Ann Oper Res (2007) 153: 9–27
DOI 10.1007/s10479-007-0169-1

The omnipresence of Lagrange

Claude Lemaréchal

Published online: 17 May 2007

Abstract Lagrangian relaxation is usually considered in the combinatorial optimization community as a mere *technique*, sometimes useful to compute bounds. It is actually a very general *method*, inevitable as soon as one bounds optimal values, relaxes constraints, convexifies sets, generates columns, etc. In this paper we review this method, from both points of view of theory (to dualize a given problem) and algorithms (to solve the dual by nonsmooth optimization).

Keywords Combinatorial optimization · Lagrange relaxation · Duality · Column generation

This paper is devoted to Lagrangian relaxation. Its earlier version (Lemaréchal 2003) was written in the spirit of (Lemaréchal 2001), which was itself inspired from (Hiriart-Urruty and Lemaréchal 1993); Chap. XII of this latter work is devoted to the theory of Lagrangian relaxation, and its Chap. XV gives a detailed account of bundle methods. For a simplified account in the framework of combinatorial optimization, we can suggest (Geoffrion 1974; Reeves 1993, Chap. 6) among others.

1 The basic idea

Consider an optimization problem, which we write abstractly as:

$$\sup f(x), \quad x \in X, \qquad c(x) = 0 \in \mathbb{R}^m \quad [\text{i.e. } c^j(x) = 0,\ j = 1, \dots, m]. \tag{1}$$

We will call it the primal problem.

This is an updated version of the paper that appeared in *4OR*, *1*(1), 7–25 (2003).

C. Lemaréchal (✉)
655 avenue de l'Europe, Montbonnot, 38s334 Saint Ismier, France
e-mail: Claude.Lemarechal@inrialpes.fr

Remark 1 Some preliminary observations are already worth mentioning:

(i) Notationally, we write sup rather than max because existence of an optimal solution to (1) is not guaranteed yet. We will continue to use this notation throughout, to suggest the fact that we are generally more interested by values of *functions*, than by values of their *arguments*.
(ii) No assumption is made on f, X, c, at least for the moment. In particular, the set X is completely abstract: it could be a finite set, the space $\mathbb{R}^n$, or something more general. One of our aims is to illustrate the flexibility of Lagrangian relaxation, with many possible forms for X.
(iii) In a way, our sole assumption concerns the set of constraint-values: $\mathbb{R}^m$, considered as a Euclidean space (which could even be infinite-dimensional). To make it simple, we will denote by $u^\top c$ its inner product, even though something like $\langle u, c\rangle$ would be preferable. The reason we mention this subtlety will come in Sect. 2.6, where $\mathbb{R}^m$ will be the space $\mathbb{R}^{p(p+1)/2}$ of symmetric $p \times p$ matrices. Because $u^\top c$ is definitely not an inner product when u and c are two matrices, we will have to use a special notation there.
(iv) Considering both a constraint function $c(x)$ and a set X, which may itself be defined by constraints, is somewhat redundant. The difference will become apparent in (2, 3) below. Actually, the crux of Lagrangian relaxation precisely lies there; more will be said on this in Remark 5 and Sect. 2.

Introduce the Lagrangian, a function of the primal variable $x \in X$, but also of the *dual* variable $u \in \mathbb{R}^m$:

$$X \times \mathbb{R}^m \ni (x, u) \mapsto L(x, u) := f(x) - u^\top c(x). \tag{2}$$

In other words, the constraints $c(x) = 0$ are *relaxed*, or *dualized*; by contrast, X (the "environment set") gathers the *hard* constraints, those that are kept intact in the duality scheme (1, 2).

If, for fixed $u \in \mathbb{R}^m$, we maximize $L(x, u)$ over $x \in X$, we obtain a well-defined number which depends on the particular u. We call it the *dual function* associated with (1, 2):

$$\theta(u) := \sup_{x\in X} L(x, u). \tag{3}$$

Definition 2 Lagrangian relaxation consists in solving the so-called *dual problem*

$$\inf_{u\in\mathbb{R}^m} \theta(u). \tag{4}$$

This will be motivated shortly. In anticipation to Sect. 3, let us mention here and now that the dual problem is always easy, in some sense. Lagrangian relaxation will therefore be (possibly useless but) implementable if (3) is itself "easy".

Remark 3 Concerning (3), several cases may occur:

(i) The Lagrangian is unbounded from above; then it is natural to set $\theta(u) = +\infty$, the corresponding u is certainly of no use for the dual problem.
(ii) A finite upper bound does exist ($\theta(u) < +\infty$) but is not attained: think of a Lagrangian like $L(x, u) = u\mathrm{e}^x$ with $u < 0$.

(iii) The Lagrangian attains its maximum at a unique x_u,
(iv) or at several primal points, so that the above x_u is ambiguous.

Examples of (i) will be given below in Sects. 2.1, 2.2. As for (iv), it is in fact a typical situation.

We will assume that a Lagrangian *oracle* is available to maximize $L(\cdot, u)$, producing $\theta(u)$, as well as some corresponding maximizer $x_u \in X$ if there exists at least one (see Fig. 1). We will see in Sects. 3 and 4 that an important object is the constraint value $c(x_u) \in \mathbb{R}^m$. In contrast with $\theta(u)$, which is always a well-defined number (possibly $+\infty$), $c(x_u)$ is well defined only in case (iii) above.

The dual approach has several motivations:

(j) Symmetrically to θ, introduce the function

$$X \ni x \mapsto \varphi(x) := \inf_{u \in \mathbb{R}^m} L(x, u) = \begin{cases} f(x) & \text{if } c(x) = 0, \\ -\infty & \text{otherwise.} \end{cases}$$

Obviously, (1) is nothing other than maximizing $\varphi(x)$ over X; this somewhat artificial formulation reveals a first connection between (1) and (4):

$$\begin{array}{c|c} \text{primal} & \text{dual} \\ \sup\limits_x \inf\limits_u L(x, u) & \sup\limits_u \inf\limits_x L(x, u). \end{array}$$

(jj) A more convincing connection is the well-known *weak duality* property: by definition, $\theta(u) \geq L(x, u)$ for all $x \in X$; and if $c(x) = 0$, then $L(x, u) = f(x)$. Thus any $\theta(u)$ is an upper bound on the optimal value of (1); the dual problem therefore consists in finding the least such upper bound.

(jjj) After all, our ultimate wish is to find u so that the oracle produces an x_u which is primal optimal. For this, one must at least have $c(x_u) = 0$, in which case $L(x_u, u) = f(x_u)$; but since $L(x_u, u) = \theta(u)$ by definition of x_u, we actually have $f(x_u) = \theta(u)$, so we conclude by weak duality: to reach our goal,

- it is *necessary* that u minimizes θ;
- it is *sufficient* that x_u is feasible in (1).

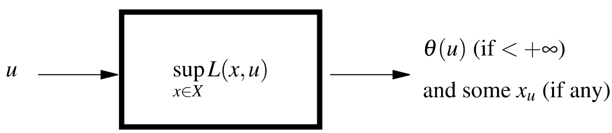
Let us sum up this last motivation: the x_u's obtained from (3) that are also optimal in (1) are exactly those that are feasible (satisfying $c(x_u) = 0$); and the only chance to obtain any of them is to solve the dual problem.

