of two optimization codes freely available for research use: the Analytic Center Cutting Plane Method (ACCPM) - interior point based decomposition algorithm and the Higher Order Primal-Dual Method (HOPDM) - general purpose interior point LP solver. Computational results are given to illustrate the potential advantages of the approach applied to the solution of very large structured linear programs.

**Key words.** Decomposition, cutting plane methods, interior point methods, warm start, block-angular linear programs.

---

*Supported by the Fonds National de la Recherche Scientifique Suisse, grant #12-42503.94.

†on leave from the Systems Research Institute, Polish Academy of Sciences, Newelska 6, 01-447 Warsaw, Poland.

# 1 Introduction

We are concerned in this paper with the solution of large scale block-angular linear programs (LP) by a decomposition approach. We assume that the subproblems are themselves very large linear programs. Consequently, we expect advantages from solving them with an interior-point method (see [2, 23]).

A decomposition scheme can be interpreted as a reformulation of the problem into a lower dimension space. The underlying convex optimization problem in this space has a non-differentiable objective function. Decomposition scheme works with the continuously improved approximations of this function. Indeed, it is well known that the optimal solution to the subproblems generate *subgradients* to this nondifferentiable function. We go a step further, i.e., we weaken the optimality requirements in subproblems and generate $\epsilon$-*subgradients*.

Throughout this paper we focus on decomposition for LP because the approach is popular and relevant. However, the ideas naturally extend to the more general classes of nondifferentiable optimization solved by the cutting plane scheme.

In any classical decomposition scheme, every subproblem is sequentially solved for varying objective functions (Dantzig-Wolfe decomposition [6]) or right hand sides (Benders decomposition [4]). In both cases, the linear problems associated with the subproblem change very little. It is thus crucial for an efficient approach to be able to exploit the optimal solution of the previous problem when solving a subsequent one (with a modified objective function or right hand side).

This can easily be done within the simplex method since the optimal basis of the previous problem remains primal feasible, when the objective is changed, or dual feasible, when the right hand side is altered. Hence the new modified problem can be solved efficiently with the primal simplex or with the dual simplex, respectively.

In this paper we show that reoptimization with the infeasible primal-dual interior point method can also be done efficiently. Whenever subproblems are large and/or difficult (possibly degenerate), reoptimization with the interior point method becomes an attractive alternative over reoptimization with the simplex method.

Interior point methods for linear programming offer a reliable control of the evolution of the duality gap. The infeasible primal-dual interior point method usually attains primal and dual feasibility much before optimality. For an explanation and a practical illustration of this phenomenon, see e.g., [20]. (Section 3.6, Table 7.) Thus early termination may yield a primal-dual feasible solution with guaranteed distance to optimality. We exploit this feature to generate $\epsilon$-subgradients.

Weaker optimality requirements in subproblems immediately translate into savings in the interior point iterations. In contrast, $\epsilon$-subgradients translate into less efficient cuts that may slow down the decomposition process. The question arises whether the savings in interior point iterations when solving the subproblems dominate the possible increase in decomposition iterations.

The answer is positive in the particular decomposition scheme that we apply; namely the analytic center cutting plane method [10, 11]. Instead of solving the restricted master problem to optimality, this cutting plane algorithm looks for the analytic center of the current localization set (see Section 3). Such a solution remains "central" in the localization set and is less sensitive to the $\epsilon$-displacements of the constraints, i.e., to the use of $\epsilon$-subgradients.

The approach presented in this paper has been implemented. We used two codes freely available for research use: the Analytic Center Cutting Plane Method (ACCPM) - interior point based decomposition algorithm and the Higher Order Primal-Dual Method (HOPDM) - general purpose interior point LP solver.

Computational results are given to illustrate the potential advantages of the approach applied to the solution of very large structured linear programs. These results confirm, in particular, that the reoptimizations with the infeasible primal-dual interior point method can be done efficiently, i.e., with an average of about 10 to 15 iterations regardless the size of the problem: the use of $\epsilon$-subgradients within the analytic center cutting plane method has little effect on convergence.

In this study we found that it is important to have a good starting point for the restricted master problem. It is also desirable to regularize somehow its dual variables. We refine earlier ideas and propose a simple trust region scheme that proves to be particularly useful when applied to the analytic center cutting plane method.

The paper is organized as follows. In Section 2, we discuss the particular properties of the block-angular linear programs and the decomposition approach applied to their solution. In Section 3 we recall the analytic center cutting plane method. We discuss modifications to handle $\epsilon$-subgradients, the convergence of the method and also the trust region scheme to control the dual variables in the restricted master program. In Section 4 we address the issues of warm start with the infeasible primal-dual method applied to solve the subproblems. In Section 5, we give the computational results on some large structured linear programs. Finally, in Section 6 we give our conclusions.

## 2 Block-angular linear programs

We are concerned in this paper with the solution of large scale block-angular linear programs of the following form

$$\begin{array}{lll} \text{maximize} & \sum_{i=1}^{p} \langle c_i, x_i \rangle & \\ \text{subject to} & \sum_{i=1}^{p} A_i x_i = a & \\ & B_i x_i = b_i, & i = 1, 2, \ldots, p, \\ & x_i \geq 0, & i = 1, 2, \ldots, p, \end{array} \tag{1}$$

where $c_i, x_i \in \mathcal{R}^{n_i}, i = 1, 2, ..., p$, $a \in \mathcal{R}^{m_0}$, $b_i \in \mathcal{R}^{m_i}, i = 1, 2, ..., p$, and all matrices $A_i, i = 1, 2, ..., p$, and $B_i, i = 1, 2, ..., p$, have appropriate dimensions. The Benders [4] technique of

variable partitioning applies to the dual formulation of this problem. It is thus equivalent to the approach discussed in the paper.

### 2.1 Partial Lagrangian

We associate with (1) the partial Lagrangian function

$$\begin{aligned} L(x,u) &= \textstyle\sum_{i=1}^{p}\langle c_i, x_i\rangle + \langle \sum_{i=1}^{p} A_i x_i - a, u\rangle \\ &= -\langle a, u\rangle + \textstyle\sum_{i=1}^{p}\langle c_i + A_i^T u, x_i\rangle. \end{aligned} \tag{2}$$

Assuming that problem (1) has an optimal solution, we may, by duality, replace it with

$$\min_u L(u), \tag{3}$$

where $L(u)$ is the optimal value of

$$\begin{array}{lll} \text{maximize} & -\langle a, u\rangle + \sum_{i=1}^{p}\langle c_i + A_i^T u, x_i\rangle & \\ \text{subject to} & B_i x_i = b_i, & i = 1,2,\ldots,p, \\ & x_i \geq 0, & i = 1,2,\ldots,p. \end{array} \tag{4}$$

In the usual parlance, Problem (3) is the master program while Problem (4) is the subproblem. Note that the feasible set of the subproblem is independent of $u$.

