Università di Pisa
Dipartimento di Informatica

Technical Report: TR-04-14

About Lagrangian Methods in Integer Optimization

Antonio Frangioni

30th June 2004

ADDRESS: via F. Buonarroti 2, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

About Lagrangian Methods in Integer Optimization

Antonio Frangioni *

30th June 2004

Abstract

It is well-known that the Lagrangian dual of an Integer Linear Program (ILP) provides the same bound as a continuous relaxation involving the convex hull of all the optimal solutions of the Lagrangian relaxation. It is less often realized that this equivalence is *effective*, in that basically all known algorithms for solving the Lagrangian dual either naturally compute an (approximate) optimal solution of the "convexified relaxation", or can be modified to do so. After recalling these results we elaborate on the importance of the availability of primal information produced by the Lagrangian dual within both exact and approximate approaches to the original (ILP), using three optimization problems with different structure to illustrate some of the main points.

Keywords: Lagrangian dual, Integer Linear Programs.

Introduction

Most Integer Linear Programs (ILP) exhibit a structure that can be exploited in order to construct efficient solution approaches. One of the most general and common forms of structure is

$$(P) \qquad \min_x \{ \, cx : Ax = b, Ex = d, x \in \mathbb{N}^n \, \}$$

where the constraints $Ex = d$ are "easy", i.e., (P) would be an "easy" problem if the *complicating* constraints $Ax = b$ could be removed. In other words, the constraints $Ax = b$ break down the structure of the constraints $Ex = d$, and such a structure could be exploited to efficiently solve the problem if the complicating constraints were not present; one typical example is that of *separable* constraints $Ex = d$, i.e., the one where (P) would decompose into a number of smaller independent subproblems if the *linking* constraints $Ax = b$ could be removed. We remark that the same kind of structure is often found in problems where some (or even all) of the variables are not constrained to be integer-valued; since for most purposes the two cases are identical except for the heavier notation required by the Mixed-Integer one, in the following we will mainly work with the pure integer case, discussing the extension of the results to the other one only when it is not straightforward.

As a first example (others will be presented later), let us consider the Unsplittable Multicommodity Min-Cost Flow problem (UMMCF). A directed graph $G = (N, A)$ is given, e.g. representing a communication network; each arc $(i, j) \in A$ has an associated routing cost $c_{ij} \geq 0$ and upper capacity $u_{ij} > 0$. A set K of origin-destination pairs (o_h, d_h), $h \in K$, is defined; for each o-d pair h, an amount δ_h is given which represents the "demand of communication" between o_h and d_h. The problem is to select a set of $|K|$ paths in G of minimal total cost such that all the communication between each origin and the corresponding destination can be routed on the corresponding

*Università di Pisa, Dipartimento di Informatica, Largo B. Pontecorvo 1, 56127 Pisa (ITALY), e-mail: frangio@di.unipi.it

path without violating the arc capacity constrains. Using flow variables $x^h = [x^h_{ij}]_{(i,j)\in A}$ for each *commodity* $h \in K$, the problem can be formulated as

$$\begin{array}{rcll} \min \sum_{h\in K} \delta_h \sum_{(i,j)\in A} c_{ij} x^h_{ij} & & & \\ \sum_{(j,i)\in A} x^h_{ji} - \sum_{(i,j)\in A} x^h_{ij} & = & b^h_i & i \in N, h \in K \qquad (1) \\ \sum_{h\in K} \delta_h x^h_{ij} + s_{ij} & = & u_{ij} & (i,j) \in A \qquad (2) \\ x^h_{ij} \in \{0,1\}, s_{ij} \geq 0 & & & (i,j) \in A, h \in K \end{array}$$

where b^h_i is -1 if $i = o_h$, 1 if $i = d_h$ and 0 otherwise, and the s_{ij} are slack variables.

The arc capacity constraints (2) are "complicating" in that, if they were removed, the problem would decompose into $|K|$ independent "easy" Shortest Path problems, one for each o-d pair. Thus, the complicating constraints both prevent decomposition and destroy the special structure of the "easy" constraints (total unimodularity of the flow conservation constraints). Due to the presence of (2), (UMMCF) is both large-scale and $\mathcal{NP}$-hard.

A well-known way to exploit this structure is to form the *Lagrangian relaxation* of (P) with respect to the complicating constraints $Ax = b$, i.e., the problem

$$(P_y) \qquad \min_x \{\ cx + y(b - Ax) : Ex = d, x \in \mathbb{N}^n\ \}$$

for a fixed vector y of *Lagrangian multipliers*. While the complicating constraints are relaxed in (P_y), a penalty term is added to the objective function to discourage their violation. Due to the structure of (P), (P_y) is "easy", i.e., significantly easier than (P) to solve. In the (UMMCF) case, for instance, there is a Lagrangian multiplier y_{ij} for each arc capacity constraint, and therefore one for each arc; the Lagrangian relaxation decomposes into $|K|$ Shortest Path problems with modified costs $c_{ij} + y_{ij}$, one for each o-d pair, plus the $|A|$ problems $\min_{s_{ij}}\{-y_{ij}s_{ij} : s_{ij} \geq 0\}$, that have finite optimal solution $s_{ij} = 0$ if and only if $y_{ij} \leq 0$ for all $(i,j) \in A$.

In the following, m will denote the number of complicating constraints $Ax = b$, and therefore of Lagrangian multipliers, and $v(\cdot)$ the (possibly infinite) optimal value of the objective function of an optimization problem. It is easy to check that, for each choice of $y \in \mathbb{R}^m$, (P_y) is a relaxation of, i.e., $v(P_y) \leq v(P)$. Since (P_y) can be easily solved, it can be computationally feasible to find the best possible Lagrangian relaxation, i.e., to solve the *Lagrangian dual*

$$(D) \qquad \max_y \{\ v(P_y)\ :\ y \in \mathbb{R}^m\ \}$$

of (P) with respect to the complicating constraints $Ax = b$. Since $v(P_y) \leq v(P)\ \ \forall y \in \mathbb{R}^m$, (D) is still a relaxation of (P), that is, $v(D) \leq v(P)$.

This technique has been used in the past with remarkable success in several applications to derive tight lower bounds and/or to help constructing good feasible solutions for difficult optimization problems, especially large-scale ones like multicommodity Network Design problems, Unit Commitment problems, stochastic integer problems, and others. The literature on the subject is immense and cannot be reviewed here; we just refer to a few representative papers Held and Carp (1971), Geoffrion (1974), Gavish (1985), Guignard and Kim (1987), Zhuang and Galiana (1988), Guignard and Rosenwein (1989), Balakrishnan, Magnanti and Wong (1989), Barnhart and Sheffi (1993), Farwolden and Powell (1994), Balakrishnan, Magnanti and Wong (1995), Chang and Gavish (1995), Carraresi, Girardi and Nonato (1995), Gouveia (1995), Takriti and Birge (2000), Bacaud, Lemaréchal, Renaud and Sagastizábal (2001), Crainic, Frangioni and Gendron (2001), Dentcheva, Prékopa and Ruszczynski (2002), Borghetti, Frangioni, Lacalandra and Nucci (2003), Cappanera, Gallo and Maffioli (2003), Schultz (2003), and the references therein.

Even with the best possible choice y^* of the Lagrangian multipliers, there is no guarantee that the penalty term in the objective function will lead to a feasible solution, i.e., that the optimal solution of (P_{y^*}) satisfies the relaxed constraints $Ax = b$. In fact, this is most often not the case: whenever this happens, the optimal solution of (P_{y^*}) is also optimal for (P). Thus, when solving (D) by some standard *subgradient method* (Held, Wolfe and Crowder (1974), Camerini, Fratta and Maffioli (1975), Poljak (1977)) at the end of the optimization process only an (approximately) optimal dual solution y^*, and the corresponding lower bound, are available, together with the unfeasible primal solution $\bar{x}$ of the last Lagrangian relaxation solved. The subgradient algorithm has been for a long time the only computationally viable approach for solving (D), up to the point that "Lagrangian approach" has been often used as a shorthand for "Lagrangian dual solved by means of a subgradient algorithm", and still attracts interest today (Larsson, Patriksson and Strömberg (1999), Barahona and Anbil (2000)) for its simplicity and relatively good performances. Thus, a number of ways have been proposed to exploit the unfeasible primal information provided by the Lagrangian relaxation within exact and approximate approaches for (P) (Geoffrion (1974), Guignard and Kim (1987), Guignard (1998)).

It is also well-known (Geoffrion (1974)) that the Lagrangian dual (D) is equivalent to the following *convexified relaxation* of (P)

$$(\tilde{P}) \qquad \min_x \{ \, cx : Ax = b, x \in conv(X) \, \}$$

where $X = \{ \, x \in \mathbb{N}^n : Ex = d \, \}$ is the feasible set of the Lagrangian Relaxation (P_y) of (P) with respect to constraints $Ax = b$, and $conv(\cdot)$ denotes the convex hull. In other vords, the Lagrangian dual is equivalent, from a primal viewpoint, to a *partial convexification* operation that is only applied to the "easy" constraints $Ex = b$. This result has been primarily used (in some ways to be recalled below) to compare the relative strenght of the bounds obtained by different Lagrangian relaxations and the one obtained by the continuous relaxation.

Despite efforts for recalling it (Lemaréchal (2001)), it is less well-known that the equivalence between (D) and $(\tilde{P})$ is *effective*, in that basically all known algorithms for solving (D) either naturally compute an (approximate) optimal solution of $(\tilde{P})$, or can be modified to do so. The goal of this paper is to remind this fact and point out its implications upon the possible use of Lagrangian approaches within algorithms for solving (P), i.e., that a Lagrangian dual provides as much as precious information as a continuous relaxation and possibly more, and therefore that it can — at least, in theory — be used in every application where a continuous relaxation is used. We will stick to applications of the Lagrangian approach to "essentially discrete" cases, for which the only required theoretical tool is the familiar linear duality; for a more detailed and through treatment of the technical issues in Lagrangian duality the interested reader is warmly referred to the excellent Lemaréchal (2001). Also, we will not even attempt at describing the numerous and intricate relationships between choosing the right model and choosing the right Lagrangian approach for different classes of problems, since these aspects are fully and clearly covered in the recent Guignard (2003).

The structure of the paper is the following: in Section 1 the main theoretical results about the Lagrangian dual (D) and the convexified relaxation $(\tilde{P})$ are reviewed. Section 2 is devoted to remind how an (approximate) optimal solution of $(\tilde{P})$ is automatically obtained when solving (D) by means of one of the very many variants of the Cutting Plane/Dantzig Wolfe approach or by some recent forms of subgradient algorithms; moreover some relevant algorithmic details and extensions are discussed. Then, in Section 3 and in Section 4 the usefulness of the primal information within approximate and exact approaches, respectively, for the solution of (P) is discussed, using three optimization problems with very different structure to illustrate some of the main issues. Finally, in Section 5 some conclusions are drawn.