In view of (jj) above, the following concept is relevant.

Definition 4 The *duality gap* is the (nonnegative) difference between the optimal values of (1) and (4).

Accordingly, Lagrangian relaxation will be successful if the duality gap is small, hopefully zero.

Fig. 1 The Lagrangian oracle



Remark 5 It is important to understand that, for a given primal problem (1), there may exist many possible duality schemes, which are just *characterized* by the choice of the Lagrangian. Section 2.2 will give a rudimentary illustration of this.

Selecting an appropriate Lagrangian is actually an art, in which a balance must be found between two elements:

- How easily can (3) be solved?
- How good is the duality gap?

2 Examples

The examples below will illustrate the approach; they will also reveal how versatile and how ubiquitous Lagrangian relaxation can be.

2.1 Inequalities

Suppose that the relaxed constraints in (1) are actually inequalities: we wish to apply duality to

$$\sup f(x), \quad x \in X, \qquad c(x) \leq 0. \tag{5}$$

A simple way to recover the previous framework is to introduce slack variables: our problem can be written

$$\sup f(x), \quad x \in X, \qquad y \geq 0 \in \mathbb{R}^m, \qquad c(x) + y = 0. \tag{6}$$

Changing the primal variable $x \in X$ to $(x, y) \in X \times \mathbb{R}^m_+$, we form the "slackened" Lagrangian $f(x) - u^\top(c(x) + y) = L(x, u) - u^\top y$, where L is still the function (2). Clearly, the dual function is now

$$\sup_{x \in X, y \geq 0} L(x, u) - u^\top y = \begin{cases} \theta(u) & \text{if } u \geq 0, \\ +\infty & \text{otherwise,} \end{cases}$$

θ still being the function (3). The corresponding dual problem is just (4), with the constraint $u \geq 0$ inserted. We recognize a familiar fact: inequality constraints result in signed dual variables.

We have here an illustration of Remark 3(i): the dual function may take on the value $+\infty$; but we have good information concerning the so-called *domain* of the dual function (the set over which it is finite). Thus, the oracle of Fig. 1 will of course not be called with an $u \not\geq 0$. We have also an illustration of Remark 3(iv): in fact, let $u \geq 0$ be given; then the primal point in Fig. 1 has two components:

- One is $x_u \in X$, which may exist or not, be unique or not.
- The other is $y_u \in \mathbb{R}^m$. For a positive component u^j, the corresponding y_u^j is unambiguously 0; but if $u^j = 0$, then the whole half-line $[0, +\infty[$ is valid for y_u^j.

2.2 Linear programming

Let (1) be a linear program in standard form:

$$\sup b^\top x, \quad x \geq 0 \in \mathbb{R}^n, \qquad Ax = a \in \mathbb{R}^m.$$

For illustration, we consider two duality schemes.

Dualizing the linking constraints First take $X := \mathbb{R}^n_+$ and $c(x) := Ax - a$, so that the Lagrangian is

$$b^\top x - u^\top(Ax - a) = \left(b - A^\top u\right)^\top x + a^\top u.$$

The resulting dual function is obtained just as in the previous example:

$$\sup_{x \geq 0} \left(b - A^\top u\right)^\top x + a^\top u = \begin{cases} a^\top u & \text{if } b - A^\top u \leq 0, \\ +\infty & \text{otherwise,} \end{cases}$$

which results in the familiar dual linear program: to minimize $a^\top u$ subject to $A^\top u \geq b$.

Dualizing all constraints Now set $X := \mathbb{R}^n$, so that there are two types of constraints: $Ax - a = 0$ as before (with multipliers $u \in \mathbb{R}^m$), and $-x \leq 0$ (with multipliers $v \in \mathbb{R}^n_+$). This gives raise to the Lagrangian

$$b^\top x - u^\top(Ax - a) + v^\top x = \left(b - A^\top u + v\right)^\top x + a^\top u,$$

which must be maximized over the whole of $\mathbb{R}^n$: the dual function is

$$\sup_{x \in \mathbb{R}^n} \left(b - A^\top u + v\right)^\top x + a^\top u = \begin{cases} a^\top u & \text{if } b - A^\top u + v = 0, \\ +\infty & \text{otherwise.} \end{cases}$$

Needless to say, the extra dual variable v plays no role: in fact $v = A^\top u - b$, which must be nonnegative; we recover the previous dual problem.

With relation to Remark 5, this is an example where both duality schemes are just the same.

2.3 Column generation

Consider a linear version of (1), namely:

$$\sup b^\top x, \quad x \in X, \qquad Ax = a; \tag{7}$$

here X is usually a discrete set; anyway we assume here for simplicity that X is finite. A particular technique to solve (7) is column generation. Its essential idea starts by extracting from X a subset $\{\tilde{x}_k\}_{k=1}^K$ of moderate size, thereby simplifying (7). The resulting restricted problem is further simplified by *convexification*, i.e. we take the convex hull of the $\tilde{x}_k$'s. Thus, (7) is replaced by the *restricted master* program of Dantzig–Wolfe

$$\sup b^\top x, \quad x \in X_K := \operatorname{conv}\{\tilde{x}_1, \ldots, \tilde{x}_K\}, \qquad Ax = a. \tag{8}$$

Remark 6 Column generation can be derived as in (Wolsey 1998, Chap. 11; Vanderbeck 2000), which gives an interesting anticipation to Theorem 12 below. Call $\propto$ (a supposedly big integer) the total number of points in X. Clearly, (7) can be formulated as the equivalent *master* program

$$\sup \sum_{k=1}^{\propto} \alpha_k b^\top \tilde{x}_k, \qquad \sum_{k=1}^{\propto} \alpha_k A\tilde{x}_k = a, \quad \alpha_k \in \{0, 1\}, \ \sum_{k=1}^{\propto} \alpha_k = 1.$$

If we *relax* the 0–1 constraints to $\alpha_k \in [0,1]$ and if we *restrict* the number $\propto$ of columns down to K, we obtain

$$\sup \sum_{k=1}^{K} \alpha_k b^\top \tilde{x}_k, \qquad \sum_{k=1}^{K} \alpha_k A\tilde{x}_k = a, \quad \alpha_k \geq 0, \ \sum_{k=1}^{K} \alpha_k = 1,$$

which is clearly the restricted master (8) in terms of convex components of x.

Now comes the question of iterating the algorithm with a new "column" $\tilde{x}_{K+1}$. For this, call u_K the vector of multipliers associated with $Ax = a$ in the restricted master (8); then $\tilde{x}_{K+1}$ is taken as solving the subproblem or *satellite*: to maximize $b^\top x - u_K^\top Ax$ over the whole of X (this problem has always an optimal solution because X is a finite set). Naturally, the satellite is just equivalent to solving $\sup_{x\in X} L(x, u_K)$, the constant term $a^\top u_K$ being neglected.