Let us observe that subproblem (4) is *separable*; in consequence, the partial Lagrangian is *additive*

$$L(u) = -\langle a, u\rangle + \sum_{i=1}^{p} L_i(u), \tag{5}$$

where, for $i = 1, 2, ..., p$

$$L_i(u) = \max_{x_i}\left\{\langle c_i + A_i^T u, x_i\rangle \,:\, B_i x_i = b,\, x_i \geq 0\right\}. \tag{6}$$

It has been noted [8] that *disaggregation* usually allows dramatic improvement in the performance of cutting plane methods. We also recall that subproblem (4) corresponds to the solution of $p$ independent linear programs (6) that could be solved in parallel.

### 2.2 Polyhedral approximation with the subgradients

The function $L(u)$ is convex but nondifferentiable. Solving the subproblem for different values of $u$ generates subgradients of $L_i$ that are used to construct a polyhedral approximation of the function $L$. Indeed, let $u = \bar{u}$ and assume that every subproblem (6) has an optimal solution $\bar{x}_i$ for $i = 1, 2, \ldots, p$. Thus, for any arbitrary $u$, we have the subgradient inequality

$$L_i(u) \geq \langle c_i + A_i^T u, \bar{x}_i\rangle = \langle A_i \bar{x}_i, u - \bar{u}\rangle + L_i(\bar{u}). \tag{7}$$

Each function $L_i$ has its own support $\xi_i = A_i\bar{x}_i$ at the point $\bar{u}$.

In this disaggregated formulation we add a cut for each $i = 1, 2, ..., p$. To this end, we introduce variables $z_i, i = 0, 1, 2, ..., p$, and define the LP relaxation in the extended space $\mathcal{R}^{m_0+p+1}$ by

$$\begin{array}{ll} \text{minimize} & z_0 \\ \text{subject to} & z_0 = -\langle a, u\rangle + \sum_{i=1}^{p} z_i, \\ & z_i \geq \langle \xi_i^k, u - \bar{u}^k\rangle + L_i(\bar{u}^k), \quad i = 1, 2, \ldots, p, \quad k = 1, 2, \ldots\kappa, \end{array} \tag{8}$$

where $k$ is the index of a call to the subproblem.

We denote $z = (z_1, z_2, \ldots, z_p)$ and associate with (8) the *localization set*

$$\begin{array}{rl} \mathcal{L} = & \{(z_0, z, u) : z_0 \leq \theta^\kappa, \ z_0 = -\langle a, u\rangle + \sum_{i=1}^{p} z_i, \\ & z_i \geq \langle \xi_i^k, u - \bar{u}^k\rangle + L_i(\bar{u}^k), \quad i = 1, 2, ..., p, \quad k = 1, 2, \ldots\kappa\}, \end{array} \tag{9}$$

where $\theta^\kappa$ is the current upper bound, i.e., the best value of $L(u)$ up to iteration $\kappa$

$$\theta^\kappa = \min_{k=1,2,\ldots,\kappa} L(\bar{u}^k). \tag{10}$$

### 2.3 A relaxed polyhedral approximation with the $\epsilon$-subgradients

Assume now that the subproblem is solved with an $\epsilon$-precision, $\epsilon > 0$. That is, we generate a point $\bar{x}_i$ that is feasible for the subproblems $i = 1, 2, \ldots, p$ with

$$\langle c_i + A_i^T \bar{u}, \bar{x}_i\rangle \geq L_i(\bar{u}) - \epsilon. \tag{11}$$

Thus for an arbitrary $u$, we have the $\epsilon$-subgradient inequality

$$L_i(u) \geq \langle A_i\bar{x}_i, u - \bar{u}\rangle + L_i(\bar{u}) - \epsilon. \tag{12}$$

The $\epsilon$-subgradients can be used to define a weaker LP relaxation in the extended space $\mathcal{R}^{m_0+p+1}$ by

$$\begin{array}{ll} \text{minimize} & z_0 \\ \text{subject to} & z_0 = -\langle a, u\rangle + \sum_{i=1}^{p} z_i, \\ & z_i \geq \langle \xi_i^k, u - \bar{u}^k\rangle + L_i(\bar{u}^k) - \epsilon, \quad i = 1, 2, \ldots, p, \quad k = 1, 2, \ldots\kappa, \end{array} \tag{13}$$

and a corresponding larger localization set

$$\begin{array}{rl} \mathcal{L}_\epsilon = & \{(z_0, z, u) : z_0 \leq \theta_\epsilon^\kappa, \ z_0 = -\langle a, u\rangle + \sum_{i=1}^{p} z_i, \\ & z_i \geq \langle \xi_i^k, u - \bar{u}^k\rangle + L_i(\bar{u}^k) - \epsilon, \quad i = 1, 2, ..., p, \quad k = 1, 2, \ldots\kappa\}. \end{array} \tag{14}$$

The accuracy level $\epsilon$ need not remain fixed within the course of the algorithm. As the cutting plane scheme approaches a solution, the level of accuracy with which the subproblem is solved should increase. Let $\epsilon^k$ be the accuracy at the $k$th iteration of the cutting plane method, $k = 1, 2, \ldots, \kappa$. The $\epsilon$-subgradient approach yields a weaker upper bound

$$\theta_\epsilon^\kappa = \min_{k=1,2,\ldots,\kappa} (-\langle a, \bar{u}^k \rangle + \sum_{i=1}^{p} (L_i(\bar{u}^k) + \epsilon^k)) = \min_{k=1,2,\ldots,\kappa} (L(\bar{u}^k) + p\epsilon^k). \tag{15}$$

However, the $\epsilon^k$ could vary for each subproblem.