1 The equivalence result

It is well-known that, for any (ILP), the feasible set can be replaced with its convex hull; as a consequence, one has

$$v(P_y) = \min_x \{ \, cx + y(b - Ax) : x \in conv(X) \, \}$$

for each $y \in \mathbb{R}^m$. It is also well-known that, if all entries of E and d are rational, then $conv(X)$ is a convex polyhedron, i.e., there exists a finite set of linear constraints such that

$$conv(X) = \{x : \tilde{A}x \geq \tilde{b}\} \ .$$

We can therefore show the announced result:

Teorem 1.1 *$v(D) = v(\tilde{P})$.*

Proof: We have

$$v(\tilde{P}) = \min_x\{ \ cx : Ax = b, \tilde{A}x \geq \tilde{b} \ \} \ .$$

Then, by linear duality

$$v(\tilde{P}) = \max_{y,w}\{ \ yb + w\tilde{b} : yA + w\tilde{A} = c, w \geq 0 \ \}$$

which can be rewritten as

$$\max_y\{ \ yb + \max_w\{ \ w\tilde{b} : w\tilde{A} = c - yA, w \geq 0 \ \} \ \} \ .$$

Then, applying again linear duality to the inner problem, one has

$$v(\tilde{P}) = \max_y\{ \ yb + \min_x\{ \ (c - yA)x : \tilde{A}x \geq \tilde{b} \ \} \ \} = v(D).$$

□

A straightforward corollary of the above Theorem is that the Lagrangian dual of a Linear Program (LP) is equivalent to the classical linear dual; X is a convex set, and therefore it is equal to its convex hull. For instance, for the continuous relaxation of (P)

$$(\bar{P}) \qquad \min_x\{ \ cx : Ax = b, Ex = d, x \geq 0 \ \} \ ,$$

the Lagrangian dual with respect to $Ax = b$ is

$$(\bar{D}) \qquad \max_y\{ \ v(\bar{P}_y) = \min_x\{ \ cx + y(b - Ax) : Ex = d, x \geq 0 \ \} \ \}$$

$$= \max_{y,w}\{ \ yb + wd : wE + yA \leq c \ \} \ ;$$

i.e., the linear dual of $(\bar{P})$. In particular, the Lagrangian multipliers are the dual variables relative to the relaxed constraints $Ax = b$, and the Lagrangian dual can be considered a *partial dual* of $(\bar{P})$. Indeed, the Lagrangian dual follows the same rules as the linear dual regarding the sign of the multipliers; for instance, if the complicating constraints have the form $Ax \leq b$, then the Lagrangian multipliers are constrained to be nonpositive ($y \leq 0$), while if the constraints have the form $Ax \geq b$ then the Lagrangian multipliers are constrained to be nonnegative ($y \geq 0$).

Theorem 1.1 is typically used to prove the following facts:

- Since the feasible region of $(\tilde{P})$ is contained in that of $(\bar{P})$, and the two problems have the same objective function, $(\tilde{P})$ is a better (not worse) relaxation of (P) than $(\bar{P})$, i.e., $v(D) \geq v(\bar{P})$.
- However, if the "easy" constraints $Ex = d$ have the *integrality property*, i.e., $v(P_y) = v(\bar{P}_y)$ for each $y \in \mathbb{R}^m$, then $v(D) = v(\tilde{P}) = v(\bar{P})$. The integrality property implies that all vertices of the feasible region of $(\bar{P}_y)$ are integral, i.e., that (P_y) is as easy as a (LP); this is a "no free lunch" principle, in that, in order for the Lagrangian dual to provide a better bound than the continuous relaxation, the Lagrangian relaxation must be "more difficult" than a (LP), i.e., the $Ex = d$ constraints must not already provide an "exact" description of $conv(X)$.

- The above observations may allow to estimate the quality of the bound provided by different Lagrangian duals of the same problem. For instance, consider the Lagrangian dual of (P) with respect to $Ex = d$

$$(D') \qquad \max_w \{ \min_x \{ cx + y(d - Ex) : x \in X' \} \} .$$

  where $X' = \{ x \in \mathbb{N}^n : Ax = b \}$. If both blocks of constraints $Ax = b$ and $Ex = d$ have the integrality property, then $v(D') = v(D) = v(\bar{P})$. If, instead, $Ex = d$ has the integrality property but $Ax = b$ has not, then (D') may provide a strictly better bound than (D), i.e., $v(D') \geq v(D) = v(\bar{P})$ and the inequality can be strict. This happens for instance for (UMMCF): the Lagrangian dual with respect to the flow conservation constraints (1) decomposes into $|A|$ independent Knapsack problems. These subproblems are easier to solve than (UMMCF) but they do not have the integrality property, and therefore the corresponding Lagrangian bound can be strictly better than the one provided by the continuous relaxation, or, equivalently, by the Lagrangian dual with respect to the capacity constraints (2), whose subproblems have the integrality property.

- A different way of exploiting the structure of (P) via a Lagrangian relaxation is the *Lagrangian Decomposition* (Guignard and Kim (1987)), which amounts at rewriting (P) in its equivalent form

$$(P) \qquad \min_{x,x'} \begin{cases} c(x + x')/2 \\ Ax' = b, \;\; x' \in \mathbb{N}^n \\ Ex = d, \;\; x \in \mathbb{N}^n \\ x = x' \end{cases}$$

  and solving the corresponding Lagrangian dual with respect to the constraints $x = x'$, i.e.,

$$(D'') \qquad \max_w \begin{cases} \min_x \{ (c/2 - w)x : x \in X \} + \\ \min_{x'} \{ (c/2 + w)x' : x' \in X' \} \end{cases} .$$

  From Teorem 1.1 it is easy to prove that

$$v(D'') = \min_x \{ cx : x \in \; conv(X) \; \cap \; conv(X') \} ,$$

  and therefore $v(D'') \geq \max\{v(D'), v(D)\} \geq v(\bar{P})$. Hence, the Lagrangian Decomposition provides a bound on $v(P)$ that is not worse than the best among the ones provided by the Lagrangian duals with respect to $Ax = b$ and $Ex = d$ individually, at the cost of a Lagrangian relaxation that requires the solution of both the individual Lagrangian subproblems. In particular, it is easy to see that $v(D'') = \max\{v(D'), v(D)\}$ if at least one of the two blocks of constraints has the integrality property, that $v(D'') = v(\bar{P})$ if both blocks have it, while $v(D'')$ can be strictly better than $\max\{v(D'), v(D)\}$ if none of the two blocks of constraints has the integrality property.

All the above observations derive from the fact that the optimal objective function values of (D) and $(\tilde{P})$ are equal. However, the proof of Theorem 1.1 actually shows something more:

- (D) and $(\tilde{P})$ are *the linear dual* of each other;
- in order to *prove* the optimality of an optimal solution y^* to (D), any algorithm for (D) must also construct and *optimal solution* $\tilde{x}$ of $(\tilde{P})$ (as an "optimality certificate").

In the next section we remind that one of the simplest algorithms that can be used to solve (D) *proving* optimality of the obtained solution, Kelley's Cutting Plane method (Kelley (1960)), is the "dual interpretation" of the well-known Dantzig-Wolfe decomposition method (Dantzig and Wolfe (1960)) for structured (LP)s, and therefore it provides an optimal solution of $(\tilde{P})$ as well.

2 Cutting Planes, Dantzig-Wolfe decomposition and columns generation

To simplify the treatment we temporarily assume that X is a *compact* set; later on in this section we show how this assumption can be dropped at the only cost of slightly complicating the notation. We also assume that X is nonempty; if this is not the case, (P) has no feasible solution, and we can assume that this is "easily" found out the first time (P_y) is solved (whatever the vector y). Therefore, the *Lagrangian function*

$$\phi(y) = \min_x\{\ cx + y(b - Ax) : x \in X\ \}$$

is finite everywhere; also, ϕ is *proper*, i.e., $\phi(y) < +\infty \quad \forall y$. Being the pointwise minimum of a set of linear functions, ϕ is concave; in particular, since X is a discrete set ϕ is *polyhedral*, i.e., its *epigraph*

$$Epi(\phi) = \{\ (v, y) : v \leq \phi(y)\ \}$$

is a polyhedron. In other words, each solution $\bar{x}$ to (P_y) for a given y is associated with the *support hyperplane* $b - A\bar{x}$ to $Epi(\phi)$, known as a *subgradient* of ϕ in y[1]. Note that this is true even if (P) is a Mixed-Integer Linear problem, since in this case too (P_y) has only a finite number of possible extremal solutions; for each non-extremal solution x the corresponding constraint $v \leq cx + y(b - Ax)$ in the definition of $Epi(\phi)$ can be obtained as a convex combination of the constraints corresponding to some extremal (optimal) solutions, and therefore it is redundant. Clearly, ϕ is not differentiable everywhere, and therefore the algorithms for smooth optimization cannot be used for solving (D).

2.1 The Cutting Plane algorithm

It is straightforward to see that (D) can be rewritten as a Linear Program:

$$(D) \qquad \max_{v,y}\{\ v : v \leq cx + y(b - Ax)\ ,\ x \in X\ \} \tag{3}$$

(this is not surprising since, as shown above, (D) is the dual of $(\tilde{P})$). This does not mean that solving (D) is an easy task: (3) has, in principle, as many constraint as there are solutions of (P_y), and this number can be huge. However, not all constraints in (3) are actually necessary; indeed, the $m + 1$ constraints corresponding to an optimal base would suffice. This immediately suggests a *constraints generation* approach, where at each iteration the (Dual) *Master Problem*

$$(D_{\mathcal{B}}) \qquad \max_{v,y}\{\ v : v \leq cx + y(b - Ax)\ ,\ x \in \mathcal{B}\ \} \tag{4}$$

is solved where $\mathcal{B}$ is a "small" subset of X. This corresponds to solving

$$\max_y\{\ \phi_{\mathcal{B}}(y) = \min_x\{\ cx + y(b - Ax) : x \in \mathcal{B}\ \}\ \}\ ,$$

i.e., to minimizing the concave polyhedral upper approximation $\phi_{\mathcal{B}}$ of ϕ; $\phi_{\mathcal{B}}$ is known as the *cutting plane model* of ϕ. It is easy to check whether or not the optimal solution $(\tilde{v}, \tilde{y})$ of $(D_{\mathcal{B}})$ is also optimal for (D) by solving the *separation problem* $(P_{\tilde{y}})$, i.e., computing $\phi(\tilde{y})$: if $\tilde{v} > \phi(\tilde{y})$ then any optimal solution $\bar{x}$ of $(P_{\tilde{y}})$ provides a violated constraint of (3) that can be added to $\mathcal{B}$, otherwise $\tilde{v} = \phi(\tilde{y}) = v(D)$.

Kelley's Cutting Plane algorithm applied to the solution of (D), shown in Figure 1, determines at each step a lower and an upper bound on $v(D)$, in that $\tilde{v} \geq v(D) \geq \phi(\tilde{y})$, and terminates in a finite number of steps (eventually, $\mathcal{B} = X$) when the two coincide. In order for the algorithm to be well-defined, the initial set $\mathcal{B}$ must be "sufficiently large" to ensure that $(D_{\mathcal{B}})$ has a finite optimal solution; this can always be obtained e.g. by setting $\mathcal{B} = \{\hat{x}\}$ where $\hat{x}$ is a feasible solution for (P), which corresponds to inserting the constraint $v \leq c\hat{x}$ into $(D_{\mathcal{B}})$.