The whole business of this technique is to find appropriate polyhedra X_K, i.e. in fact appropriate columns $\tilde{x}_k$, i.e. in fact appropriate multipliers u_k. The role of the optimal solutions of (8)—call them $\hat{x}$—is only to approximate optimal solutions of (7); they have no influence on the progress of the algorithm. Keeping in mind point (jjj) in Sect. 1, we are bound to conclude that column generation does the same thing as Lagrangian relaxation, and with the same tool: it *has to* minimize the dual function with the help of the oracle of Fig. 1. Indeed, we will see in Sect. 4.2 that Dantzig–Wolfe is indeed a particular—and not particularly efficient—algorithm to minimize $\theta(u)$.

Remark 7 Identity between the two approaches is often hard to distinguish because:

- Lagrangian relaxation emphasizes the oracle, which "softens" the linking constraints $Ax = a$ and keeps X as a "hard" set;
- in column generation, one usually does the contrary, emphasizing the restricted master (8) which keeps $Ax = a$ as "hard", while X is "softened" via the operation $X \rightsquigarrow X_K$.

Thus the two approaches tackle the same problem (7), but from two opposite points of view.

Lagrangian relaxation elevates column generation to the level of a general *methodology*, with its theory (Sect. 3) as well as algorithms (Sect. 4), and allowing generalizations of (7) to (1).

2.4 Quadratic programming

With a symmetric $n \times n$ matrix Q, let us dualize the linear–quadratic problem $\sup_{Ax=a} b^\top x - \frac{1}{2}x^\top Qx$ (where the constraints could also be inequalities). The Lagrangian is the quadratic function

$$L(x,u) = \left(b - A^\top u\right)^\top x - \frac{1}{2}x^\top Qx + a^\top u.$$

Without entering tedious algebraic details, we just make some simple observations:

- If Q is not positive semidefinite, $\theta(u)$ is clearly $+\infty$ for all u. Lagrangian relaxation leads nowhere in this situation.

- On the other hand, let Q be positive definite. Then $L(\cdot, u)$ is maximized at the unique $x_u = Q^{-1}(b - A^\top u)$ and the dual function is obtained by plugging x_u into the Lagrangian:

$$\theta(u) = \frac{1}{2}\big(b - A^\top u\big)^\top Q^{-1}\big(b - A^\top u\big) + a^\top u.$$

A dual optimal u is characterized by $\nabla\theta(u) = AQ^{-1}(A^\top u - b) + a = 0$. Then the corresponding x_u satisfies $Ax_u = AQ^{-1}(b - A^\top u) = a$: in view of Sect. 1(jjj), this x_u is primal optimal.

In the present situation, solving the dual readily provides a primal optimal solution, the key being that x_u is unique (remember Remark 3(iv); and note that this property is not shared by linear programming).
- Between these two extremes, we have the case of $Q \succcurlyeq 0$ but not invertible—which includes linear programming $Q = 0$. To say that the equation $Qx = b - A^\top u$ has a solution (i.e. that $L(\cdot, u)$ has a finite maximum) is to say that u satisfies the constraints $b - A^\top u \in \operatorname{Im} Q$; the domain of θ is an affine subspace. For u dual-feasible, the possible x_u's also make up an affine subspace, parallel to $\operatorname{Ker} Q$. Needless to say, minimizing θ and computing the primal optimal solution(s) amounts to solving a linear system, which is just the optimality condition of the original quadratic problem.

2.5 Homogeneous quadratic constraints; max-cut

Now take $m + 1$ symmetric matrices and dualize the problem

$$\sup \frac{1}{2}x^\top Q_0 x, \qquad \frac{1}{2}x^\top Q_j x = a_j, \quad j = 1, \dots, m; \tag{9}$$

we do not consider any linear terms in this example; this greatly simplifies the calculations. The Lagrangian is

$$L(x, u) = \frac{1}{2}x^\top Q(u)x + u^\top a \quad \text{with } Q(u) := Q_0 - \sum_{j=1}^m u_j Q_j.$$

Clearly we can have $\theta(u) < +\infty$ only when $Q(u)$ is negative semidefinite. In this case, the maximum of $L(\cdot, u)$ is attained for x arbitrary in $\operatorname{Ker} Q(u)$ (which may reduce to $\{0\}$). In a word, the dual problem of (9) is

$$\inf a^\top u, \qquad \sum_{j=1}^m u_j Q_j - Q_0 \succcurlyeq 0,$$

a so-called semi-definite programming problem (SDP).

An instance of (9) is to find a maximal cut in a graph, which we write[1]

$$\sup x^\top Q x, \quad x \in \mathbb{R}^n, \qquad x_i^2 = 1, \quad i = 1, \dots, n. \tag{10}$$

[1] See (Goemans 1997; Lemaréchal and Oustry 1999). In fact, partition the node set of a graph into two subsets (a cut); take $x_i = 1$ (resp. -1) if node i lies in the first (resp. second) subset. If $-Q_{ij}$ is the value of arc (i, j), the value of the partition is $\sum_{i<j} \frac{1}{2}(-Q_{ij})(1 - x_i x_j)$; maximizing this is to maximize $\sum_{ij} x_i Q_{ij} x_j$.

Calling $D(u)$ the diagonal matrix formed by the vector $u \in \mathbb{R}^n$, the Lagrangian is $L(x,u) = x^\top(Q - D(u))x + e^\top u$, where $e \in \mathbb{R}^n$ is the vector of all ones. The dual of (10) then comes readily:

$$\inf e^\top u, \quad u \in \mathbb{R}^n, \qquad D(u) - Q \succcurlyeq 0. \tag{11}$$

The next subsection will reveal an interesting outcome of this duality scheme.

2.6 Conic and SDP duality

The case of inequality constraints $c(x) \le 0$ of (5) is just a particular case of

$$\sup f(x), \quad x \in X, \quad c(x) \in K, \tag{12}$$

where K is a closed convex cone; in Sect. 2.1, $K = -\mathbb{R}^m_+$ (and in (1), $K = \{0\}$). Introducing again the slack variable $y = c(x) \in K$ and the slackened Lagrangian $f(x) - u^\top(c(x) - y)$, the dual function of (3) becomes

$$\sup\left\{f(x) - u^\top c(x) + u^\top y \colon x \in X, y \in K\right\} = \theta(u) + \sup_{y \in K} u^\top y.$$

It is easy to see that

$$\sup_{y \in K} u^\top y = \begin{cases} 0 & \text{if } u^\top y \le 0 \text{ for all } y \in K, \\ +\infty & \text{otherwise;} \end{cases}$$

this reveals the *dual feasible set* [those u such that $u^\top y \le 0$ for all $y \in K$], which is a familiar object in convex analysis: the *polar cone* K° of K. In summary, the dual problem is now

$$\inf_{u \in K^\circ} \theta(u).$$

Thus, we generalize Sect. 2.1 as follows: when (1) is replaced by (12), just take the same Lagrangian (2); this yields the same dual function (3) *but* introduces the dual constraint $u \in K^\circ$. With respect to Sect. 2.1, observe that $(-\mathbb{R}^m_+)^\circ = \mathbb{R}^m_+$ (and for (1), $\{0\}^\circ = \mathbb{R}^m$).