# 3 Analytic center cutting plane method

We assume throughout the paper that we are given an initial box $l \leq u \leq h$ and that the partial Lagrangian problem to solve is

$$\min_u \{L(u) \, : \, l \leq u \leq h\}.$$

## 3.1 ACCPM with exact subgradients

Let $\{u^k\}_{k=1}^\kappa$ be a sequence of query points at which subproblem $i$, $i = 1, 2, \ldots, p$, returns a subgradient inequality of the form

$$-\langle \xi_i^k, u \rangle + z_i \geq \gamma_i^k, \quad k = 1, 2, \ldots, \kappa. \tag{16}$$

By identification with (7) and (8),

$$\xi_i^k = A_i \bar{x}_i^k, \text{ and } \gamma_i^k = \langle c_i, \bar{x}_i^k \rangle. \tag{17}$$

The best recorded value for the objective $L(u)$ is

$$\theta^\kappa = \min_{k \leq \kappa} \{L(u^k) = \langle c_i, \bar{x}_i^k \rangle + \langle \sum_{i=1}^{p} A_i \bar{x}_i^k - a, u^k \rangle\}.$$

We can gather the valid inequalities

$$z_0 \leq \sum_{i=1}^{p} \langle c_i, \bar{x}_i^k \rangle + \langle \sum_{i=1}^{p} A_i \bar{x}_i^k - a, u^k \rangle, \quad k = 1, 2, \ldots, \kappa \tag{18}$$

and add them to the current set of previous subgradient inequalities. We then build the current localization set $\mathcal{L}^\kappa$ which is formally defined by

$$\begin{array}{rll} z_0 & \leq \sum_{i=1}^{p} \langle c_i, \bar{x}_i^k \rangle + \langle \sum_{i=1}^{p} A_i \bar{x}_i^k - a, u^k \rangle, & k = 1, \ldots, \kappa \\ -\langle \xi_i^k, u \rangle + z_i & \geq \gamma_i^k, \quad i = 1, \ldots p, & k = 1, \ldots, \kappa \\ l & \leq u \leq \; h. & \end{array} \tag{19}$$

The localization set is a subset of the epigraph subspace of $L$ that contains the optimal solution $(L(u^*), z^*, u^*)$. We assume that this set has a nonempty interior. Define the analytic center of $\mathcal{L}^\kappa$ as the unique solution of the strictly convex problem

$$\max_{(z_0,z,u)\in\mathcal{L}^\kappa} \{\sum_{k=1}^{\kappa} \Big(\log(\langle\sum_{i=1}^p A_i\bar{x}_i^k - a, u^k\rangle + \sum_{i=1}^p\langle c_i, \bar{x}_i^k\rangle - z_0) + \sum_{i=1}^p \log(z_i - \gamma_i^k - \langle\xi_i^k, u\rangle)\Big) + \log(h-u) + \log(u-l)\}. \tag{20}$$

In order to simplify the definition of the analytic center, we replace the first $\kappa$ terms in the objective of Problem (20) by $\kappa\log(\theta^\kappa - z_0)$. This is equivalent to repeating $\kappa$-times the constraint associated with the best recorded value. A similar rule is used in the standard path-following method of Renegar [26] for linear programming.

The analytic center cutting plane method generates a sequence of localization sets and analytic centers which are query points for the oracle. At each iteration of the algorithm, the subproblem generates $p+1$ cuts. At least one of them cuts off the query point.

### 3.2 Convergence of ACCPM with $\varepsilon$-subgradients

The convergence analysis of the analytic center cutting plane scheme has been performed by Nesterov [25] and the followers [1, 12, 13, 22]. In practice, the subproblem may generate very deep cuts. It is shown in [14] that, in the case of a single cut generation scheme, very deep cuts does not alter the convergence estimate. Naturally, $\varepsilon$-subgradients produce less deep cuts. If one can make sure that the cut at lease cuts off the current query point, all the above analyses go through and the convergence result holds.

Let $\varepsilon^k$ be the accuracy of the solution of each subproblem at iteration $k$. The computation of the best recorded value is modified as follows:

$$\theta_\varepsilon^\kappa = \min_{k\le\kappa}\{\sum_{i=1}^p\langle c_i, \bar{x}_i\rangle + \langle\sum_{i=1}^p A_i\bar{x}_i^k - a, u^k\rangle + p\varepsilon^k\}. \tag{21}$$

If $\varepsilon^k$ is too large, it is no longer possible to guarantee that at least one of the $p+1$ cuts generated by the subproblem is deep. Since testing the depth of the cut is immediate, the subproblem can be called again to achieve a new better accuracy. That is, $\varepsilon^k$ is decreased by some factor and the subproblem resumes optimization from its current solution. With this technique, one can make sure that the cut is never shallow, and the convergence analysis holds.

### 3.3 Trust region control of $u$

ACCPM, just as other cutting plane methods (CPMs), must be initiated with bounds on the variables. It is customary to take the $l_\infty$ box

$$l \le u \le h,$$

though other choices are conceivable. (See Nesterov [25] for a theoretical analysis with an $l_2$ box.) The box reflects the a priori knowledge of the user about the optimal solution. Those bounds may be derived on a theoretical basis, and thus be certain, or they can incorporate some fuzzier knowledge.

Tight bounds are liable to allow the algorithm to achieve faster convergence. In the meantime, a poor estimation of these bounds may slow down the algorithm or even prevent it from reaching the optimal solution. To balance the negative effect of a poor initial estimation, it is natural to adopt a trust region scheme. However, this scheme must be adapted to the specific behavior of the analytic center cutting plane method.

We define the trust region (TR) as the $l_\infty$ box

$$\lambda \leq u \leq \eta,$$

with an obvious condition that this box matches the certain information

$$l \leq \lambda < \eta \leq h.$$

Note that one may well have $l_j = -\infty$ or $h_j = +\infty$. If the algorithm starts to generate points at the boundary of the TR, or close to it, then there is an indication that the trust region should be expanded or shifted. Note that in the commonly used CPM of Kelley [21], in the presence of relatively few cutting planes, the method will usually yield query point on the boundary of the trust region. Therefore, for general CPMs, having a query point on the boundary of the TR does not always reflect the need to expand the TR.

In the analytic center approach the situation is more favorable, as the barrier prevents the algorithm from approaching too fast the boundary of the TR. Let us describe the mechanism we adopted. It is based on empirical observations. Essentially, the rules are as follows.