[1] Actually, the terms "subgradient" and "epigraph" are suited to a convex function, while for the concave ϕ "supergradient" and "ipograph" should be used.

$\langle$ initialize $\mathcal{B}$ $\rangle$;
do
  $(\tilde{v}, \tilde{y}) = \text{argmax}_{v,y}\{\ v : v \leq cx + y(b - Ax)\ ,\ x \in \mathcal{B}\ \}$;   /* $(D_{\mathcal{B}})$ */
  $\bar{x} = \text{argmin}_x\{\ (c - \tilde{y}A)x : x \in X\ \}$;   /* $(P_{\tilde{y}})$ */
  $\phi(\tilde{y}) = c\bar{x} + \tilde{y}(b - A\bar{x})$; $\mathcal{B} = \mathcal{B} \cup \{\bar{x}\}$;
while($\tilde{v} > \phi(\tilde{y})$);

Figure 1: The Cutting Plane Algorithm

It may be worth noting in passing that Kelley's algorithm works for any convex function, and therefore it has numerous applications outside Lagrangian relaxation. Also, this is by no means the only important algorithmic paradigm using the cutting plane idea; apart from polyhedral approaches in integer programming, either "pure" or "Branch & Cut" ones, the "dual" of the Dantzig-Wolfe decomposition method described next, *Bender's Decompositon* (Benders (1962)), is also a cutting plane algorithm.

2.2 The Dantzig-Wolfe decomposition method

From a "primal viewpoint", the Cutting Plane algorithm is equivalent to the well-known *Dantzig-Wolfe decomposition method.* In fact, the linear dual of $(D_{\mathcal{B}})$ is the *Primal Master Problem*

$$(P_{\mathcal{B}}) \qquad \min_\theta \left\{\ c\left(\textstyle\sum_{x\in\mathcal{B}} x\theta_x\right) : A\left(\textstyle\sum_{x\in\mathcal{B}} x\theta_x\right) = b, \theta \in \Theta\ \right\} \tag{5}$$

where $\Theta = \{\ \theta \geq 0 : \sum_{x\in\mathcal{B}} \theta_x = 1\ \}$ is the unitary simplex of proper dimension. $(P_{\mathcal{B}})$ has a variable for each row of $(D_{\mathcal{B}})$, i.e., for each element of $\mathcal{B}$. The "explicit form" (5) of $(P_{\mathcal{B}})$ is equivalent to the following "implicit form":

$$(P_{\mathcal{B}}) \qquad \min_x\{\ cx : Ax = b, x \in X_{\mathcal{B}} = conv(\mathcal{B})\ \}\ . \tag{6}$$

The latter definition immediately shows the relationship between the Cutting Plane algorithm and $(\tilde{P})$: (5) with $\mathcal{B} = X$ is just a formulation of $(\tilde{P})$ where the constraint $x \in conv(X)$ is explicitly written in terms of the convex multipliers θ. In particular, (5) with $\mathcal{B} = X$ is the linear dual of (3), as anticipated by Theorem 1.1; here $conv(X)$ is expressed in terms of its extreme points, rather than in terms of its faces.

From the primal viewpoint, the cutting plane algorithm is the same as a column generation approach for solving the "explicit form" of $(\tilde{P})$, which is a (LP) with "many" columns. At each step, the restriction of $(\tilde{P})$ to the subset of columns $\mathcal{B}$ is solved, yielding a primal optimal solution θ^*, and therefore a feasible solution $\tilde{x} = \sum_{x\in\mathcal{B}} x\theta^*_x$ for $(\tilde{P})$, and a dual solution $\tilde{y}$ corresponding to constraints $Ax = b$. The *pricing problem* $(P_{\tilde{y}})$ determines the column $\bar{x} \in X$ of least *reduced cost* $(c - \tilde{y}A)x$: if $(c - \tilde{y}A)\bar{x} < 0$ then $\bar{x}$ can be added to $\mathcal{B}$ in the hope to construct a better primal solution at the next iteration, otherwise $\tilde{x} = x^*$ is optimal for $(\tilde{P})$. These steps are exactly those of the algorithm shown in Figure 1, except that there $\tilde{x}$ is not explicitly computed. Hence, upon termination, the Cutting Plane algorithm constructs an optimal solution x^* for $(\tilde{P})$.

The above analisys shows that the Cutting Plane algorithm used to solve (D) solves also $(\tilde{P})$, and suggests the following considerations:

- From the primal viewpoint, the Master Problem uses the *inner approximation* $X_{\mathcal{B}}$ of $conv(X)$, that is iteratively enlarged with the points $\bar{x}$ generated by the pricing problem $(P_{\tilde{y}})$ until it contains one optimal solution of $(\tilde{P})$; from the dual viewpoint, the Master Problem uses the *outer approximation* $\phi_{\mathcal{B}}$ of ϕ, that is iteratively refined with the subgradients $b - A\bar{x}$ generated by the separation problem $(P_{\tilde{y}})$ until it is "exact" at one optimal solution of (D).

- The structure of (P) is used to generate (extreme) points of $conv(X)$; thus, this approach is particularly attractive for all situations where obtaining an explicit representation of $conv(X)$ is more difficult than actually optimizing over X. In fact, the Lagrangian dual is especially interesting when the $Ex = d$ constraints do *not* possess the integrality property, but a "reasonably efficient" way for solving the Lagrangian relaxation is available.

- The Cutting Plane algorithm for solving (D) is *exactly* the standard column generation approach applied to the "explicit form" of $(\tilde{P})$ as a (LP) with many columns; hence, there is a deep link between Lagrangian techniques and column generation approaches, that is discussed more in details below.

2.3 Extension: noncompact X

The Cutting Plane algorithm can be extended to the case where X is noncompact, provided that the algorithm that solves the Lagrangian relaxation is capable of providing a *unbounded ascent direction* ν for $conv(X)$ whenever $\phi(y) = -\infty$. Since for any y such that $(c - yA)\nu < 0$ the Lagrangian relaxation (P_y) is unbounded, each such ν is associated with a linear constraint $(c - yA)\nu \geq 0$ that is valid for the *effective domain* Y of $\phi(y)$ (the set of all points where it is finite-valued). In other words, the the extreme rays of the *recession cone* C of $conv(X)$ carachterize the effective domain of ϕ.

The Cutting Plane algorithm can be extended to the noncompact case by replacing $\mathcal{B}$ by $\mathcal{B}^0 \cup \mathcal{B}^1$ where $\mathcal{B}^0 \subseteq C$ and $\mathcal{B}^1 \subseteq X$. The Primal and Dual Master Problem become respectively

$$(P_{\mathcal{B}}) \qquad \min_\theta \left\{ \begin{array}{l} c\left(\sum_{x \in \mathcal{B}^1} x\theta_x + \sum_{\nu \in \mathcal{B}^0} \nu\theta_\nu\right) \\ A\left(\sum_{x \in \mathcal{B}^1} x\theta_x + \sum_{\nu \in \mathcal{B}^0} \nu\theta_\nu\right) = b \\ \sum_{x \in \mathcal{B}^1} \theta_x = 1 \\ \theta \geq 0 \end{array} \right. , \tag{7}$$

$$(D_{\mathcal{B}}) \qquad \max_{v,y} \left\{ \begin{array}{ll} yb + v & \\ v \leq (c - yA)x & x \in \mathcal{B}^1 \\ 0 \leq (c - yA)\nu & \nu \in \mathcal{B}^0 \end{array} \right.$$

or, equivalently,

$$(P_{\mathcal{B}}) \qquad \min_x\{\ cx : Ax = b, x \in conv(\mathcal{B}^1) + cone(\mathcal{B}^0)\ \}\ ,$$

$$(D_{\mathcal{B}}) \qquad \max_y\{\ \phi_{\mathcal{B}^1}(y) : y \in Y_{\mathcal{B}}\ \}\ ,$$

where $Y_{\mathcal{B}} = \{\ y : (c - yA)\nu \geq 0, \nu \in \mathcal{B}^0\ \}$ is an outer approximation of the effective domain of ϕ. At each iteration of the algorithm, the Lagrangian relaxation computes either an optimal solution $\bar{x} \in X$, that is added to $\mathcal{B}^1$, or a feasible unbounded direction $\bar{\nu} \in C$, that is added to $\mathcal{B}^0$. It should be noted that in the latter case $\phi(\tilde{y}) = -\infty$, and therefore no new lower bound on $v(D)$ is found.

It is interesting to remark that $conv(X)$ may be a cone; in this case, *no* iteration, except the last one, produces a lower bound on $v(D)$, i.e., at *all* iterations, except the last one, $\phi(\tilde{y}) = -\infty$. The corresponding "explicit" form (7) is a generic problem amenable of solution by a column generation approach, where $\mathcal{B}_1 = \{0\}$, i.e., there is no convexity constraint. Note that, without loss of generality, one may always assume $conv(X)$ to be a cone by simply regarding the convexity constraint as a generic one, i.e., considering the Lagrangian multipliers space to be $\mathbb{R}^{m+1}$, and v to be just like any other Lagrangian multiplier. However, exploiting the presence of a convexity constraint in the formulation allows one to derive lower bounds on $v(D)$ at all iterations, which is useful to early terminate the algorithm (see e.g. Carraresi, Girardi and Nonato (1995), Ben Amor (2002)).

2.4 Extension: decomposable X

The Cutting Plane algorithm can also be specialized to the *decomposable* case where the Lagrangian relaxation decomposes into k independent problems, as e.g. for (UMMCF). This means that $X = X_1 \times X_2 \times \ldots \times X_k$, i.e., X is the Cartesian product of k sets (which we temporarily assume to be compact); any optimal solution $\bar{x}$ of the Lagrangian relaxation has the form $[\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^k]$, where $\bar{x}^h$ is an optimal solution of the h-th subproblem. In other words, the Lagrangian function

$$\phi(y) = yb + \textstyle\sum_{h \in K} \phi^h(y)$$

($K = \{1, 2, \ldots, k\}$) is the sum of its k component functions

$$\phi^h(y) = \min_{x^h} \{ \, (c^h - yA^h)x^h : x^h \in X^h \, \} \; ,$$

and $-A^h \bar{x}^h$ is a subgradient of ϕ^h.