Most representative examples of such conic duality are SDP programs, which have the general form

$$\sup b^\top x, \quad x \in \mathbb{R}^n, \qquad Q_0 + \sum_{i=1}^n x_i Q_i \succcurlyeq 0;$$

here the Q_i's lie in the space $\mathcal{S}$ of $p \times p$ symmetric matrices. Just as (12), this problem does enter our general framework (1). Let us introduce a slack variable Y (or rather a slack matrix), formulate the problem as

$$\sup b^\top x, \quad x \in \mathbb{R}^n, \qquad Y \succcurlyeq 0, \quad Y = Q_0 + \sum_i x_i Q_i,$$

and reproduce the above reasoning:

- The environment space X is $\mathbb{R}^n \times \mathcal{S}_+$, where $\mathcal{S}_+$ is the positive semidefinite cone in $\mathcal{S}$.
- The constraint to be dualized is $c(x,Y) = Q_0 + \sum_i x_i Q_i - Y$.
- This constraint takes its values in $\mathcal{S}$, so we need an inner product in that space—remember Remark 1(iii). The most convenient one turns out to be $\langle U, Y \rangle := \sum_{i,j} U_{ij} Y_{ij}$,

- so that the slackened Lagrangian is $b^\top x - \langle U, Q_0 + \sum_i x_i Q_i - Y\rangle$.
- Its maximization w.r.t. x unconstrained imposes to cancel the coefficient of each x^i: U must satisfy $b_i - \langle Q_i, U\rangle = 0$.
- Its maximization w.r.t. $Y \succcurlyeq 0$ forces U in the polar of the cone of $\mathcal{S}_+$. A result of linear algebra says that this is the cone of negative semidefinite matrices: $\mathcal{S}_+^\circ = -\mathcal{S}_+$ (Horn and Johnson 1989, Corollary 7.5.4, for example).

Changing signs for the sake of elegance,[2] our dual problem is

$$\sup\langle Q_0, U\rangle, \quad U \succcurlyeq 0, \quad \langle Q_i, U\rangle + b_i = 0, \quad i = 1, \dots, n.$$

Let us apply this duality scheme to the dual (11) of maxcut (beware that the variable U is already a dual variable! in a way, we make up a bidual problem; so the notation X is natural for the (bi)dual variable). With appropriate selection of signs, we obtain

$$\sup\langle Q, X\rangle, \quad X \succcurlyeq 0, \quad X_{ii} = 1,\ i = 1, \dots, n.$$

We recognize the SDP relaxation of Goemans and Williamson (1995). At this point we mention that the celebrated ϑ number of Lovász can likewise be obtained by a dualization *à la* Sect. 2.5, followed by the present SDP dualization. Known SDP bounds in combinatorial optimization (max cut, ϑ) seem to have been derived via Lagrangian relaxation already in the 80's, by N.Z. Shor: see for example (Stetsenko and Shor 1984). Such derivations have been made systematic since then, in (Alizadeh 1995; Poljak et al. 1995; Vandenberghe and Boyd 1996) among others. The technique is probably best explained in (Lemaréchal and Oustry 1999).

2.7 A decomposable problem: unit-commitment

In decomposable problems, one has to optimize a sum of functions, over a Cartesian product of sets, with constraints that are also of the sum type. An illustrative example is to optimize the daily production of electrical power plants $1, \dots, I$, over a time period $1, \dots, T$; see (Fisher 1973; Muckstadt and Koenig 1977; Bertsekas et al. 1983) among others; (Lemaréchal and Renaud 2001) takes a point of view similar to the present paper and contains recent bibliographical references.

The problem is to compute a production schedule $x^i = (x_1^i, \dots, x_T^i)$ for each plant i. Each x^i is to be taken in a set X^i of feasible schedules (the technological constraints defining X^i are fairly complicated; and different plants—thermal, hydraulic, etc.—have of course completely different feasible schedules). Besides, each feasible schedule x^i induces a cost $c^i(x^i)$. Now, call $p_t^i(x^i)$ the power delivered by schedule x^i at time t; the total power thus delivered must meet at time t the total demand d_t (assumed known; remember that electricity can hardly be stored). The problem is therefore

$$\begin{cases} \inf \sum_{i=1}^I c^i(x^i), & \\ x^i \in X^i & \text{for } i = 1, \dots, I, \\ \sum_{i=1}^I p_t^i(x^i) \geq d_t & \text{for } t = 1, \dots, T. \end{cases} \tag{13}$$

[2] Note that a $+$ sign can be used instead of $-$ in (2); or u can be changed to $-u$; or (1) could be a minimization problem, etc. While these operations imply some mental agility, none of them changes the essence of the dual problem.

Clearly, the demand constraints link the plants together. Were these constraints not present, each plant could be optimized separately. As a result, form the Lagrangian

$$L(x,u) := \sum_{i=1}^{I} c^i(x^i) - \sum_{t=1}^{T} u_t \left(\sum_{i=1}^{I} p_t^i(x^i) - d_t \right)$$

(remember footnote,[2] we leave it as an exercise to realize that the u_t's must be nonnegative!). Its minimization with respect to x reduces to I independent optimization problems

$$\inf_{x^i \in X^i} c^i(x^i) - \sum_{t=1}^{T} u_t p_t^i(x^i).$$

3 Minimal convex analysis

Having thus introduced and motivated Lagrangian relaxation, we now turn to some theory: how the dual problem can be solved, and what it is good for, in terms of the primal.

3.1 The dual problem

The following result is the basis for solving the dual problem:

Theorem 8 *The dual function θ is convex. If x_u maximizes $L(\cdot, u)$, then $g_u := -c(x_u)$ is a subgradient of θ at u:*

$$g_u \in \partial\theta(u), \quad \textit{i.e.} \quad \theta(v) \geq \theta(u) + g_u^\top (v-u) \quad \textit{for all } v \in \mathbb{R}^m. \tag{14}$$

It is important to realize that this result holds without any assumption on the data (X, f, c) in (1). From a numerical point of view, it clearly shows the following:

Lagrangian relaxation, *and therefore column generation,* both amount to minimizing a convex function

with the help of the information delivered by the oracle of Fig. 1 (i.e. function- and subgradient-values).

Two consequences are worth mentioning for combinatorial optimization:

- Convex optimization is admittedly an "easy" problem. Complexity of the dual problem (4) therefore reduces to complexity of the Lagrangian problem (3). Via the ellipsoid algorithm, both are indeed [polynomially] equivalent; see (Grötschel et al. 1981).
- Lagrangian relaxation definitely belongs to the (nonlinear) *continuous* world. More precisely, its theoretical aspects (studied in the present Sect. 3) rely a lot on convex analysis, while numerical aspects form the quasi entirety of *nondifferentiable or nonsmooth* optimization, the subject of Sect. 4 below.

3.2 Primal–dual relations

The aim of the present section is to study the question: how good is it to solve the dual, in terms of the primal? Naturally, the ultimate answer lies in the duality gap and we are interested in whether this duality gap is zero.