Let $\delta$ and $\beta$ be two positive parameters. If $u_i - \lambda_i \leq \delta(\eta_i - \lambda_i)$, then $\lambda_i$ is replaced with

$$\lambda_i^+ = \max\{l_i,\, \lambda_i - \beta(\eta_i - \lambda_i)\}.$$

Analogously, if $\eta_i - u_i \leq \delta(\eta_i - \lambda_i)$, then $\eta_i$ is replaced with

$$\eta_i^+ = \min\{h_i,\, \eta_i + \beta(\eta_i - \lambda_i)\}.$$

The default values for $\delta$ and $\beta$ are $\delta = 0.05$ and $\beta = 10$. Note that this process always increases the box size.

Practical considerations led us to tighten the rules in the early stage of the algorithm. Indeed, in this stage, the cutting planes may not suffice to restrict by themselves the query point to the TR. It may be desirable to collect more information within the TR before modifying it. Therefore, we adopt the rule that the trust region remains unchanged for the first $k$ iterations. The default value is $k = 5$. Finally, the default values for $l_j$ and $h_j$ are $-10^6$ and $10^6$, respectively.

The initial guess for the trust region is very much problem dependent. For problems with a primal block-angular structure, one may interpret the coupling constraints as the restriction on

common resources. If some initial partition of these resources among subproblems is available, the subproblems can be solved independently to produce their independent estimations of the dual prices. The average of these estimates can be a good initial point for the CPM while some interval around this point can serve as the initial trust region. More generally, any estimate of dual variables corresponding to the coupling constraints can be taken as the initial point, and any interval that contains it can be an initial trust region.

# 4 Warm start

Let us now focus the attention on the solution of subproblem (4). It consists of the solution of $p$ independent linear programs (6) for $i = 1, 2, \ldots, p$. Every linear program is repeatedly solved in subsequent iterations of the cutting plane method. Its primal feasible set remains unchanged but its objective depends on the query point $u = u^k$

$$f_i = c_i + A_i^T u, \quad i = 1, 2, \ldots, p. \tag{22}$$

Subsequent problems stay closely related, especially, when the cutting plane algorithm approaches the optimum. Then $u$ approaches its optimal value and $f_i$ varies very little from iteration to iteration. Any efficient implementation of the decomposition method has to take advantage of this fact and exploit the optimal solution of the previous problem when solving subsequent one.

If the simplex method is used to solve the subproblem, the optimal basis of the previous problem remains primal feasible to the new one and is used as an initial basis for solving subsequent problems. This approach is efficient unless the subproblem is degenerate, in which case the convergence of the simplex method can become quite slow.

If the subproblem is large and/or difficult, possibly degenerate, the simplex method may converge slowly or even stall. Then solving the problem with the interior point method, even from scratch, becomes an attractive alternative. Naturally, we want to specialize the method further and exploit the solution of the previous problem when solving subsequent ones.

The technique of warm start is at the core of the analytic center cutting plane method. As the theory [27] and the computational experience [8, 18] show, an analytic center that solves the master program to some degree of accuracy is an excellent starting point for adding or deleting constraints to the master program.

Gondzio [17] combined the use of approximate analytic centers with the use of specially chosen targets in the space of complementarity products and the use of generalized centrality correctors [16] and applied it within the infeasible primal-dual interior point method. Although the primary aim of this approach was to embed the primal-dual method in the column generation scheme [19], it turned out that this warm start routine can easily be adopted to perform reoptimizations in other applications. We recall its fundamental ideas below. To keep the notation

simpler, we drop subproblem index $i$ and consider the following (primal) subproblem

$$\begin{aligned} \text{minimize} \quad & f^T x \\ \text{subject to} \quad & Bx = b, \\ & x \geq 0, \end{aligned} \tag{23}$$

where $f, x \in \mathcal{R}^n, b \in \mathcal{R}^m$ and $B \in \mathcal{R}^{m \times n}$. With this linear program we associate the dual

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & B^T y + s = f, \\ & s \geq 0, \end{aligned} \tag{24}$$

where $y \in \mathcal{R}^m$ and $s \in \mathcal{R}^n$. Having replaced the variable's nonnegativity constraints with the logarithmic barrier function, we write down the following Lagrangian

$$L(x, y, s) = f^T x - \mu \sum_{j=1}^{n} \ln x_j - y(Bx - b), \tag{25}$$

where $\mu \geq 0$ is the barrier parameter. The associated first order optimality conditions give

$$\begin{aligned} Bx &= b, \\ B^T y + s &= f, \\ XSe_n &= \mu e_n, \end{aligned} \tag{26}$$

where $X$ and $S$ are diagonal matrices with the elements $x_j$ and $s_j$, respectively and $e_n$ is the $n$-vector whose elements are all equal to one. If the feasible sets of the primal and dual problems (23) and (24) have nonempty interiors, then for any $\mu > 0$, there exists a unique point $(x(\mu), y(\mu), s(\mu))$ $x(\mu) > 0, s(\mu) > 0$ that satisfies (26). This point is called a $\mu$-center or an analytic center.

The gap between these primal and dual analytic centers depends on $\mu$ and measures the distance to optimality

$$f^T x - b^T y = f^T x - x^T A^T y = x^T (f - A^T y) = x^T s = n\mu. \tag{27}$$

Classical primal-dual method for linear programming starts from infeasible solutions that may, in general, violate all three equations in (26). The method fixes $\mu$ and executes one damped Newton step towards the $\mu$-center. Regardless of how successful this step was, the infeasible primal-dual method updates the barrier parameter, usually by decreasing it. The reader interested in the issues of its implementation may consult the surveys [2, 23] or the book of Wright [28]. In contrast to the nonlinear conditions $x_j s_j = \mu$ in (26), the primal and dual feasibility constraints are linear. Newton's method works with a linear model of the system of equations (26), in which the first two equations are linear. At each step of the algorithm, the

primal and dual infeasibilities are thus reduced proportionally to the stepsizes. Therefore, the method produces very quickly a primal and dual feasible solution.

Because the infeasible primal-dual method executes only one Newton step before altering (usually reducing) $\mu$ in the subsequent iteration, the primal and dual iterates may seriously violate conditions (26). For the sake of efficiency, however, it is preferable that all complementarity products stay close to $\mu$. The method of [16] controls the ratio between the largest and the smallest complementarity product

$$v_{max} = \max_j x_j s_j \quad \text{and} \quad v_{min} = \min_j x_j s_j, \tag{28}$$

through the use of special centrality correctors.

Note that the infeasible primal-dual interior point method can be modified easily to attain an approximate analytic center corresponding to a prescribed barrier parameter $\mu_0$. This modification resolves to disabling the decrease of $\mu$ below $\mu_0$ and to the use of centrality correctors [16] instead of Mehrotra's predictor-corrector technique when close to the analytic center corresponding to $\mu_0$ (see [17] for details).