At each iteration the *disaggregated Primal Master Problem*

$$(P_{\mathcal{B}}) \qquad \min_\theta \left\{ \begin{array}{l} \sum_{h \in K} c^h \sum_{x \in \mathcal{B}^h} x\theta_x \\ \sum_{h \in K} A^h \sum_{x \in \mathcal{B}^h} x\theta_x = b \\ \sum_{x \in \mathcal{B}^h} \theta_x = 1 \qquad h \in K, \qquad \theta \geq 0 \end{array} \right. \tag{8}$$

is solved instead of (5). Each component x^h of each of the solutions generated so far in the process is kept in the set $\mathcal{B}^h$ and has an associated convex multiplier θ_{x^h}, independent from the multiplier associated to other components belonging to the same solution. The corresponding *disaggreagted Dual Master Problem* is

$$(D_{\mathcal{B}}) \qquad \max_{v,y} \left\{ \; yb + \textstyle\sum_{h \in K} v^h \; : \; v^h \leq (c^h - yA^h)x^h \quad x^h \in \mathcal{B}^h \quad h \in K \; \right\} \; .$$

The two problems can be equivalently rewritten as

$$(P_{\mathcal{B}}) \qquad \min_x \left\{ \; \textstyle\sum_{h \in K} c^h x^h \; : \; \sum_{h \in K} A^h x^h = b \;\; , \;\; x^h \in conv(\mathcal{B}^h) \;\; h \in K \; \right\} \tag{9}$$

$$(D_{\mathcal{B}}) \qquad \max_y \left\{ \; \textstyle\sum_{h \in K} \phi_{\mathcal{B}^h}(y) \; \right\}$$

where $\phi_{\mathcal{B}^h}$ is the cutting plane model of the h-th component ϕ^h of ϕ, while $conv(\mathcal{B}^h)$ is an inner approximation of $conv(X^h)$. From the column generation viewpoint, this corresponds to the case where there is not a unique convexity constraint covering all columns, but the columns can be partitioned into k disjoint subsets, each one having its separate convexity constraint; v^h are the dual variables of the convexity constraints, and their sum (plus the linear term yb, corresponding to a linear "0-th component" of ϕ) gives a lower bound on $v(\tilde{P})$.

It is easy to see that, for the same set of Lagrangian solutions $\mathcal{B} \subseteq X$, the feasible set of (9) strictly contains that of (6); in fact, (5) is the restriction of (8) where all the components $\bar{x}^h$ corresponding to the same solution $\bar{x}$ must have the same convex multiplier. In other words, $conv(\mathcal{B}^1) \times conv(\mathcal{B}^2) \times \ldots \times conv(\mathcal{B}^k)$ is a better approximation of $conv(X)$ than $conv(\mathcal{B})$; equivalently, the sum of the k cutting plane models $\phi_{\mathcal{B}^h}$ is a better approximation of ϕ than the "aggregated" cutting plane model $\phi_{\mathcal{B}}$. The tradeoff is that the disaggregated Master Problems are k times larger than the corresponding aggregated ones, and therefore they are more costly to solve; however, they use the available information about the "true" problems $(\tilde{P})$ / (D) much more efficiently, which often results in a much faster convergence, and ultimately in better overall performances (Jones, Lustig, Farwolden and Powell (1993), Borghetti, Frangioni, Lacalandra and Nucci (2003)). This approach can be easily extended to the case where some of the sets X^h are noncompact in the same way as in the aggregated case.

2.5 Extension: the nonlinear case

The above analysis can be easily extended to the case where (P) is

$$(P) \qquad \min_x \{ \, c(x) : Ax = b, x \in X \, \}$$

where $c()$ is a nonlinear, non necessarily convex, function. Although extending the proof of Theorem 1.1 to this case is not trivial, it is still true that (D) can be written as

$$\max_{v,y}\{\ v : v \leq c(x) + y(b - Ax)\ ,\ x \in X\ \}$$

and therefore its linear dual $(\tilde{P})$ has the form

$$\min_\theta \left\{\ \sum_{x \in X} c(x)\theta_x : A\left(\sum_{x \in X} x\theta_x\right) = b, \theta \in \Theta\ \right\}\ .$$

$(\tilde{P})$ can be stated also in the "explicit form"

$$\min_x\{\ \tilde{c}_X(x) : Ax = b, x \in conv(X)\ \}$$

where

$$\tilde{c}_X(x) = \min_v \{\ v : (v, x) \in conv(\{(c(\bar{x}), \bar{x}) : \bar{x} \in X\})\ \}$$

is the "smallest" convex function "compatible" with $c()$ on X. This shows once again that the Lagrangian Dual is a "convexification tool": the (possibly nonconvex) set X and function $c()$ are replaced with their "convexified versions" $conv(X)$ and $\tilde{c}_X()$, i.e., the better convex approximations that can be built up with the information produced by the Lagrangian relaxation.

The above treatment is still valid for the Mixed-Integer case if $c()$ is nonlinear only on the integer variables, and linear on the continuous ones; in fact, once again the set of extremal solutions of the Lagrangian relaxation is finite. For more general cases, tricky things may happen and more sophisticated tools from convex analisys are required that are out of the scope of the present paper; the interested reader is referred e.g. to Hiriart-Urruty and Lemaréchal (1993), Lemaréchal (2001).

2.6 Other algorithms for (D)

In the previous sections we have focussed on the Cutting Plane/Dantzig Wolfe algorithm because, with the sole help of elementary duality theory, it points out the *algorithmic* relationships between (D) and $(\tilde{P})$. However, basically all the above considerations extend to most known algorithmic approaches to (D).

The study of other approaches for solving (D) has been motivated by some serious drawbacks of the Cutting Plane algorithm, which severely impact on its performances in practice. Without going into details, we only mention that the main problems with the Cutting Plane algorithm are its *instability*, i.e., the fact that the sequence of dual solutions $\{\tilde{y}\}$ generated by the approach has no locality properties, and the need for solving Master Problems whose size grows with the iterations. The instability of the approach is the main cause of its slow convergence rate on many practical problems (the well-known "tailing-off" effect); in turn, this may cause the Master Problem to become exceedingly large and costly. It should be noted, however, that despite these problems the Cutting Plane approach, possibly "hybridized" (Fréville, Guignard and Zhu (1999)), still manages to obtain very interesting performance in some applications (Jones, Lustig, Farwolden and Powell (1993), Carraresi, Girardi and Nonato (1995), Ben Amor (2002)).

The alternative approaches proposed for the solution of (D) can be grouped in two categories:

- cutting-plane-type approaches;
- subgradient-type approached.

The first class of algorithms can be further subdivided into two different (but related) groups: bundle-type algorithms and algorithms based on "centers". The first group squarely aim at reducing the instability of the Cutting Plane algorithm by introducing some "stabilization device" that enhances the locality properties of the sequence of dual solutions, thereby improving the rate of convergence of the approach. A number of different variants of this approach have been proposed: among them, generalized bundle methods (Frangioni (2002)) (that comprise proximal bundle methods (Hiriart-Urruty and Lemaréchal (1993) – Algorithm XV.3.1.4), trust region bundle methods (Hiriart-Urruty and Lemaréchal (1993) – Algorithm XV.2.1.1) and methods based on exponential

(Grigoriadis and Kahchiyan (1995)) or linear-quadratic (Pinar and Zenios (1994)) stabilization terms), proximal level methods (Lemaréchal, Nemirovskii and Nesterov (1995)), ε-descent methods, (Hiriart-Urruty and Lemaréchal (1993) – Algorithm XIV.3.4.2), proximal trajectory methods (Fuduli and Gaudioso (2000)) and Bregman barrier methods (Kiwiel (1999)). These approaches differ in the form (and, therefore, the cost) of the Master Problem they solve and in the properties required for convergence. In most cases they permit to control the size of the set $\mathcal{B}$, allowing some control on the trade-off between the size of the Master Problem and the corresponding convergence speed of the method (Crainic, Frangioni and Gendron (2001), Lemaréchal (2001)); this is simply done by inserting the optimal solution $\tilde{x}$ of the previous Master Problem in $\mathcal{B}$ and then deleting any number of the other elements in $\mathcal{B}$. The second group of approaches focuses instead on providing performance guarantees for the convergence speed by choosing the point $\tilde{y}$ as a "center" of the set of points that is known to contain the optimum (localization set). For instance, approaches based on the analytic center (du Merle, Goffin and Vial (1998)) have better theoretical performances (Nesterov (1995)) than bundle-type methods and good practical performances on "difficult" cases (Goffin, Gondzio, Sarkissian and Vial (1997)); however, the corresponding Master Problem may be more costly, and they are somewhat less flexible as far as the handling of the set $\mathcal{B}$ is concerned.

Subgradient-type methods are simpler in that they do not, on the surface, require the solution of a Master Problem. However, most variants of the algorithm chose the next point $\tilde{y}$ where the Lagrangian function has to be evaluated along the direction $d = g + \theta d_{-}$, where g is the subgradient obtained at the current iteration, d_{-} is the direction used at the previous iteration and $\theta \geq 0$ is chosen according to some rule. It is easy to check that d is therefore chosen using "very aggregated information" about the points $\bar{x}$ obtained at all the previous iterations, i.e., that $d = \alpha(b - A\tilde{x})$ for some $\alpha \geq 0$ and $\tilde{x} \in conv(X)$; hence, even subgradient methods in fact compute convex combinations of points in X. It is therefore not surprising that, with a proper choice of the weighting scheme, $\tilde{x}$ can be made to converge to an optimal solution x^* of $(\tilde{P})$ (Larsson, Patriksson and Strömberg (1999), Barahona and Anbil (2000)). In fact, this kind of subgradient approach is very similar to a bundle-type approach where the set $\mathcal{B}$ is always kept at a maximum size of two (Bahiense, Maculan and Sagastizábal (2002)).

For some specific optimization problems, dual approaches that exploit more closely the structure have been proposed (e.g. Balakrishnan, Magnanti and Wong (1989), Guignard and Rosenwein (1989)); these are usually aimed at obtaining a potentially rough estimate of $z(D)$ at a very low computational effort, and can therefore be useful within enumerative approaches. This kind of methods do not, in general, "automatically" produce primal optimal solutions.

However, it can be safely asserted that whatever method one elects for solving (D) can be easily modified to also produce a solution of (P); at the very least, the obtained estimate of y^* can be used to "warm-start" a cutting-plane-like method. Example of "hybrid" procedures along these lines can be found e.g. in Fréville, Guignard and Zhu (1999), where, for the case of the generalized assignment problem, subgradient optimization and column generation are used in conjunction.

It is out of the scope of this paper to discuss the relative merits of the different algorithms for the solution of (D); the interested reader is referred to the relevant works in the bibliography (e.g. Crainic, Frangioni and Gendron (2001), Lemaréchal (2001), Bahiense, Maculan and Sagastizábal (2002)). What is important for the present discussion is that all these methods compute a set $\mathcal{B}$ which contains points of $conv(X)$, and that this information is used, at each iteration, to construct a primal point $\tilde{x}$ that can be made to converge to an optimal solution x^* of $(\tilde{P})$. Thus, the availability of primal information produced by a Lagrangian approach is the rule, rather than the exception: we can assume that an estimate of an optimal solution of $(\tilde{P})$, and the corresponding set $\mathcal{B}$, is always provided by any "modern" algorithm to solve (D). In the next sections some possible uses of this information and their impact in some real applications are discussed.

3 Lagrangian Heuristics

Lagrangian heuristics (Balakrishnan, Magnanti and Wong (1989), Barnhart and Sheffi (1993), Farwolden and Powell (1994)) exploit the solution process of the Lagrangian dual in order to obtain feasible solutions to (P). Since at each iteration of such a process an optimal solution $\bar{x}$ of the Lagrangian relaxation is available, most Lagrangian heuristics proposed so far attempt at making it feasible, e.g. by means of a greedy procedure, possibly followed by a refinement phase generally based on local search. Although subgradient optimization has mostly been used to drive the solution of the Lagrangian dual, other approaches, such as column generation (Fréville, Guignard and Zhu (1999)) or Bundle methods (Carraresi, Girardi and Nonato (1995), Borghetti, Frangioni, Lacalandra and Nucci (2003)) have also been proposed.