The filling property Here comes a technical concept concerning a converse to (14). Indeed the vector g_u singled out in Theorem 8 is the partial derivative of L (with respect to u) at the point (x_u, u). There are (at least) as many such "derivatives" as possible maximizers of $L(\cdot, u)$; all of them contribute to the subdifferential of θ at u. It is then relevant to ask whether they *suffice* to describe this set:

Definition 9 (Filling property) Define the set made up of the image by $\nabla_u L = -c$ of all the possible maximizers of $L(\cdot, u)$:

$$G_u := \{g_u = -c(x_u)\colon\ L(x_u, u) = \theta(u)\}. \tag{15}$$

We say that the filling property holds at u when the subdifferential of θ at u is the convex hull of G_u: $\partial\theta(u) = \operatorname{conv} G_u$.

In words: if the oracle of Fig. 1 were able to output every possible x_u maximizing the Lagrangian, then any subgradient could be constructed by taking all possible convex combinations of the constraints at these x_u's.

Remark 10 Reasonable assumptions generally ensure the filling property; cases where it does not hold can be considered as pathological. In the applications of Sect. 2, for example, delicate situations occur only

- in column generation, when X of (7) is neither bounded nor polyhedral;
- in SDP programming of Sect. 2.6, because the cone of positive semidefinite matrices is not polyhedral;
- in a "general" problem such as in Sect. 2.7.

Indeed the filling property automatically holds in (1) whenever X is a compact set (for example finite), over which f and c are continuous functions. Establishing results of this sort is a fairly technical domain of convex analysis; see for example (Hiriart-Urruty and Lemaréchal 2001, Sect. D.4.4).

Implications of the filling property Now take a u such that the filling property does hold.

When $L(\cdot, u)$ is maximized at a *unique* $x_u \in X$, then $\partial\theta(u) = G_u = \{-c(x_u)\}$ is a singleton. This means that θ is *differentiable* at u, its gradient being $\nabla\theta(u) = -c(x_u)$. Nondifferentiability of θ occurs when x_u is not unique.

Let in particular u minimize θ. In the differentiable case, $\nabla\theta(u) = 0$, i.e. the corresponding (unambiguous) x_u from Fig. 1 has $c(x_u) = 0$; hence

$$\theta(u) = L(x_u, u) = f(x_u) - u^\top c(x_u) = f(x_u)$$

and x_u solves (1) in view of weak duality; remember (jjj) in Sect. 1. This observation is important: with no assumption on the data (X, f, c), [filling property and] uniqueness of x_u from the oracle at a dual optimal u guarantees primal optimality of this x_u.

More generally, the mere characterization $0 \in \partial\theta(u) = \operatorname{conv} G_u$ of an optimal u gives directly a fundamental relation:

Theorem 11 *Let $\hat{u}$ minimize θ over $\mathbb{R}^m$ and assume the filling property at $\hat{u}$. Then there exist*

- *primal points $\{\tilde{x}_k\}$ maximizing $L(\cdot, \hat{u})$ over X, and*

– *corresponding convex multipliers* $\{\alpha_k\}$ ($\alpha_k \geq 0$, $\sum_k \alpha_k = 1$)

such that $\sum_k \alpha_k c(\tilde{x}_k) = 0$.

Again this result makes no assumption on the data in (1), other than the filling property. It blatantly reveals a *convexification* effect of dualization and explains what relevant properties (X, f, c) should enjoy for Lagrangian relaxation to work:

– If X is a convex set, we can define the primal point $\hat{x} := \sum_k \alpha_k \tilde{x}_k$; it lies in X.
– If, in addition, the constraints are affine: $c(x) = Ax - a$; then $c(\hat{x}) = c(\sum_k \alpha_k \tilde{x}_k) = \sum_k \alpha_k c(\tilde{x}_k) = 0$; hence $\hat{x}$ is feasible in (1).
– Finally, if also f is a concave function on X, then $L(\cdot, \hat{u})$ is concave as well, hence

$$L(\hat{x}, \hat{u}) \geq \sum_k \alpha_k L(\tilde{x}_k, \hat{u}) = \theta(\hat{u});$$

because we already know that $f(\hat{x}) = L(\hat{x}, \hat{u})$ ($c(\hat{x}) = 0$), weak duality tells us that $\hat{x}$ is primal optimal.

In a word,

$$\boxed{\begin{array}{c}\text{to prove optimality of a dual } \hat{u}\\ \textit{is} \text{ to construct a primal } \hat{x}\end{array}} \tag{16}$$

which does solve (1) if appropriate convexity properties hold: convexity of X, affinity of the dualized constraints, concavity of f. When they do not hold, $\hat{x}$ still solves a *relaxed* problem, obtained by convexifying (1); a result essentially due to (Geoffrion 1974; Magnanti et al. 1976):

Theorem 12 *Suppose* $c(x) = Ax - a$ *in* (1) *and let a dual optimum* $\hat{u}$ *satisfy the filling property. Then the above* $\hat{x}$ *maximizes, under the same constraints* $Ax = a$, *the function* f_X *obtained by taking the concave hull of* f *on the convex hull of* X.

When c is not affine, the convexified problem takes a definitely more complicated form; see (Feltenmark and Kiwiel 2000; Lemaréchal and Renaud 2001) for some indications along these lines.

3.3 The case of column generation

In view of its equivalence with Lagrangian relaxation, column generation *has to* solve the dual problem—even though its aim is primarily to solve the primal problem (7): (16) can be stated equivalently as

$$\boxed{\begin{array}{c}\text{primal optimality of a solution of the restricted master (8)}\\ \textit{is} \text{ dual optimality of the corresponding multiplier } u_K.\end{array}}$$

In fact, the column generation mechanism stops when the satellite produces a new column $\tilde{x}_{K+1}$ already lying in X_K. If $\hat{x}_K$ is an optimal solution of (8), this means that

$$\theta(u_K) = \max_{x \in X_K} L(x, u_K) = b^\top \hat{x}_K$$

where the second equality comes from standard LP theory. Therefore [$\hat{x}_K$ is an optimal solution of (7) and] u_K minimizes θ.

Besides, the definition of θ implies

$$\theta(u_K) \geq L(\tilde{x}_k, u_K) = b^\top \tilde{x}_k - u_K^\top (A\tilde{x}_k - a), \quad \text{for } k = 1, \dots, K.$$

Express $\hat{x}_K$ as $\sum_k \alpha_k \tilde{x}_k$ and suppose strict inequality holds for some k with $\alpha_k > 0$. By summation we obtain the contradiction

$$\theta(u_K) > \sum_{k=1}^{K} \alpha_k \big[b^\top \tilde{x}_k - u_K^\top (A\tilde{x}_k - a)\big] = b^\top \hat{x}_K.$$

Thus, $L(\tilde{x}_k, u_K) = \theta(u_K)$ for each such k: solving (8) appears as an attempt to computing the $(\tilde{x}_k, \alpha_k)$'s of Theorem 11.

Generalizing column generation from (7) to (1) is then a natural operation: the restricted master (8) then becomes

$$\sup \sum_{k=1}^{K} \alpha_k f(\tilde{x}_k), \qquad \sum_{k=1}^{K} \alpha_k c(\tilde{x}_k) = 0, \quad \sum_{k=1}^{K} \alpha_k = 1, \ \alpha_k \geq 0.$$

This is usually called generalized linear programming.