Standard primal-dual methods for LP continue iterating until the relative duality gap drops below a prescribed optimality tolerance $\epsilon_{opt}$, i.e.

$$\frac{|f^T x - b^T y|}{|b^T y| + 1} < \epsilon_{opt}, \tag{29}$$

where $\epsilon_{opt}$ is usually set between $10^{-8}$ and $10^{-6}$. The first reliable (though still very rough) estimate of the optimal objective function value $f_0 = |f^T x|$ is known already when (29) is satisfied for e.g. $\epsilon_{opt} = 10^{-1}$ and current primal and dual solutions approach feasibility. This rough estimate can already be used to fix the barrier

$$\mu_0 = \epsilon_0 \frac{f_0}{n}, \tag{30}$$

such that the corresponding $\mu_0$-center is an optimal solution with the relative tolerance $\epsilon_0$ (see eqn. (27)).

In our implementation the desired relative optimality tolerance $\epsilon_0$ of the approximate analytic center is the only parameter passed to the primal-dual method. The algorithm continues with the classical primal-dual steps until it approaches the primal and dual feasibility and reaches the rough approximation of the optimal solution corresponding to the relative tolerance $\epsilon = \min\{10^{-1}, 100\epsilon_0\}$. At this point the final barrier parameter $\mu_0$ is determined by equation (30). The remaining primal-dual steps do not use Mehrotra's predictor-corrector technique; they use centrality correctors [16]. The algorithm continues until

$$v_{max} < 10\mu \quad \text{and} \quad v_{min} > 0.1\mu,$$

which is considered to be the condition of the sufficient quality of the approximate $\mu_0$-center.

The solution of subproblem $i = 1, 2, \ldots, p$ at the query point $u = u^k$ is split into two steps. In the first one, an $\epsilon_0$-optimal approximate analytic center is constructed. The relative

optimality tolerance is significantly less demanding then the desired final accuracy $\epsilon$. In our implementation $\epsilon_0$ varies from $10^{-1}$ at the beginning of the cutting plane method (when violent changes of $u$ at (22) are expected) to $10^{-4}$ when $u$ approaches its optimal value.

The analytic center corresponding to the relative accuracy $\epsilon_0$ is saved to be used as an initial point when subsequent subproblem will be solved (at the query point $u = u^{k+1}$). Once this point is saved, the second step of the subproblem's solution is executed: the primal-dual algorithm is continued, this time with the classical termination criteria (29), until the desired accuracy $\epsilon$ ($\epsilon < \epsilon_0$) in the subproblem is reached.

Summing up, our warm start routine does not really execute reoptimization in a strict sense. It uses the approximate analytic center (corresponding to rather loose accuracy $\epsilon_0$) of subproblem $i$ called at $u = u^k$ as a starting point when looking for the analytic center (again corresponding to the loose accuracy $\epsilon_0$) of subproblem $i$ called at the query point $u = u^{k+1}$. The use of loose accuracy $\epsilon_0$ is a price to pay for the ability to re-optimize after expected violent changes of $u$ in subsequent iterations of the cutting plane method. We cannot allow the $\mu_0$-center to approach the optimal solution too much when we predict that the subsequent problem may have a completely different objective function. Obviously, whenever modifications of $u$ become less important in subsequent iterations of the cutting plane method and the objective $f$ in (22) varies less, we may be more demanding about the accuracy $\epsilon_0$ of the analytic center used in warm start.

The true optimization follows the update of the analytic center. This step consists in improving the accuracy of the optimal solution of subproblem from $\epsilon_0$, that corresponds to the analytic center saved, to the required $\epsilon$ ($\epsilon < \epsilon_0$) for which an $\epsilon$-subgradient is returned. This second phase of the optimization is usually executed in only a few interior point iterations. There are at least two reasons for that. The first comes from the fact that $\epsilon$ is usually only from $10^{-4}$ to $10^{-2}$ smaller than $\epsilon_0$. The second is a perfect centrality of the $\epsilon_0$-center: the step in a primal-dual direction computed at this point brings often considerable reduction of the duality gap.

## 5 Computational experience

The implementation of the method presented in this paper uses two interior point codes freely available for research use: the Analytic Center Cutting Plane Method (ACCPM) [18] and the Higher Order Primal-Dual Method (HOPDM) [15]. Both codes can be retrieved from the Web site `http://ecolu-info.unige.ch/~logilab/software/`.

Our implementation uses 3 programming languages: ACCPM is written in FORTRAN and C++, and HOPDM is written in FORTRAN and ANSI C. It has been developed on IBM RS/6000 machine (133MHz, 128MB of RAM, type 7011, model 25T). The program has been compiled with the IBM's compilers: xlf, xlc and xlC of FORTRAN, C and C++, respectively and run under AIX 4.2 operating system.

Table 1: The effect of the use of trust region technique in ACCPM.

| Problem | No trust region | | | With trust region | | |
|---|---|---|---|---|---|---|
| | Outer | Inner | IPM | Outer | Inner | IPM |
| MARKAL | 36 | 126 | 2259 | 28 | 72 | 1864 |
| MACHINES | 16 | 85 | 1486 | 13 | 73 | 1149 |
| ENERGY-S | 27 | 96 | 1843 | 11 | 34 | 627 |

The main objective of our development is the solution of very large block-angular structured linear programs. By very large we mean problems in which the subproblems are large enough to take advantage of the interior point technology. Naturally, we also want to learn how the trust region technique works within ACCPM, how the use of $\epsilon$-subgradients influences the convergence of the central cutting plane method, and how the warm start routine performs in practice. All these issues will be addressed in the following subsections.

### 5.1 Trust region in ACCPM

We start the analysis of the computational results with a closer look at the behavior of the trust region technique within the analytic center cutting plane method. We solve three real-life medium-scale block-angular linear programs: MARKAL, MACHINES and ENERGY-S. The statistics and the origin of these problems are given in Table 4 in Section 5.2.

We compare in Table 1 two different variants of ACCPM: the one without the trust region technique (bounds in this case are set to $l = -10^4$ and $h = +10^4$, respectively), and the one applying the trust region technique. Initial trust region is a box of size $10^2$ around the initial estimate of $u$. In both cases the subproblems are always solved to optimality, which means that exact subgradients are used. In Table 1 we report the number of outer iterations (subproblem calls), the number of inner iterations, i.e., Newton steps made by the projective algorithm to find the analytic center of the localization set within ACCPM, and the total number of interior point iterations executed when solving the subproblems with the infeasible primal-dual method.