The iterative process used to solve (D) acts thus as a multistart to classical heuristics, which can also possibly exploit the Lagrangian costs $\tilde{c} = c - \tilde{y}A$ to guide the search toward feasible solutions. The heuristic procedures can be run at each iteration, or at specific iterations a priori established, or whenever the solution to the Lagrangian relaxation satisfies some criteria; the Lagrangian heuristics can be invoked either with regularity or more intensely toward the end of the iterative process, when the Lagrangian multipliers are supposed to be of good quality. The feasible solutions provided by the Lagrangian heuristics might even be used as the initial population in a genetic algorithm.

However, other important primal information is obtained as by-product of the iterative process when a "modern" algorithm is used. The algorithms described (or just alluded to) in the previous sections all collect a set $\mathcal{B} \subseteq conv(X)$ and, at each iteration, a multiplier θ_x^* is associated with each $x \in \mathcal{B}$ which produces a point $\tilde{x} \in conv(X)$. Whether $\tilde{x}$ is feasible or not depends on the algorithm chosen; as a general rule, $\tilde{x}$ rapidly becomes "almost feasible", i.e., the violation of constraints $Ax = b$ quickly diminishes when compared with the "average" one relative to the solutions of the Lagrangian relaxations. It is worth noting that the cardinality of $\mathcal{B}$ (very limited in the subgradient algorithms) and the way its elements are stored (aggregated vs. disaggregated) varies from one algorithm to another, but we need not enter into details here. The solution $\tilde{x} \in conv(X)$ and the associated multipliers θ^* provide one alternative source of primal information, about which the following considerations can be made:

- $\tilde{x}$ is a continuous (almost) feasible solution, and therefore all the rounding techniques developed in the Linear Programming context can be used as well in the Lagrangian one. For instance, the components of $\tilde{x}$ corresponding to 0-1 variables, if any, can be interpreted as the probability that the variable is set to 1 in the optimal solution, thus suggesting randomized heuristics (Vazirani (2001)).

- The multipliers θ^* can also be thought of as a "probability distribution" on the elements of the set $\mathcal{B}$, and this information can be used to combine the solutions $x \in \mathcal{B}$ in order to yield a feasible solution of (P). When X is decomposable and the solution information is maintained in disaggregated form, using the multipliers θ^* naturally leads to construct "hybrid" solutions where the components x^h belonging to different solutions of the Lagrangian relaxations are mixed together. Consider for instance (UMMCF): the set of solutions relative to commodity h ($\mathcal{B}^h$) contains paths from o^h to d^h, and the multipliers θ^* can be interpreted as the probability that a given path belong to an optimal solution to (P).

- When the Lagrangian relaxation decomposes in subproblems with different structure, the role of the corresponding components of $\tilde{x}$ may be different. The case where only some components of $\tilde{x}$ correspond to integer variables is particularly interesting, as shown later on.

- Of course, exploiting $\tilde{x}$ or θ^* does not rule out exploiting the solution $\bar{x}$ of the Lagrangian relaxation and the Lagrangian costs $\tilde{c}$ as well; in fact, the combined use of all this information can be very effective.

The meaning of some of the above considerations can be made clearer by considering some actual Lagrangian heuristic implemented for one specific difficult optimization problem.

The hydro-thermal Unit Commitment (UC) problem in electrical power production (Zhuang and Galiana (1988), Bacaud, Lemaréchal, Renaud and Sagastizábal (2001), Borghetti, Frangioni, Lacalandra and Nucci (2003)) is as follows. A set P of thermal power generation units (burning some type of fuel) and a set H of hydro power generation units (exploiting waterfalls in river basins) are given on a discretized time horizon. Each unit is carachterized by a set of technical constraints. For instance, for thermal units it is often required that whenever an unit is turned on it must remain active (committed) for at least a given number of consecutive time instants, and, analogously, whenever an unit is turned off it must remain inactive (decommitted) for at least a given number of consecutive time instants. Hydro units do not have such constraints, but the power they can produce is limited by the total quantity of water that can be extracted from the basin at any given period of time. The objective of the problem is to determine, in each time period, the commitment status of thermal units and the power production of both termal and hydro units so as to satisfy a forecasted power demand at minimal total (essentially fuel) cost. The problem can be written as a large-scale (due to the large number of units and time instants) nonlinear Mixed-Integer problem, where the nonlinear objective function takes into account complex phenomena like the impact of the cooling of a thermal unit over time on its start-up cost. The problem is highly structured; the variable set x is subdivided into "combinatorial" variables z, corresponding to the 0-1 commitment decisions, and "continuous" variables w corresponding to power production decisions (w_p relative to thermal units and w_h relative to the hydro ones). The Lagrangian relaxation of the power demand constraints which link together w_p and w_h decomposes into $|P| + |H|$ independent problems, one for each unit; the "thermal problems" can be solved by dynamic programming, while the "hydro problems" can be solved by means of flow algorithms – that is, the technical constraints on each hydro unit define a convex set.

Apart from providing tight lower bounds on the optimal objective function value of the original problem, the Lagrangian relaxation can be used to construct good solutions. In Borghetti, Frangioni, Lacalandra and Nucci (2003) the following three-stage heuristic is provided:

- some power production $\hat{w}_h$ for the hydro units is fixed, and total power demand is decreased accordingly;
- a greedy heuristic is used to decide a commitment status $\check{z}$ of the thermal units in order to guarantee that the remaining power demand can be satisfied;
- finally, the actual power production (w_p, w_h) of thermal and hydro units is determined by solving a large-scale convex program (the "economic dispatch" problem), given the commitment status $\check{z}$ of thermal units.

Such complex heuristic is motivated by the fact that adjusting the commitment status of thermal units is relatively simple because the commitment decision at time t directly impacts only commitment decisions in a small set of time instants centered on t, while changing the power output of some hydro units at a certain time instant potentially impacts the hydro power output of the units in all the time horizon. However, once the combinatorial decisions have been taken, the remaining continuous problem is easy.

A number of possibile choices exist for implementing the above scheme. First of all, the "fixed " hydro power production $\hat{w}_h$ can be the hydro solution $\bar{w}_h$ of the latest Lagrangian problem, or the "convexified" hydro power production $\tilde{w}_h$ of the latest primal Master Problem; since the constraints involving w_h define convex sets, this is a feasible solution. The greedy heuristic has then to decide which units have to be turned on to satisfy the remaining power demand; for this purpose, a priority list of units is formed to decide which ones are more "promising" at any given time instant. This priority list can be based either on the Lagrangian cost of turning on a unit, or on the "convexified" committment status $\tilde{z}$ of the latest primal Master Problem, interpreted as a "probability" that the unit should be turned on at any time period, or on a combination of

	$\bar{w}_h, \tilde{c}$	$\bar{w}_h, \tilde{z}$	$\tilde{w}_h, \tilde{c}$	$\tilde{w}_h, \tilde{z}$	fs
30, 10	5.02	3.82	1.01	0.57	0.99
60, 25	5.74	4.73	1.54	0.69	0.98
90, 40	3.87	3.23	0.94	0.32	0.63

Table 1: Comparison of Lagrangian heuristics for (UC)

both. Finally, the greedy heuristic may construct $\check{z}$ either starting from scratch or starting from the commitment matrix $\bar{z}$ produced by the solution of the latest Lagrangian problem, modifying it only if needed.

The results of these different variants are shown in Table 1 for some classes of instances. Each class is characterized by the pair (p, h) of, respectively, the number of thermal and hydro units; for all instances, a time horizont of 24 instants (a day) is used. For each class, the average gap obtained by five variants of the Lagrangian heuristic are reported. The first four variants are labelled by a pair (w, z) where $w \in \{\bar{w}_h, \tilde{w}_h\}$ and $z \in \{\tilde{c}, \tilde{z}\}$; this indicates whether the "Lagrangian" hydro power $\bar{w}_h$ or the "convexified" hydro power $\tilde{w}_h$ is fixed, and whether the Lagrangian costs $\tilde{c}$ or the "convexified commitment" $\tilde{z}$ are used to drive the greedy heuristic. In all these cases, the greedy heuristic modifies from the "Lagrangian" commitment matrix $\bar{z}$; the fifth column, labelled fs, uses the same heuristic as column $(\tilde{w}_h, \tilde{z})$ except that the $\check{z}$ is constructed "from scratch".

The results clearly show that using the "convexified" hydro power $\tilde{w}_h$ from the Master Problem produces much better results than using the Lagrangian hydro power $\bar{w}_h$, and that using the "convexified" committment status $\tilde{z}$ produces much better results than using only the Lagrangian costs $\tilde{c}$. Thus, the primal "convexified" information produced by the Lagrangian dual is very useful to construct efficient heuristics for the problem. Even more interesting is perhaps the fact that the versions of the heuristic exploiting the Lagrangian commitment $\bar{z}$ are more efficient than those starting from scratch, even if the "convexified" committment status $\tilde{z}$ is used; in other words, in this case the *combination* of the integer solution produced by the Lagrangian relaxation and the continuous solution produced by the Master Problem turns out to be the most effective one. This shows that a Lagrangian approach produces *even more* valuable information than "just" the optimal solutions of (D) and $(\tilde{P})$.

Each time a Lagrangian heuristic is implemented, both lower and upper bounds are at easy hand; it is thus a natural choice to design an enumerative (Branch and Bound) approach which is based on such bounds.

4 Enumerative algorithms

The main use of Lagrangian approaches within a Branch and Bound (B&B) is that of computing (hopefully tight) bounds on the optimal value of the problem, in order to fathom as many nodes in the search tree as possible. As shown in Section 1, the Lagrangian bound is equivalent to that of the continuous relaxation if the Lagrangian relaxation has the integrality property. However, the mere existence of an equivalent linear formulation does not mean that the Lagrangian approach has to be discarded in favor of one based on standard Linear Programming tools.

For instance, the Lagrangian relaxation described in the Introduction for computing a lower bound on (UMMCF) is equivalent to solving a continuous Multicommodity Min Cost Flow problems. The sheer scale of the (LP) may not allow to efficiently (if at all) solve it by standard Linear Programming codes, whereas a Lagrangian approach may be feasible. A small sample is provided in Table 4, taken from Frangioni and Gallo (1999), for problems with $|K| = 256$ and increasing graph size (the instances have $|A| \approx 4|N|$); the row Bdl and Cpx report the running time required to solve the problem using a proximal bundle approach and `Cplex 6.0` respectively, with a "*" indicating that the corresponding problem is not solved due to memory shortage). Both methods provide solutions with at least 6-digits accuracy, and the Lagrangian approach provides at

$\|N\|$	64	128	256	512	1024
Bdl	17.52	48.72	458.36	218.84	898.51
Cpx	668.27	3525.27	56756.90	*	*

Table 2: Comparison of (LP) and Lagrangean methods for (MMCF)

termination the primal optimal solution x^* together with the dual bound.