3.4 The case of conic constraints

Adapting the development in Sect. 3.2 to the more general situation of Sects. 2.1, 2.6 is an interesting and useful exercise.

Simple inequalities First consider the formulation (6). The set of partial derivatives of the slackened Lagrangian is here $G_u + N_u$, where G_u is defined in (15) and $N_u \subset \mathbb{R}^m$ is defined coordinatewise by:

$$-y \in N_u \quad \iff \quad y^j \begin{cases} = 0 & \text{if } u^j > 0, \\ \text{is arbitrary } \geq 0 & \text{if } u^j = 0, \end{cases} \quad \text{for } j = 1, \dots, m.$$

Take an optimal $\hat{u}$ satisfying the filling property: $\partial\theta(\hat{u}) = \operatorname{conv}(G_u + N_u)$; it is not difficult to check that $\operatorname{conv}(G_u + N_u) = (\operatorname{conv} G_u) + N_u$. Thus (Theorem 11), optimality of $\hat{u}$ is characterized by the existence of

- primal points $\tilde{x}_k$ maximizing $L(\cdot, \hat{u})$,
- corresponding convex multipliers α_k,
- and $\hat{y} \in -N_u$,

such that $0 = -\sum_k \alpha_k c(\tilde{x}_k) - \hat{y}$.

Then reproduce the reasoning of Sect. 3.2:

- If X is a convex set, the primal point $\hat{x} := \sum_k \alpha_k \tilde{x}_k$ lies in X.
- If, in addition, each c^j is a convex function over X, then

$$c^j(\hat{x}) \leq \sum_k \alpha_k c^j(\tilde{x}_k) = -\sum_k \alpha_k \hat{y}^j = -\hat{y}^j \leq 0,$$

  so $\hat{x}$ is feasible in (5).

– Finally, if f is a concave function on X, then $L(\cdot,\hat u)$ is concave as well (remember that $u \geq 0$), hence

$$L(\hat x,\hat u) \geq \sum_k \alpha_k L(\tilde x_k,\hat u) = \theta(\hat u).$$

– At this point, we invoke one more argument: from the properties of $\hat y$, either $\hat u^j = 0$ or $c^j(\hat x) = -\hat y^j = 0$, hence $\hat u^\top c(\hat x) = 0$ and $L(\hat x,\hat u) = f(\hat x)$. Weak duality tells us as before that $\hat x$ is optimal.

Naturally, the pair $(\hat x,\hat u)$ satisfies *complementary slackness*:

$$\hat u \in \mathbb{R}^m_+, \qquad c(\hat x) \in -\mathbb{R}^m_+, \qquad \hat u^\top c(\hat x) = 0.$$

General cones This development is put in perspective by the more intrinsic formulation (12), whose dual problem is to minimizes $\theta(u)$ for $u \in K^\circ$. Then invoke convex analysis (Hiriart-Urruty and Lemaréchal 1993, Chap. VII, for example): an optimal $\hat u$ is characterized by a subgradient $\hat g \in \partial\theta(\hat u)$ (possibly nonzero but) whose opposite lies in the so-called *normal* cone[3] to K° at $\hat u$. This normal cone is easily seen to be the set of $v \in K$ which are orthogonal to $\hat u$. Thus, admitting the filling property, we have at a dual optimum $\hat u$:

– a set of primal points $\tilde x_k$ maximizing $L(\cdot,\hat u)$,
– a set of corresponding convex multipliers α_k,
– and a normal $\hat v$ to K° at $\hat u$,

such that $\sum_k \alpha_k c(\tilde x_k) = \hat v$.

Once again:

– Assume convexity of X, so that $\hat x := \sum_k \alpha_k \tilde x_k$ lies in X.
– For simplicity, assume c is affine, so that $c(\hat x) = \sum_k \alpha_k c(\tilde x_k) = -\hat v$.
– Assume concavity of f—hence of $L(\cdot,\hat u)$—so that $L(\hat x,\hat u) \geq \theta(\hat u)$.
– Remembering that $c(\hat x) = \hat v$ is orthogonal to $\hat u$, we finally have $f(\hat x) = L(\hat x,\hat u) \geq \theta(\hat u)$; by virtue of weak duality, $\hat x$ is primal optimal.

Note the property

$$\hat u \in K^\circ, \qquad c(\hat x) \in K, \qquad \hat u^\top c(\hat x) = 0,$$

which generalizes complementary slackness, remembering that $K = -\mathbb{R}^m_+$ and $K^\circ = \mathbb{R}^m_+$ in the situation of (5). The above development can of course be particularized to the case where K is the SDP cone $\mathcal{S}_+$; then make use of a result of linear algebra: the polar cone to $\mathcal{S}_+$ is $-\mathcal{S}_+$.

4 Dual algorithms

The task of a dual algorithm is to minimize the convex function θ of (4), with the help of the oracle of Fig. 1. Theorem 8 makes it clear that θ is nonsmooth at points u such that several $c(x_u)$ could be answered by the oracle: a dual algorithm must be a *nonsmooth optimization*

[3]The normal cone to a set S at $\hat u \in S$ is the set of v such that $v^\top(u - \hat u) \leq 0$ for all $u \in S$. Note that, if S is a cone, its polar is its normal cone at 0.

algorithm. From this point of view, the only relevant information delivered by the oracle is the value of the dual function and the subgradient, which we will denote by g. The fact that θ and g depend on some primal variable is of little importance for a dual algorithm. Nevertheless, we will sometimes need the primal point computed by the black box; then it will be denoted by $\tilde{x}$.

Thus, a call to the oracle at some point $u_k \in \mathbb{R}^m$ yields a number $\theta(u_k)$ and an m-vector g_k. They have primal counterparts:

$$c(\tilde{x}_k) = -g_k, \qquad f(\tilde{x}_k) = \theta(u_k) + u_k^\top c(\tilde{x}_k) = \theta(u_k) - u_k^\top g_k. \tag{17}$$

4.1 Subgradient methods

One often (wrongly) takes "Lagrangian relaxation" and "subgradient method" as equivalent concepts. The simplest subgradient algorithm defines the sequence of iterates by the formula $u_{K+1} = u_K - t_K g_K$, where t_K is a stepsize "suitably chosen".

The main advantage of the subgradient method is to be extremely simple. Besides, it has given birth to the important *ellipsoid* method, and to the lesser known, though more efficient, r-algorithm; see (Shor 1985). In these subgradient methods, convergence of $\{u_K\}$ to a dual solution is proved (at least as a subsequence); as predicted by (16), recovering a primal relaxed solution as in Sect. 3.2 is therefore possible (Larsson et al. 1999; Anstreicher and Wolsey 1993). This, however, is not classical and the more recent *volume* algorithm (Barahona and Anbil 2000) was precisely motivated by this question; as shown in (Sagastizábal et al. 2002), it is close in spirit to the bundle approach of Sect. 4.3 below.

4.2 Method of Kelley, or Cheney–Goldstein

Every call to the oracle defines via (14) an affine function minorizing the dual function. This allows the construction, at the current iteration, of a polyhedral function $\hat{\theta}(u) := \sup_k [\theta(u_k) + g_k^\top (u - u_k)] \le \theta(u)$, which estimates the real θ from below.