To obtain an initial estimate, we used the resource allocation technique explained at the end of Section 3. However, our implementation of this technique is very crude: we partitioned the resources uniformly between the subproblems. This gave very good results in a case of ENERGY-S problem. Unfortunately, the MARKAL problem has very unbalanced subproblems and it needs more sophisticated partition of resources. The estimates obtained with the uniform partition were quite poor. This translates into the poorer results, though the TR technique remains beneficial. In MACHINES, the coupling constraints are all but one balance equations with a zero right hand side. The partitioning resources technique is then even more arbitrary choice. Nonetheless, we still observe an improvement.

Table 2: Statistics of the problems (only one subproblem).

| Problem | Constraints | | | Variables |
|---|---|---|---|---|
| | total | coupling(active) | subproblem | |
| AFIRO | 27 | 2 (1) | 25 | 32 |
| ADLITTLE | 56 | 2 (2) | 54 | 97 |
| KB2 | 43 | 9 (9) | 34 | 41 |
| BRANDY | 220 | 3 (2) | 217 | 249 |
| PILOT4 | 410 | 6 (4) | 404 | 1000 |

Since in most real life applications the user either knows or has a way to compute an initial estimate of the dual variables, we can safely state that the TR technique leads to significant improvements. It stabilizes the analytic center cutting plane method, and accelerates the optimization. Another advantage of this technique is the reduction of the changes of the objective function in the subproblem: this improves the conditions in which warm start is employed and translates into the reduction of the number of the primal-dual iterations.

### 5.2 ACCPM with $\epsilon$-subgradients

The efficiency of ACCPM with $\varepsilon$-subgradients is determined by two opposite factors. The computation of an $\varepsilon$-optimal solution of the subproblems is likely to be far less demanding than computing the optimal solution. On the other hand, $\varepsilon$-subgradient inequalities are less powerful: localization sets are less restrictive so the cutting plane algorithm may require more iterations to converge.

To analyze these effects, we first solve a problem with a unique subproblem ($p = 1$). This amounts to a fully aggregated subproblem. As it was proved in [8], this setting is the most demanding one for a cutting plane method. Therefore, the impact of $\varepsilon$-subgradients is amplified. For this experiment we used the standard LP problems from the Netlib collection [9]. We selected a small subset of constraints, some of them known to be active at the optimum, and used them as the coupling constraints in (1). The other constraints define the feasible set of the subproblem. The nondifferentiable problem created by the partial Lagrangian relaxation is rather difficult and is by no means the proper way to solve the original problem. However, it generates a very demanding and illustrative test for cutting plane methods.

In Table 2 we give statistics of decomposed problems. The columns contain: the problem name, the number of constraints, the number of coupling rows $m_0$ and, in parenthesis, the number of rows that are active at the optimum, the number of subproblem's constraints $m_1$ and the number of variables.

Table 3: The effect of the use of $\varepsilon$-subgradients in ACCPM.

| Problem | Exact subgradients | | | $\varepsilon$-subgradients | | |
|---|---|---|---|---|---|---|
| | Outer | Inner | IPM | Outer | Inner | IPM |
| AFIRO | 13 | 32 | 108 | 19 | 40 | 61 |
| ADLITTLE | 21 | 46 | 167 | 21 | 48 | 109 |
| KB2 | 44 | 68 | 351 | 60 | 91 | 297 |
| BRANDY | 15 | 29 | 303 | 16 | 30 | 161 |
| PILOT4 | 40 | 71 | 1089 | 43 | 76 | 643 |

Table 3 collects results of the comparison of two variants of ACCPM applied to the solution of these problems: the method with exact subgradients and the method with $\varepsilon$-subgradients. When exact subgradients are used, at each iteration the subproblem is solved to the relative accuracy $\varepsilon = 10^{-8}$. When $\varepsilon$-subgradients are used, the relative accuracy $\varepsilon$ varies from $10^{-3}$ at the beginning to $10^{-8}$ at the end. The exact value is chosen by the formula

$$\varepsilon = \min\{10^{-3}, 10^{-2}r\}, \tag{31}$$

where $r$ is the current relative accuracy reached by the cutting plane algorithm. ACCPM terminates when the relative precision $r = 10^{-6}$ is achieved. Table 3 reports for the two versions under study: the number of outer iterations (subproblem calls), the number of IPM iterations required to reach the analytic centers within ACCPM (inner iterations), and the total number of IPM iterations required to reach the $\varepsilon$-optimal solution in the subproblem.

Table 4: Statistics of the decomposable problems.

| Problem | Master | | | Subproblems | | | Balance | |
|---|---|---|---|---|---|---|---|---|
| | rows | active | GUB | rows | cols | nonz | min rows | max rows |
| AFIRO | 2 | 1 | 4 | 100 | 128 | 320 | 25 | 25 |
| ADLITTLE | 3 | 3 | 4 | 212 | 388 | 1804 | 53 | 53 |
| KB2 | 9 | 9 | 4 | 136 | 164 | 992 | 34 | 34 |
| MARKAL | 7 | 7 | 3 | 17393 | 22798 | 204108 | 3859 | 7202 |
| MACHINES | 17 | 17 | 16 | 1600 | 25864 | 321250 | 100 | 100 |
| ENERGY-S | 5 | 5 | 4 | 14364 | 27000 | 85440 | 3591 | 3591 |

The analysis of the results collected in Table 3 indicates that the use of $\varepsilon$-subgradients causes relatively little increase of the number of outer iterations and remarkable savings in the

Table 5: Solution of block-angular LPs with ACCPM.

| Problem | Exact subgradients | | | $\varepsilon$-subgradients | | |
|---|---|---|---|---|---|---|
| | Outer | Inner | IPM | Outer | Inner | IPM |
| AFIRO | 14 | 37 | 356 | 17 | 44 | 281 |
| ADLITTLE | 13 | 41 | 367 | 15 | 49 | 309 |
| KB2 | 38 | 106 | 1051 | 43 | 124 | 992 |
| MARKAL | 28 | 72 | 1864 | 31 | 91 | 1450 |
| MACHINES | 13 | 73 | 1149 | 14 | 79 | 937 |
| ENERGY-S | 11 | 34 | 627 | 12 | 39 | 491 |

overall numbers of IPM iterations performed in subproblem optimizations. Since the latter number is the dominating factor in the overall computational effort, we may conclude that the use of $\varepsilon$-subgradients is an advantageous technique in this particular application.