The table clearly shows that (MMCF) instances with very many commodities cannot be solved in reasonable time by standard (LP) techniques, while a Lagrangian approach can be successfully used to compute an optimal solution to the problem. We should point out that *specialized* (LP) techniques, such as those of McBride (1998), Castro (2000), may be competitive with Lagrangian ones – but in fact the Lagrangian approach applied to this case *is* a specialized (LP) technique itself, namely a variant of the Dantzig-Wolfe decomposition algorithm. An interesting feature of the Lagrangian approach is that it can be more easily adapted to related but different problems (e.g. Crainic, Frangioni and Gendron (2001)) with respect to other specialized (LP) techniques.

However, computing the (LP) bound – or a tighter bound if the Lagrangian problem does not have the integrality property – efficiently at the root node is not enough to make a Lagrangian approach a viable alternative to (LP) techniques; it must also effectively support all the operations of a B&B or Branch and Cut (B&C) (Padberg and Rinaldi (1991)) algorithm. In particular, three aspects have to be properly taken into account:

- branching rules: the relaxation must provide appropriate information upon which a good branching decision can be taken;
- valid inequalities: the relaxation must provide appropriate information to be used as input for either standard or custom separation algorithms that provide (hopefully strong) new valid inequalities to tighten the formulation;
- reoptimization: the relaxation must be capable of re-using previously obtained solution information to speed-up the computation of the bound after that some branching decision has been taken and/or some new valid inequalities have been added.

It is worth noting here that the use of Lagrangian techniques embedded in enumerative algorithms generalizes the so-called Branch & Price and Branch & Price & Cut algorithms (Barnhart, Hane and Vance (1998)), where the bounding procedure requires the solution of a large-scale (LP) by means of a column generation approach; in fact, any Lagrangian dual can be seen as a large-scale (LP). Thinking the approach in Lagrangian terms can be useful just because it suggests the application of the several different algorithms developed for NonDifferentiable Optimization apart from the pure cutting plane one. This may be useful even for problems that are most naturally viewed in terms of column generation, since e.g. the stabilized variants of the cutting plane algorithms can be remarkably more efficient than the non-stabilized ones (Ben Amor (2002)). One may also remark that the "& Price" tag is somewhat superfluous, since combined column and row generation has been a part of the B&C approach since the very definition of the approach (Padberg and Rinaldi (1991); in other words, using Lagrangian techniques can be seen as just complementing simplex or interior-point methods with ideas from the NonDifferentiable Optimization world.

4.1 Branching rules

One of the main factors discouraging the use of Lagrangian techniques within B&B approaches has been the perceived need to develop ad-hoc branching rules based on the optimal solution $\bar{x}$ of the Lagrangian problem and/or the Lagrangian costs $\tilde{c}$. Although some general ways for constructing such rules have been proposed (Geoffrion (1974), Guignard and Kim (1987)), they have been most often perceived as more difficult and less efficient that standard branching rules based on fractional variables.

However, "modern" algorithms for the solution of the Lagrangian Dual also provide the primal solution x^*, that is clearly the ideal candidate for implementing branching rules based on fractional variables. In fact, an integer x^* is optimal for (P), and thus each time the Lagrangian dual reports a nonzero duality gap some fractional entry of x^* exists, exactly as when a standard (LP) is used. We should note here that although Lagrangian approaches can in theory – and often in practice too – produce primal solutions of the same accuracy as those generated by an (LP) approach, the "quality" of such solutions from a branching standpoint may be quite different, especially on highly degenerated problems. Common experience is that the solutions produced by Lagrangian approaches tend to be "more fractional" than basic solutions produced by a simplex algorithm, when applicable, and possibly more similar to the "central" solution produced by interior point methods. This fact may have an impact on the branching rules and on the overall efficiency of the B&B method whose magnitude is still not very well understood. Yet, at least in principle all the classical branching rules developed for (LP)-based B&B algorithms can be used for Lagrangian-based ones. For instance, choosing the "most fractional" variable x_i^* and adding the constraints $x_i \geq \lceil x_i^* \rceil$ and $x_i \leq \lfloor x_i^* \rfloor$ to the two sons of the current node in the enumeration tree is always possible, such as branching on special-ordered sets or strong branching.

The previous assertion has to be qualified if the branching decision is, as commonly happen, directly implemented in the Lagrangian relaxation, that is – assuming the above binary branching for simplicity of notation – if the branching constraints are added to the Lagrangian problem, thereby impacting the "easy" set X. In this case, particular attention has to be paid in preventing the branching rules from affecting the structure of the Lagrangian subproblem.

Consider for instance (UMMCF) with the above binary branching, that is, where a fractional variable x_{ij}^h is selected and fixed to 0 and 1 in the two subproblems respectively. Fixing a set of variables to 0 corresponds to removing the relative arcs from the graph the Shortest Path problem (for that commodity) is computed on; however, fixing a set of variables to 1 in the Lagrangian problem requires to compute a shortest path which touches a given set of arcs, which is known to be a $\mathcal{NP}$-hard problem. Thus, careless branching may destroy the structure upon which the Lagrangian approach is based. However, it is usually possible to design branching rules that are compatible with the structure of X. For the (UMMCF) case, for instance, a suitable rule is proposed in Barnhart, Hane and Vance (1998) in the context of a Branch & Price & Cut approach. A "divergence" node is selected, that is, a node i with two outgoing arcs with nonzero (hence fractional) flow for one commodity h. Its forward star is partitioned in two subsets, each one containing one of the arcs with nonzero flow. Then, in each son of the current node in the enumeration tree, the flow variables of all the arcs in one subset are fixed to zero. Note that, when a Lagrangian approach is used to compute the fractional solution, i is a divergence node if there are two paths entering i, produced at two different Lagrangian iterations, which have a nonzero optimal multiplier θ_x^* and which leave i along two different arcs.

However, "directly" implementing the branching decision in the Lagrangian subproblem is not the only choice. In fact, once again considering the simple binary branching rule for simplicity of notation, an alternative is to consider the branching constraints $x_i \geq \lceil x_i^* \rceil$ and $x_i \leq \lfloor x_i^* \rfloor$ among the "complicating" ones, and relax them (each one in the corresponding node of the B&B tree). This clearly does not impact on the structure of X, but it changes the Lagrangian function, adding one new Lagrangian variable. Thus, branching can be considered as a special case of producing valid inequalities – for the subproblems in each node – which lead us to the next point.

4.2 Valid inequalities

The use of "strong" valid inequalities for the convex hull of the feasible region is well-known to be essential to enhance the performances of enumerative approaches for very many difficult optimization problems. Thus, the ability to efficiently accommodate generation of valid inequalities within the lower bound computation is crucial for every bounding approach that aims at being widely applicable.

Although in some cases separation can be performed on the (integer) solution of the Lagrangian problem (Gavish (1985), Gouveia (1995)), most separation algorithms input a fractional solution to be "cut"; thus, the ability of producing fractional solutions is usually required in order to accommodate the vast majority of valid inequalities. Once again, the "convexified" fractional solution x^* is clearly the ideal candidate here; in fact, when the Lagrangian relaxation has the integrality property this is exactly the optimal solution of the equivalent (LP) – barring the issue relative to being possibly "more fractional", which is however likely to be less important in this case. Note that when, conversely, X does not have the integrality property, x^* belongs to $conv(X)$ and therefore none of the valid inequalities for $conv(X)$ need to be generated; in other words, using a "difficult" Lagrangian problem is equivalent to adding to an "easy" (LP) all the defining valid inequalities for $conv(X)$, except that the extreme points of $conv(X)$, rather than its faces, are actually generated. Finally, as for the branching rules we can safely assert that basically all separation algorithms devised for (LP)-based B&C approaches can be used within Lagrangian-based ones. The use of Lagrangian techniques can be preferable to that of (LP) techniques, as shown in the example below.

The well-known Max-Cut problem, which requires to find the minimum-weight cutset of a given weighted graph $G = (N, V)$, can be formulated as

$$\max\{cx \ : \ Ax \leq b \ , \ x \in \{0,1\}^{|V|} \ \}$$

where the *triangle inequalities* $Ax \leq b$ define the *semimetric polytope* (for complete graphs):

$$\mathcal{M}: \qquad \left\{ \begin{array}{rl} x_{ij} + x_{ik} + x_{jk} \leq 2 & \\ x_{ij} - x_{ik} - x_{jk} \leq 0 & 1 \leq i < j < k \leq n \\ -x_{ij} + x_{ik} - x_{jk} \leq 0 & \\ -x_{ij} - x_{ik} + x_{jk} \leq 0 & \end{array} \right. . \qquad (10)$$

The continuous relaxation of the above (ILP) is a sparse Linear Program with $O(|N|^2)$ variables and $O(|N|^3)$ constraints; somewhat surprisingly, this problem is nontrivial to solve with standard (LP) technology for large graphs, even if a row generation approach is used (since the number of triangle inequalities is polynomial, the separation problem is easy), as shown below. It may be worth noting that, other than for its relationships with the Max-Cut problem, the semimetric polytope is relevant in several areas of mathematics (Deza and Laurent (1997)), e.g., feasibility of Multicommodity flows (Lomonosov (1985)). Hence, the problem of optimizing a linear function over $\mathcal{M}$ has several relevant applications.

Although at a first glance, the Max-Cut problem does not seem to possess much of a structure, a Lagrangian relaxation is still possible with respect to the whole constraints set $Ax \leq b$; this leaves $X = \{0,1\}^{|V|}$ as the "easy" set, and therefore the corresponding Lagrangian dual is equivalent to the continuous relaxation. However, solving the Lagrangian dual with a cutting-plane-type method can be remarkably more efficient than using state-of-the-art (LP) technology for solving the continuous relaxation. This is shown in Table 4.2 for complete random graphs with 25, 50, 100 and 150 nodes: the column "Cpx" reports the running time required to solve the continuous relaxation of Max-Cut using `Cplex 7.5`, while the column "Bdl" reports the running time required to solve the corresponding Lagrangian dual with a proximal bundle approach. In both cases a row generation approach is used, at least 8-digits precision is obtained in the objective function value and a time limit of 10000 seconds is set.

The table shows very clearly that for graphs with 100 nodes the Lagrangian approach is already much faster, and for larger graphs the standard (LP) approach is not capable of solving the problems within the required time limit, while the Lagrangian approach solves them in less than 10 minutes. Actually, the Lagrangian approach can be made even more efficient by exploiting a combinatorial structure "hidden" in proper subsets of the whole set of triangle inequalities, as shown in Frangioni, Lodi and Rinaldi (2003). The success of the Lagrangian approach in this case mainly depends on the fact that the separation routine for collecting new violated triangle

$\lvert N \rvert$	Cpx	Bdl	Bdl-O	Bdl-D
25	0.48	0.95	4.39	13.53
50	14.93	12.54	157.22	490.52
100	862.48	182.15	2212.06	10000+
150	10000+	532.89	10000+	10000+

Table 3: Comparison of (LP) and Lagrangean methods for Max-Cut

inequalities is called every few iterations of the cutting-plane-like algorithm for the solution of (D), using the (potentially unfeasible w.r.t. the constraints $Ax \leq b$) solution $\tilde{x}$ of the Master Problem; that is, separation is attempted early on with a (potentially very rough) approximation of the optimal solution x^* for the given set of rows. The importance of this choice is shown by column Bdl-O which report the running time required by the bundle method if the separation routine is called only when x^* available; not surprisingly, using a Lagrangian approach with such a "weak" structure to replace an efficient (LP) solver is not convenient unless the extra flexibility provided by the different algorithm is exploited. This example shows that the row generation strategies required by B&C approaches can be efficiently integrated with Lagrangian algorithms.