The method then consists in minimizing $\hat{\theta}$. The next iterate therefore solves

$$\inf r, \quad r \ge \theta(u_k) + g_k^\top (u - u_k), \quad k = 1, \ldots, K, \tag{18}$$

a linear program with as many constraints as iterations already done.

This method is known to be *desperately slow*: see in (Hiriart-Urruty and Lemaréchal 1993, Sect. XV.1.1) a counterexample due to Nemirovski (1983), showing that the error after k iterations can be as bad as $\sqrt[m]{1/k}$. Yet, it is fairly interesting. Remember (17), consider the linear case ($f(x) = b^\top x$, $c(x) = Ax - a$) and dualize (18), calling α_k the multipliers (with respect to (1) or (7), we rather do a bidualization); direct calculations give

$$\sup b^\top x, \qquad Ax = a, \qquad x = \sum_k \alpha_k \tilde{x}_k, \quad \alpha_k \ge 0, \ \sum_k \alpha_k = 1. \tag{19}$$

One recognizes the restricted master program (8), as written in Remark 6.

Two conclusions entail: (i) column generation and subgradient just solve identical problems—but via different algorithms; (ii) Dantzig–Wolfe is nothing more than a desperately slow algorithm for convex optimization.

4.3 Bundle methods

Two approaches are known to improve the previous method. One—ACCPM of (Goffin et al. 1992)—adopts a philosophy of interior points. The other—bundle method, going back to (Lemaréchal 1974; Wolfe 1975)—can be considered as improving the subgradient method as well; it works basically as follows.

Call $\hat{u}$ the iterate among the u_k's that yielded the best value of the dual function. Then add to the lower estimate $\hat{\theta}$ a quadratic term to penalize the deviation from $\hat{u}$; (18) becomes

$$\inf r + \frac{1}{2}|u - \hat{u}|^2, \qquad r \geq \theta(u_k) + g_k^\top(u - u_k), \quad k = 1, \ldots, K,$$

which has a unique solution u_+. Before performing the next iteration, we just update $\hat{u}$ to u_+ if $\theta(u_+) < \theta(\hat{u})$.

Just as with (18), it is interesting to dualize the above quadratic program. Assume again linear f and c for simplicity; the calculations are made as in Sect. 2.4 and produce the modified (19)

$$\begin{cases} \sup \left(b^\top x - \hat{u}^\top(Ax - a) - \frac{1}{2}|Ax - a|^2\right), \\ x = \sum_k \alpha_k \tilde{x}_k, \quad \alpha_k \geq 0, \ \sum_k \alpha_k = 1. \end{cases} \tag{20}$$

This allows an interpretation of bundle methods in the context of column generation: we replace the linear restricted master (19) by a quadratic one, based on *augmented Lagrangian*—a classical technique in nonlinear programming, see (Bertsekas 1995, Sect. 4.2). In fact, (20) amounts to maximizing over the X_K of (8) the augmented Lagrangian $x \mapsto L(x, \hat{u}) - \frac{1}{2}|Ax - a|^2$. The solution $\hat{x}$ of (20) is not feasible; it is only asymptotically that $A\hat{x} \to a$, alongside with the convergence of $\hat{u}$ to a dual solution.

Let us mention here that our description is oversimplified: $\hat{u}$ is not exactly the best dual point u_k and the quadratic stabilizing term is tuned via a coefficient $t_K > 0$. Besides, a mechanism can be inserted to clean the *bundle* $\{\theta(u_k), g_k\}_{k=1,\ldots,K}$ and keep its size reasonable.

Numerical illustrations We illustrate the bundle method—more precisely the implementation of (Lemaréchal and Sagastizábal 1997) (with the quadratic solver of Kiwiel 1986) on the Held–Karp relaxation (Held and Karp 1971) of traveling salesman problems. Table 1 gives some results with a few problems from `tsplib`.

- The number m of dual variables is given by the problem's name.
- The second column is the optimal dual value.
- Columns "2, 3, 4" (resp. ∞) give the number of iterations taken to obtain 2, 3, 4 exact digits on θ (resp. the optimal value) (1 iteration = 1 resolution of (20) + 1 computation of θ).

Table 1 Held–Karp relaxation of TSP solved by bundle

Problem	$\theta(\hat{u})$	2	3	4	∞	ℓ	CPU	% (20)	% θ
gr120	1606.3125	3	72	102	132	40	1 s	35	44
pcb442	50499.499	30	77	261	556	109	151 s	64	28
pcb1173	56350.993	22	93	236	502	65	438 s	31	64
pcb3038	136587.50	28	146	782	4212	324	10 h	30	56
fnl4461	181569.21	21	120	840	6965	470	30 h?	8?	87?

- Column ℓ gives the number of $\tilde{x}_k$'s used to form $\hat{x}$—the number of nonzero α_k's in the last resolution of (20). This ℓ is interesting for the following reason. At an optimal $\hat{u}$, $m+1$ subgradients $g_k = -A\tilde{x}_k + a$ are *a priori* needed from the oracle to make up the 0 vector in $\partial\theta(\hat{u})$ (Carathéodory's theorem); but $\ell \ll m+1$ are actually used. Alternatively, the linear master (19) (formulated in terms of the α's only) has *a priori* $m+1$ positive basic variables; but there exists an optimal basis with only $\ell \ll m+1$ of them: the linear master is highly degenerate. We see that the quadratic version (20) is not perturbed by this degeneracy, which even becomes an advantage: the necessary number of iterations for a bundle method is driven by ℓ, rather than m.
- The last three columns give an idea of the computing time, and of the percentages of this time spent respectively in the quadratic master (20) and in the oracle computing θ. These informations, obtained by `gprof`, are moderately reliable, though; in particular for the last problem fnl4461, which took nearly a week of computation, rather than 30 hours.

The tests were performed on a Sun Ultra 1 (167 MHz, 768 Mb).

Additional results Another application of Lagrangian relaxation is multicommodity flow routing; see (Ouorou et al. 2000) for a review, and some recent developments in (Babonneau and Vial 2007; Lemaréchal et al. 2006). For other applications in combinatorial optimization, (Briant et al. 2005) contains comparative results between volume, Kelley and bundle on various problems: cutting stock, vertex coloring, capacitated vehicle routing, multiple-item lot sizing.

In some applications, maximizing $L(\cdot, u)$ can be NP-hard; a dual solver accepting inaccurate oracles is then extremely useful. Because θ is the result of a maximization process, the oracle will probably be able to compute underestimates $\theta_u \le \theta(u)$ for given u. Subgradient algorithms versions exist that cope with this situation: see (Nurminskii and Zhelikhovskii 1977). It is well-known that Kelley is not affected by inaccuracies. As for bundle, the recent work (Kiwiel 2006) has proposed an ingenious way to cope with inaccurate oracles; efficiency of the resulting algorithm is assessed in (Kiwiel 2004; Kiwiel and Lemaréchal 2006) on cutting-stock problems.