We repeated the same experiment for a set of real decomposable linear programs with the number of subproblems varying from 3 to 16. The statistics of these problems are collected in Table 4. Its columns 2, 3 and 4 specify for every problem: the size of the master, i.e., the number of coupling constraints $m_0$, the number of rows that are active at the optimum, and the number of GUB rows which is the same as the number of subproblems $p$, respectively. The following columns characterize the subproblems. They report the total number of rows, columns and nonzero elements in all $p$ subproblems. Several problems have well balanced subproblems, the other have not. The last two columns give an insight into this balance: they specify the minimum and the maximum row size of the subproblems. The first 3 problems have been created by random modifications of linear programs from the Netlib collection in which the first $m_0$ constraints were included into the master. The remaining problems are structured LPs arising from real-life applications. MARKAL is the multi-country, multi-period energy planning model with the global constraint limiting the emission of $CO_2$ [3]. MACHINES is the decentralized version of Markov decision model in manufacturing [7]. ENERGY-S is the energy planning problem developed at IIASA [24].

Table 5 reports the results of the solution of all problems from Table 4 with two variants of ACCPM: the method with exact subgradients and the method with $\varepsilon$-subgradients. For both variants of the decomposition, we report: the number of outer iterations, the number of inner iterations in ACCPM and the total number of IPM iterations required to reach the $\varepsilon$-optimal solutions in all $p$ subproblems.

The analysis of the results collected in Table 5 substantiates our conclusion on the merits of $\varepsilon$-subgradients in ACCPM. The optimization of subproblems to a less demanding accuracy leads to remarkable savings in the number of IPM iterations. This effect is more important

Table 6: Single subproblem's statistics.

| Original size | | | Cplex presolve | | | HOPDM presolve | | |
|---|---|---|---|---|---|---|---|---|
| m | n | nonz | m | n | nonz | m | n | nonz |
| 41580 | 33643 | 188036 | 35057 | 31620 | 144941 | 33248 | 30506 | 141975 |

Table 7: Solution of the single subproblem.

| Cplex 3.0, simplex | | Cplex 3.0, barrier | | HOPDM 2.20 | |
|---|---|---|---|---|---|
| iters | time | iters | time | iters | time |
| 179464 | 41397 | 64 | 2380 | 50 | 2880 |

than the induced increase of the number of outer iterations (calls to subproblem). We remind the reader that both variants work with the warm start option for the subproblems. For a comparison with the cold start performance, see Table 9.

Finally, we solve large-scale problem ENERGY-L arising from the energy planning model [24]. This problem is particularly interesting due to its size and the difficulty it causes to the simplex method. The characteristics of a single subproblem — the numbers of rows, columns and nonzero entries before and after default presolve analysis of Cplex 3.0 [5] and HOPDM 2.20 [15] — are given in Table 6. We solved such a single subproblem with the simplex method of Cplex and with the two interior point codes: Cplex Barrier and HOPDM. The results of these runs are collected in Table 7. One can see from these results that the interior point codes solve this problem about 15 times faster than the simplex method. This indicates that using IPM to solve subproblems in ENERGY-L is a real must.

ENERGY-L is particularly demanding with respect to the memory requirements. Each subproblem needs more than 80 MB of storage. As a result, we could not solve the global formulation of the two subproblem example with none of the two LP codes we used. It is worth pointing out that decomposition easily handled the case of two subproblems on a machine with 128 MB of RAM. Since a sequential implementation of decomposition calls for only one

Table 8: Solution of ENERGY-L problems with ACCPM.

| Problem | Exact subgradients | | | | $\varepsilon$-subgradients | | | |
|---|---|---|---|---|---|---|---|---|
| | Out | Inner | IPM | Time | Out | Inner | IPM | Time |
| ENERGY-L1 | 44 | 82 | 898 | 23956 | 32 | 57 | 534 | 16590 |
| ENERGY-L2 | 31 | 69 | 1239 | 37980 | 36 | 71 | 994 | 32894 |

subproblem in the core memory at the time, one can conceivably clean the memory after each subproblem optimization to keep only one currently loaded subproblem. In this way, the only limit on solving a variant with many subproblems would be the CPU time. (Note that loading and unloading of the subproblem is a negligible effort compared with the time of its solution.) The obvious way to deal with the difficulty arising from a multiplication of subproblems is parallel computation. Since we plan to develop a parallel implementation in a very near future, we did not attempt to go beyond the two subproblem variant of ENERGY-L with a current sequential implementation. The results of runs are collected in Table 8. L1 and L2 in the names denote the one- and the two-subproblems variants of the problem, respectively.

### 5.3 Warm start of the primal-dual method

All the computational results presented in Sections 5.1 and 5.2 have been obtained with the warm start technique applied in the solution of subproblems. To give the reader some idea of the performance of the warm start routine, we solved medium- and large-scale problems with this technique disabled. The results of these experiments are collected in Table 9. We report in it the total number of IPM iterations and the average number of IPM iterations per call to a single subproblem for cold and warm starts, respectively. These results show that, on the average, the warm start technique requires about one third of the iterations needed by the cold start.

Table 9: Cold start vs. warm start.

| Problem | Cold Start | | Warm Start ($\varepsilon$-subgr) | |
|---|---|---|---|---|
| | total | average | total | average |
| MARKAL | 2502 | 30 | 1450 | 15 |
| MACHINES | 2047 | 10 | 937 | 4 |
| ENERGY-S | 1241 | 28 | 491 | 10 |
| ENERGY-L1 | 2283 | 52 | 534 | 17 |
| ENERGY-L2 | 3207 | 51 | 994 | 14 |

Some more insight into the performance of the reoptimizations with the infeasible primal-dual method is given in Table 10. We report in it the average number of iterations spent to restore the $\varepsilon_0$-center for an altered objective (AC) and the average number of iterations spent to reach the $\varepsilon$-optimality (OPT). We report these numbers again for two variants of ACCPM: with exact subgradients and with $\varepsilon$-subgradients.

The analysis of the results collected in Table 10 confirm that when the sequence of linear programs with varying objective functions is solved, the reoptimizations with the infeasible primal-dual method can be done quite efficiently.