We should note at this point that, in general, the constraints resulting from the separation algorithms cannot be treated inside the Lagrangian subproblem; thus, they have to be treated as new "complicating" constraints and relaxed. Hence, at each node of the enumeration tree we have to deal with a different Lagrangian function; in particular, the number of Lagrangian multipliers may significantly increase when descending towards the leaves of the tree. Since each Lagrangian problem is – being NonDifferentiable – nontrivial to solve, especially as the number of Lagrangian multipliers grows (Lemaréchal (2001), there is the concrete possibility that a Lagrangian approach may become less and less competitive w.r.t. a standard (LP) approach as the enumeration proceeds. Furthermore, the simplex method is known to reoptimize very efficiently after the data changes corresponding to branching decisions and generation of valid inequalities (in fact, it is still the preferred algorithm to be used within a B&B approach, even for the problems where it is outperformed by interior point methods at the root node), thus a Lagrangian method that is even very competitive at the root node may still not be competitive as far as the time required to solve the (many, closely related) relaxations at all nodes is concerned.

Yet, in many cases where B&C approaches are effective the overall number of enumeration nodes is pretty low, helping to limit the possible "curse of dimensionality" in which a Lagrangian approach could incur. Furthermore, "modern" Lagrangian approaches are able to reoptimize as well, which lead us to the final point.

4.3 Reoptimization

Upon termination of an algorithm for solving (D) a set of relevant information about the problem has been collected; this is $\mathcal{B}$, a set of points in $conv(X)$ whose convex hull contains x^*. The discussion in Section 2.6 shows that the fundamental difference between subgradient-type and cutting-plane-type algorithms lies precisely in the size of the corresponding set $\mathcal{B}$: very small for the former – essentially containing only x^* at termination – and potentially larger for the latter. The information in $\mathcal{B}$, together with the optimal solution y^* of (D), can be exploited to speed-up the solution of a different but related Lagrangian dual, thereby suggesting that cutting-plane-type algorithms may be better suited than subgradient methods for building enumerative approached. We also mention that, depending on the form of the Master Problem and on the algorithm that is used to solve it, some other information from the previous problem, such as the "active set" of nonzero θ multipliers (Frangioni (1996)), may be kept which can be very useful to reduce the – possibly large (Crainic, Frangioni and Gendron (2001), Lemaréchal (2001)) – time spent in the solution of the Master Problem.

Reoptimization for a cutting-plane-like approach after a change in the Lagrangian function corresponding to the insertion of new "complicating constraints" in the formulation (either valid inequalities or branching constraints) is quite easy. If a new set of constraints $A'x = b'$ is appended after $Ax = b$, the value of the old Lagrangian function ϕ in y^* is equal to the value of the new Lagrangian function

$$\phi'(y, y') = \min_x \{ \ cx + y(b - Ax) + y'(b' - A'x) : x \in X \ \}$$

in $(y^*, 0)$; in other words, an algorithm for optimizing ϕ' can be provided with a starting point whose $\phi'-$value is the optimal value of ϕ. Furthermore, if the primal information $x \in \mathcal{B}$ is explicitly kept, as opposed to only the corresponding dual information $g = b - Ax$ for $x \in \mathcal{B}$ being kept, it is immediate to exploit it to construct a new Master Problem that contains as much information as the last one in the previous optimization; in the pure Cutting Plane case, for instance, this is just

$$\max_{v,y,y'} \{ \ v : v \leq cx + y(b - Ax) + y'(b' - A'x) \ , \ x \in \mathcal{B} \ \} \ .$$

In dual words, each available vector $g = b - Ax$ for some $x \in \mathcal{B}$, which was a subgradient of ϕ at some y, can be immediately "extended" to a subgradient $[g, g' = b' - A'x]$ of ϕ' in $[y, 0]$. Thus, inserting – and relaxing – new complicating constraints in the problem is equivalent to an implicit "active set" method on the Lagrangian variables; one is in fact solving a Lagrangian dual with "many" multipliers by allowing at each time only a "small" set of them to be nonzero, and revising the set from time to time. Although non completely trivial, this approach can be implemented into general-purpose Lagrangian solvers without too much of a problem.

The impact of reoptimization in the Max-Cut case is shown in column Bdl-D of Table 4.2, which reports the running time required by the proximal bundle method if the current set $\mathcal{B}$ is emptied each time new triangle inequalities are separated, i.e., if the Lagrangian optimization process is restarted each time from scratch (actually, in that case the previous optimal point y^* were retained). Clearly, exploiting the information provided by the set $\mathcal{B}$ is crucial for obtaining a competitive approach; this is even more important in the Max-Cut case because, as discussed in the previous section, it is crucial to generate new violated inequalities as frequently as possible, and therefore the Lagrangian approach would basically be restarted every few iterations if reoptimization would not be used.

Reoptimization is only a little bit trickier when branching decisions are "directly" implemented in the Lagrangian relaxation. Sticking to a binary branching case for simplicity of notation, the feasible set $\bar{X}$ of the Lagrangian relaxation at the current node is subdivided into the (non necessarily disjoint) sets $\bar{X}_0$ and $\bar{X}_1$ corresponding to its descendants in the enumeration tree. The set $\mathcal{B}$, which contains points of $conv(\bar{X})$, can therefore be similarly subdivided – if the primal information is explicitly kept, as opposed to only the corresponding dual information being kept – into the two subsets $\mathcal{B}_0 = \mathcal{B} \cap conv(\bar{X}_0)$ and $\mathcal{B}_1 = \mathcal{B} \cap conv(\bar{X}_1)$; these can be used, together with the previous dual optimal solution y^* (and potentially active-set information for the Master Problem), to warm-start the optimization of the Lagrangian dual corresponding to each son in the enumeration tree. Note that most "sensible" branching decision must force both $\mathcal{B}_0$ and $\mathcal{B}_1$ to be strict subsets of $\mathcal{B}$, since otherwise the previous optimal solution x^* of $(\tilde{P})$ has remained feasible for some of the descendants, and therefore the bound is not improving after branching. For instance, branching on a binary variable with fractional value x_i^* means that $\mathcal{B}$ is partitioned into the subset $\mathcal{B}_0$ containing all $\bar{x}$ such that $\bar{x}_i = 0$, the subset $\mathcal{B}_1$ containing all $\bar{x}$ such that $\bar{x}_i = 1$ and, possibly, the subset $\mathcal{B}_f$ containing all $\bar{x}$ such that $\bar{x}_i$ is fractional. Thus, some information is "lost" in each son in the enumeration tree; on the other hand, both sons are likely to start with a nonempty $\mathcal{B}$, at least if an algorithm allowing a "large" $\mathcal{B}$ is used.

It is interesting to mention at this point that another potentially relevant advantage of cutting-plane-type approaches over subgradient-type ones is their ability to more easily exploit multiple solution information if this is provided by the Lagrangian relaxation. In fact, although inserting at least one optimal solution of $(P_{\tilde{y}})$ in $\mathcal{B}$ is necessary for convergence of the algorithms, nothing prevents one from inserting multiple solutions if they can be (cheaply) obtained. Since collecting the right information is the main task of any such algorithm, and the solution of the Lagrangian relaxation may be a costly process, this may make sense in several applications. Indeed, for several column generation approaches (Carraresi, Girardi and Nonato (1995), Ben Amor (2002)) it is well-known that inserting just one column with negative reduced cost at each iteration would lead to a very inefficient algorithm, and Lagrangian optimization is a column generation approach.

It should also be noted that nothing in (4) requires the points $x \in \mathcal{B}$ to be optimal solutions of the Lagrangian relaxation anywhere; all non-extremal points are theoretically dominated by some extremal ones, but chances are that during the optimization process the dominating extremal points for some available non-extremal ones are not available. Thus, as in column generation, inserting suboptimal solutions of $(P_{\tilde{y}})$ in $\mathcal{B}$ may be a very sensible move; examples can be found e.g. in Zhao and Luh (2002), Cappanera, Gallo and Maffioli (2003). A larger set $\mathcal{B}$, other than enhancing the convergence rate of a Lagrangian approach, may make it more "resistent" to the loss of information corresponding to direct implementation of the branching decisions in the Lagrangian subproblem.

While all the above points are very clear – almost trivial – from a theoretical standpoint, their practical impact still has to be properly assessed. General guidelines about when a Lagrangian approach may be competitive with a (LP)-based one within an enumerative algorithm are still not available, mostly because they depend from a number of complex issues such as the exact algorithm used for solving the Lagrangian dual – and even the exact implementation of the algorithm used for solving the Master Problem, if any –, the structure of the Lagrangian relaxation, the relative weight of solving the Lagrangian relaxation w.r.t. solving the Master problem, the availability of multiple solution information and many others. With efficient, standard and easy to use general codes for solving Lagrangian duals still not nearly as widely available as codes for solving Linear Programs, it is not surprising that Lagrangian approaches are less commonly used than standard (LP)-based ones. Yet, the true potential of Lagrangian approaches for the solution of Integer Programs does not appear to have been fully tapped, and there seem to be room for extending the use of Lagrangian techniques to many more difficult optimization problems. We believe that the application of Lagrangian approaches for the solution of Integer Programs is a promising research avenue which still deserves attention and development, and we hope to be able to contribute to this process in the future.

5 Conclusion

Solving the Lagrangian dual (D) of an Integer Linear Program (P) is equivalent to solving its "convexified relaxation" $(\tilde{P})$; the equivalence is effective, that is, an (approximate) optimal solution x^* of $(\tilde{P})$ is constructed by most "modern" algorithms for solving (D).

Therefore, Lagrangian techniques provide an alternative to standard Linear Programming techniques for developing exact or approximate solution algorithms for (P). Barring some important but still overall minor points, discussed in the paper, approaches based on Lagrangian techniques can exploit the very same strategies (branching rules, valid inequalities, rounding schemes and so on) developed for (LP)-based approaches. Actually, in some cases Lagrangian techniques offer a "richer" set of information w.r.t. (LP) techniques: not only a continuous solution x^*, but also a set of solutions $\mathcal{B} \subseteq X$ and the relative convex multipliers θ^* that make up x^*. Conversely, the solution x^* built by Lagrangian approaches is typically not basic, and therefore can be "more fractional" than that provided by an ordinary continuous relaxation solved by means of a simplex algorithm.

Despite this, Lagrangian techniques are much less frequently used than (LP) techniques within approaches for (ILP). This is due to at least three reasons:

- Lagrangian techniques call for a detailed analysis of the model and the solution of optimization subproblem; therefore, they are more difficult to develop, less resistant to changes of the optimization problem (that may easily destroy the structure upon which they are based), and require at least some form of programming, while (LP) techniques can be used by less experienced personnel through modeling engines and standard solvers, which makes them preferable in several environments.