Conclusion We hope to have convinced the reader that Lagrangian relaxation is encountered again and again in optimization, and is a lot more than mere subgradient algorithms. It inevitably calls for convex analysis and nonsmooth optimization, two subjects which are unfortunately rather technical. First steps into them could be facilitated by (Grinold 1970; Lasdon 1970, Appendix 8).

References

Alizadeh, F. (1995). Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, *5*(1), 13–51.

Anstreicher, K., & Wolsey, L. A. (1993, in press). On dual solutions in subgradient optimization, Unpublished manuscript, CORE, Louvain-la-Neuve, Belgium. *Mathematical Programming*.

Babonneau, F., & Vial, J. P. (2007, in press). Vial, Accpm with a nonlinear constraint and an active set strategy to solve nonlinear multicommodity flow problems. *Mathematical Programming*.

Barahona, F., & Anbil, R. (2000). The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, *87*(3), 385–399.

Bertsekas, D. P. (1995). *Nonlinear programming*. Athena Scientific.

Bertsekas, D. P., Lauer, G. S., Sandell, N. R., & Posberg, T. A. (1983). Optimal short-term scheduling of large-scale power systems. *IEEE Transactions on Automatic Control*, *28*, 1–11.

Briant, O., Lemaréchal, C., Meurdesoif, P., Michel, S., Perrot, N., & Vanderbeck, F. (2005, in press). Comparison of bundle and classical column generation. RR 5453, INRIA. http://www.inria.fr/rrrt/rr5453.html. *Mathematical Programming*.

Feltenmark, S., & Kiwiel, K. C. (2000). Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems. *SIAM Journal on Optimization*, *10*(3), 697–721.

Fisher, M. L. (1973). Optimal solution of scheduling problems using Lagrange multipliers: part I. *Operations Research*, *21*, 1114–1127.

Geoffrion, A. M. (1974). Lagrangian relaxation for integer programming. *Mathematical Programming Study*, *2*, 82–114.

Goemans, M. X. (1997). Semidefinite programming in combinatorial optimization. *Mathematical Programming*, *79*, 143–161.

Goemans, M. X., & Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, *6*, 1115–1145.

Goffin, J.-L., Haurie, A., & Vial, J.-P. (1992). Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, *38*(2), 284–302.

Grinold, R. C. (1970). Lagrangian subgradients. *Management Science*, *17*(3), 185–188.

Grötschel, M., Lovász, L., & Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, *1*, 169–197.

Held, M., & Karp, R. (1971). The traveling salesman problem and minimum spanning trees: part II. *Mathematical Programming*, *1*(1), 6–25.

Hiriart-Urruty, J.-B., & Lemaréchal, C. (1993). *Convex analysis and minimization algorithms*. Heidelberg: Springer.

Hiriart-Urruty, J.-B., & Lemaréchal, C. (2001). *Fundamentals of convex analysis*. Heidelberg: Springer.

Horn, R. A., & Johnson, C. R. (1989). *Matrix analysis*. Cambridge: Cambridge University Press (new edition, 1999).

Kiwiel, K. C. (1986). A method for solving certain quadratic programming problems arising in nonsmooth optimization. *IMA Journal of Numerical Analysis*, *6*, 137–152.

Kiwiel, K. C. (2004, in press). An inexact bundle approach to cutting stock problems. Technical report, Systems Research Institute, Warsaw. *INFORMS J. of Computing*.

Kiwiel, K. C. (2006). A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, *16*(4), 1007–1023.

Kiwiel, K. C., & Lemaréchal, C. (2006, submitted). An inexact conic bundle variant suited to column generation. Open archive http://hal.inria.fr/inria-00109402. *Mathematical Programming*.

Larsson, T., Patriksson, M., & Strömberg, A. B. (1999). Ergodic, primal convergence in dual subgradient schemes for convex programming. *Mathematical Programming*, *86*(2), 283–312.

Lasdon, L. (1970). *Optimization theory for large systems*. *Macmillan series in operations research*.

Lemaréchal, C. (1974). An algorithm for minimizing convex functions. In J. L. Rosenfeld (Ed.), *Information processing '74* (pp. 552–556). Amsterdam: North-Holland.

Lemaréchal, C. (2001). Lagrangian relaxation. In M. Jünger, D. Naddef (Eds.), *Computational combinatorial optimization* (pp. 112–156). Heidelberg: Springer.

Lemaréchal, C. (2003). The omnipresence of Lagrange. *4OR*, *1*(1), 7–25.

Lemaréchal, C., Ouorou, A., & Petrou, G. (2006). A bundle-type algorithm for routing in telecommunication data networks. Technical report RR6010, INRIA. https://hal.inria.fr/inria-00110559.

Lemaréchal, C., & Oustry, F. (1999). Semi-definite relaxations and Lagrangian duality with application to combinatorial optimization. Rapport de Recherche 3710, INRIA. http://www.inria.fr/rrrt/rr-3710.html.

Lemaréchal, C., & Renaud, A. (2001). A geometric study of duality gaps, with applications. *Mathematical Programming*, *90*(3), 399–427.

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

Magnanti, T. L., Shapiro, J. F., & Wagner, M. H. (1976). Generalized linear programming solves the dual. *Management Science*, *22*(11), 1195–1203.

Muckstadt, M. A., & Koenig, S. A. (1977). An application of Lagrangian relaxation to scheduling in power-generation systems. *Operations Research*, *25*, 387–403.

Nemirovskii, A. S., & Yudin, D. (1983). *Problem complexity and method efficiency in optimization*. *Wiley-Interscience Series in Discrete Mathematics*. New York: Wiley-Interscience (Original Russian: Moscow Nauka, 1979).

Nurminskii, E. A., & Zhelikhovskii, A. A. (1977). ε-Quasigradient method for solving nonsmooth extremal problems. *Cybernetics*, *13*(1), 109–114.

Ouorou, A., Mahey, P., & Vial, J.-P. (2000). A survey of algorithms for convex multicommodity flow problems. *Management Science*, *47*(1), 126–147.

Poljak, S., Rendl, F., & Wolkowicz, H. (1995). A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, *7*, 51–73.
Reeves, C. R. (1993). *Modern heuristic techniques for combinatorial problems*. New York: Blackwell Scientific.
Sagastizábal, C., Bahiense, L., & Maculan, N. (2002). The volume algorithm revisited: relation with bundle methods. *Mathematical Programming*, *94*(1), 41–69.
Shor, N. Z. (1985). *Minimization methods for non-differentiable functions*. Berlin: Springer.
Stetsenko, S. I., & Shor, N. Z. (1984). The connection between Lovász' estimates with dual estimates in quadratic Boolean problems. In *Solutions methods of nonlinear and discrete programming, proceedings of the seminar "National Council on Cybernetics"*. Kiev: Institute of Cybernetics.
Vandenberghe, L., & Boyd, S. (1996). Semidefinite programming. *SIAM Review*, *38*(1), 49–95.
Vanderbeck, F. (2000). On Dantzig–Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, *48*(1), 111–128.
Wolfe, P. (1975). A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Study*, *3*, 145–173.
Wolsey, L. A. (1998). *Integer programming*. New York: Wiley-Interscience.