Table 10: Efficiency of the warm start routine.

| Problem | Exact subgradients | | $\varepsilon$-subgradients | |
|---|---|---|---|---|
| | AC | OPT | AC | OPT |
| MARKAL | 4 | 13 | 4 | 11 |
| MACHINES | 1 | 5 | 1 | 3 |
| ENERGY-S | 3 | 10 | 3 | 7 |
| ENERGY-L1 | 6 | 15 | 6 | 11 |
| ENERGY-L2 | 4 | 16 | 4 | 10 |

# 6 Conclusions

In this paper we discussed several enhancements of the analytic center cutting plane scheme for block angular linear programs. We showed that IPMs are good candidates for the solving of the subproblems. The advantages become all the more evident as the size or the degree of degeneracy of the subproblems increases. In that environment, the warm start technique is a must.

Using IPMs for the subproblem has the further advantage of delivering suboptimal feasible solutions with a guaranteed distance to optimality, a feature that the simplex does not offer. Suboptimal solutions generate $\varepsilon$-subgradients. The ACCPM easily handles $\varepsilon$-subgradients, without significant increase in the number of iterations.

ACCPM does not require any a priori knowledge of the optimal solution. However, should some estimate of the solution be given in advance, our trust region scheme allows ACCPM to restrict the search to a limited localization set and thus reach an optimum faster. If the information turns out to be incorrect, the expansion mechanism of the trust region allows fast recovery.

# References


[1] A. Altman and K. C. Kiwiel. A note on some cutting plane methods for convex feasibility and minimization problems. *Computational Optimization and Applications*, 5:175–180, 1996.

[2] E. Andersen, J. Gondzio, C. Meszaros, and X. Xu. Implementation of interior point methods for large scale linear programming. In T. Terlaky, editor, *Interior Point Methods in Mathematical Programming*, pages 189–252. Kluwer Academic Publishers, 1996.

[3] O. Bahn, A. Haurie, S. Kypreos, and J.-P. Vial. A multinational MARKAL model to study joint implementation of carbon dioxide emission reduction measures. In P. Ghosh

and J. Puri, editors, *Joint Implementation of Climate Change Commitments*. Tata Energy Research Institute, 1994.

[4] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[5] Cplex Optimization Inc., Incline Village, NV 89451-9436. *Using the Cplex Callable Library*, 1995.

[6] G. B. Dantzig and P. Wolfe. The decomposition algorithm for linear programming. *Econometrica*, 29:767–778, 1961.

[7] J. Day and A. Haurie. Private communication. Logilab, University of Geneva, 1997.

[8] O. du Merle, J.-L. Goffin, and J.-P. Vial. On the comparative behavior of Kelley's cutting plane method and the analytic center cutting plane method. Technical report, Logilab, University of Geneva, 102 Bd Carl-Vogt, CH-1211, March 1996.

[9] D. M. Gay. Electronic mail distribution of linear programming test problems. *Mathematical Programming Society COAL Newsletter*, 13:10–12, 1985.

[10] J.-L. Goffin, A. Haurie, and J.-P. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38:284–302, 1992.

[11] J.-L. Goffin, A. Haurie, J.-P. Vial, and D. L. Zhu. Using central prices in the decomposition of linear programs. *European Journal of Operational Research*, 64:393–409, 1993.

[12] J.-L. Goffin, Z. Q. Luo, and Y. Ye. On the complexity of a column generation algorithm for convex and quasiconvex feasibility problems. In W.W. Hager, D.W. Hearn, and P.M. Pardalos, editors, *Large Scale Optimization: State of the Art*, pages 187–196. Kluwer Academic Publishers, 1993.

[13] J.-L. Goffin, Z. Q. Luo, and Y. Ye. Complexity analysis of an interior point cutting plane method for convex feasibility problems. *SIAM Journal on Optimization*, 6:638–652, 1996.

[14] J.-L. Goffin and J.-P. Vial. Shallow, deep and very deep cuts in the analytic center cutting plane method. Technical report, Logilab, University of Geneva, 102 Bd Carl-Vogt, CH-1211, May 1996. To appear in *Mathematical Programming*.

[15] J. Gondzio. HOPDM (version 2.12) – a fast LP solver based on a primal-dual interior point method. *European Journal of Operational Research*, 85:221–225, 1995.

[16] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1996.

[17] J. Gondzio. Warm start of the primal-dual method applied in the cutting plane scheme. Technical report, Logilab, University of Geneva, 102 Bd Carl-Vogt, CH-1211, May 1996, revised in April 1997.

[18] J. Gondzio, O. du Merle, R. Sarkissian, and J.-P. Vial. ACCPM - a library for convex optimization based on an analytic center cutting plane method. *European Journal of Operational Research*, 94:206–211, 1996.

[19] J. Gondzio and R. Sarkissian. Column generation with the primal-dual method. Technical report, Logilab, University of Geneva, 102 Bd Carl-Vogt, CH-1211, June 1996.

[20] J. Gondzio and T. Terlaky. A computational view of interior point methods for large scale linear programming. In J. Beasley, editor, *Advances in Linear and Integer Programming*, pages 103–144. Oxford University Press, 1996.

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

[22] K.C. Kiwiel. Efficiency of the analytic center cutting plane method for convex minimization. *SIAM Journal on Optimization*, 7:336–346, 1997.

[23] I. J. Lustig, R.E. Marsten, and D.F. Shanno. Interior point methods for linear programming: computational state of the art. *ORSA Journal on Computing*, 6:1–14, 1994.

[24] N. Nakicenovic, A. Gruebler, A. Inaba, S. Messner, S. Nilsson, Y. Nishimura, H-H. Rogner, A. Schaefer, L. Schrattenholzer, M. Strubegger, J. Swisher, D. Victor, and D. Wilson. Long-term strategies for mitigating global warming. *Energy—The International Journal*, 18(5):409–601, 1993.

[25] Y. Nesterov. Complexity estimates of some cutting plane methods based on the analytic center. *Mathematical Programming*, 69:149–176, 1995.

[26] J. Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 40:59–93, 1988.

[27] J.-P. Vial. A generic path-following algorithm with a sliding constraint and its application to linear programming and the computation of analytic centers. Technical report, Logilab, University of Geneva, 102 Bd Carl-Vogt, CH-1211, February 1996.

[28] S. Wright. *Primal-Dual Interior Point Methods*. SIAM, Philadelphia, 1997.