- The available general codes for the solution of Lagrangian Duals are still far from the level of robustness, efficiency, ease of use and standardization currently offered by (LP) codes.
- Solving a Lagrangian dual with high precision can be a daunting task in many cases, both for the lack of really efficient codes and for the intrinsic difficulty of the problem itself.

Thus, the use of Lagrangian techniques is limited both by the intrinsic characteristics of the approach and by some technological factors, mainly the availability (or lack thereof) of efficient and easy to use codes for the solution of Lagrangian duals. Improvements in the available Lagrangian technology might allow to successfully apply these techniques to many more difficult optimization problems of both academic and practical interest, providing one more useful tool in the “bag of tricks” available for solving difficult optimization problems.

Acknowledgements

I'm indebted with Giorgio Gallo, Alberto Borghetti, Fabrizio Lacalandra, Carlo Alberto Nucci, Andrea Lodi, and Giovanni Rinaldi for their fundamental contribution in the various pieces of work from which some of the results presented herein are taken. I also acknowledge Paola Cappanera's assistance with the English language.

References

Bacaud, L., C. Lemaréchal, A. Renaud, and C. Sagastizábal. (2001). ”Bundle Methods in Stochastic Optimal Power Management: A Disaggregated Approach Using Preconditioners,” *Computational Optimization and Applications* 20, 227–244.

Bahiense, L., N. Maculan and C. Sagastizábal. (2002). ”The volume algorithm revisited: relation with bundle methods,” *Math. Prog.* 94(1), 41–70.

Balakrishnan, A., T. L. Magnanti and R. T. Wong. (1989). ”A Decomposition Algorithm for Local Access Telecommunications Network Expansion Planning,” *Op. Res.* 37, 716–740.

Balakrishnan, A., T. L. Magnanti and R. T. Wong. (1995). ”A Dual-ascent Procedure for Large-scale Uncapacitated Network Design,” *Op. Res.* 43(1), 58–76.

Barahona, F., and R. Anbil. (2000). ”The Volume Algorithm: Producing Primal Solutions with a Subgradient Method,” *Math. Prog.* 87(3), 385–400.

Barnhart, C., and Y. Sheffi. (1993). ”Dual-Ascent Methods for Large-Scale Multi-Commodity Flow Problems,” *Naval Research Logistics* 40, 305–324.

Barnhart, C., C. A. Hane and P. Vance. (1998). ”Using Branch-and-Price-and-Cut to Solve Origin-Destination Integer Multicommodity Flow Problems,” *Op. Res.* 32(3), 208–220.

Benders, J.F.. (1962). ”Partitioning procedures for solving mixed variables programming problems,” *Numerische Mathematik* 4, 238–252.

Ben Amor, H.. (2002). ”Stabilisation de l'Algoritme de Génération de Colonnes,” *Ph.D. Thesis*, GERAD, Université de Montréal, Montreal, QC, Canada.

Borghetti, A., A. Frangioni, F. Lacalandra and C. A. Nucci. (2002). ”Lagrangian Heuristics Based on Disaggregated Bundle Methods for Hydrothermal Unit Commitment,” *IEEE Transactions on Power Systems* 18, 313–323.

Camerini, P.M., L. Fratta and F. Maffioli. (1975). ”On Improving Relaxation Methods by Modified Gradient Techniques,” *Math. Prog. Study* 3, 26–34.

Cappanera, P., G. Gallo and F. Maffioli. (2003). "Discrete Facility Location and Routing of Obnoxious Activities," *Disc. Appl. Math.*, to appear.

Castro, J.. (2000). "A Specialized Interior-Point Algorithm for Multicommodity Network Flows," *SIAM J. on Opt.* 10, 852–877.

Carraresi, P., L. Girardi and M. Nonato. (1995). "Network Models, Lagrangian Relaxations and Subgradient Bundle Approach in Crew Scheduling Problems," in J. Paixao ed. *Computer Aided Scheduling of Public Transport*, Lecture Notes in Economical and Mathematical Systems, Springer-Verlag.

Chang, S.-G., and B. Gavish. (1995). "Lower Bounding Procedures for Multiperiod Telecommunications Network Expansion Problems," *Op. Res.* 43(1), 43–57.

Crainic, T.G., A. Frangioni and B. Gendron. (2001). "Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems," *Discrete Appl. Math.* 112, 73–99.

Dantzig, G.B., and P. Wolfe. (1960). "The Decomposition Principle for Linear Programs," *Op. Res.* 8, 101–111.

Dentcheva, D., A. Prékopa and A. Ruszczynski. (2002). "Bounds for integer stochastic programs with probabilistic constraints," *Discrete Appl. Math.* 124, 55–65.

Deza, M.M., and M. Laurent. (1997). *Geometry of Cuts and Metrics*, Algorithms and Combinatorics 15, Springer-Verlag, Heidelberg.

du Merle, O., J.-L. Goffin and J.-P. Vial. (1998). "On Improvements to the Analytic Center Cutting Plane Method," *Computational Optimization and Applications* 11, 37–52.

Farwolden, J.M., and W.B. Powell. (1994). "Subgradient Methods for the Service Network Design Problem," *Trans. Sci.* 28(3), 256-272.

Frangioni, A.. (1996). "Solving semidefinite quadratic problems within nonsmooth optimization algorithms," *Comput. & Oper. Res.* 23, 1099–1118.

Frangioni, A.. (2002). "Generalized Bundle Methods," *SIAM J. on Opt.* 13(1), 117–156.

Frangioni, A., and G. Gallo, "A Bundle Type Dual-Ascent Approach to Linear Multicommodity Min Cost Flow Problems," *INFORMS J. on Comp.* 11(4), 370–393.

Frangioni, A., A. Lodi and G. Rinaldi. (2003). "Lagrangian Approaches for Optimizing over the Semimetric Polytope," *working paper.*

Fréville, A., M. Guignard and S. Zhu. (1999). "Column generation and Lagrangean relaxation: two powerful related tools for Integer Programming," *OPIM Department Report* 99-12-22, The Wharton School, U. of Pennsylvania.

Fuduli, A., and M. Gaudioso. (2000). "Fixed and virtual stability center methods for convex nonsmooth minimization," in G. Di Pillo and F. Giannessi (eds.), *Nonlinear Optimization and Applications 2*, Kluwer Academic Publishers, 105–122.

Gavish, B.. (1985). "Augmented Lagrangian Based Algorithms for Centralized Network Design," *IEEE Trans. on Communications* 33, 1247–1257.

Geoffrion, A.M.. (1974). "Lagrangian Relaxation and its Uses in Integer Programming," *Math. Prog. Study* 2, 82–114.

Goffin, J.-L., J. Gondzio, R. Sarkissian and J.-P. Vial. (1997). "Solving Nonlinear Multicommodity Flow Problems by the Analytic Center Cutting Plane Method," *Math. Prog.* 76, 131–154.

Gouveia, L.. (1995). "A $2n$ Constraint Formulation for the Capacitated Minimal Spanning Tree Problem," *Op. Res.* 43, 130–141.

Grigoriadis, M.D., and L. G. Kahchiyan. (1995). "An exponential-function reduction method for block-angular convex programs," *Networks* 26, 59–68.

Guignard, M., and S. Kim. (1987). "Lagrangian Decomposition: a Model Yelding Stronger Lagrangian Bounds," *Math. Prog.* 39, 215–228.

Guignard, M.. (1998). "Efficient cuts in Lagrangean 'Relax-and-Cut' schemes," *EJOR* 105, 216–223.

Guignard, M.. (2003). "Lagrangean Relaxation," *TOP* 11(2), 151–228.

Guignard, M., and M.B. Rosenwein. (1989). "An application-oriented guide for designing Lagrangian dual ascent algorithms," *EJOR* 43, 197–205.

Held, M., and R. Carp. (1971). "The travelling salesman problem and minimum spanning trees: Part II," *Math. Prog.* 1, 6–25.

Held, M., P. Wolfe and H.P. Crowder. (1974). "Validation of Subgradient Optimization," *Math. Prog.* 6, 62–88.

Hiriart-Urruty, J.-B., and C. Lemaréchal. (1993). *Convex Analysis and Minimization Algorithms* II—*Advanced Theory and Bundle Methods*, Grundlehren Math. Wiss. 306, Springer-Verlag, New York.

Jones, K.L., I.J. Lustig, J.M. Farwolden and W.B. Powell. (1993). "Multicommodity Network Flows: The Impact of Formulation on Decomposition," *Math. Prog.* 62, 95–117.

Kiwiel, K.. (1999). "A bundle Bregman proximal method for convex nondifferentiable optimization," *Math. Prog.* 85, 241–258.

Kelley, J.E.. (1960). "The Cutting-Plane Method for Solving Convex Programs," *Journal of the SIAM* 8, 703–712.

Larsson, T., M. Patriksson and A.-B. Strömberg. (1999). "Ergodic, Primal Convergence in Dual Subgradient Schemes for Convex Programming," *Math. Prog.* 86, 283–312.

Lemaréchal, C.. (2001). *Lagrangian Relaxation*, in M. Jünger and D. Naddef (eds.) *Computational Combinatorial Optimization*, Springer Verlag, Heidelberg, 115-160.

Lemaréchal, C., A. Nemirovskii, and Y. Nesterov. (1995). "New variants of bundle methods," *Math. Prog.* 69, 111–147.

Lomonosov, M.V.. (1985). "Combinatorial Approaches to Multiflow Problems," *Discrete Appl. Math.* 11(1), 1–93.

McBride, R.D.. (1998). "Advances in Solving the Multicommodity Flow Problem," *SIAM J. on Opt.* 8, 947–955.

Nesterov, Y.. (1995). "Complexity estimates of some cutting plane methods based on the analytic barrier," *Math. Prog.* 69, 149–176.

Padberg, M.W., and G. Rinaldi. (1991). "A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems," *SIAM Rev.* 33, 60–100.

Pinar, M.C., and S.A. Zenios. (1994). "On smoothing exact penalty functions for convex constrained optimization," *SIAM J. on Opt.* 4, 486–511.

Poljak, B.T.. (1977). "Subgradient Methods: a Survey of Soviet Research", in C. Lemaréchal and R. Mifflin (eds.) *Nonsmooth Optimization*, IIASA Proceedings Series vol. 3, Pergamon Press.

Schultz, R.. (2003). "Stochastic programming with integer variables," *Math. Prog.* 97, 285–309.

Takriti, S., and J. R. Birge. (2000). "Lagrangean solution techniques and bounds for loosely coupled mixed integer stochastic programs," *Op. Res.* 48, 91–98.

Vazirani, V.V.. (2001). *Approximation Algorithms*, Springer-Verlag, New York.

Zhao, X., and P. B. Luh. (2002). "New Bundle Methods for Solving Lagrangian Relaxation Dual Problems," *JOTA* 113(2), 373–397.

Zhuang, F., and F.D. Galiana. (1988). "Towards a more rigorous and practical unit commitment by Lagrangian relaxation," *IEEE Transactions on Power Systems* 3, 763–773.