Nonlinear Optimization and Applications 2, pp. 381–399
G. Di Pillo and F. Giannessi, Editors
Kluwer Academic Publishers B.V., 1999

Extended Nonlinear Programming

R. T. Rockafellar (rtr@math.washington.edu)
Dept. of Mathematics, Box 354350
University of Washington, Seattle, WA 98195–4350

Abstract

Shortcomings of the conventional problem format in nonlinear programming suggest the need for a broader model that features composite terms. Such a model, promoting better representation of the problem structures found in applications, can be adopted with no real sacrifice in computing practicality.

Keywords: Nonlinear programming, extended linear-quadratic programming, composite optimization, optimality conditions, quadratic approximations

1 Introduction

The basic problem in nonlinear programming, and for that matter in all of finite-dimensional optimization, is usually explained as having the form

$$(\mathcal{P}_0) \qquad \text{minimize } f_0(x) \text{ subject to } f_i(x) = \begin{cases} \leq 0 & \text{for } i = 1, \ldots, s, \\ = 0 & \text{for } i = s+1, \ldots, m. \end{cases}$$

Sometimes a geometric constraint like $x \in X$ is included, where X is an orthant or more generally a box, i.e., a product of closed intervals (whether bounded or unbounded). For purposes such as the introduction of Lagrange multipliers, however, X is suppressed in favor of additional equalities or inequalities.

This description is by now so ingrained that many people accept it without question. It is thought to provide a good statement of what optimization is about, and to offer the right model for setting up applications. In that role, it has become the language of communication between the optimization community and its clients. But is this really the best we can do?

All of optimization revolves around minimizing or maximizing some function over some feasible set. The issue is what framework is most advantageous for specifying that function and set, in order to promote effective computation and analysis along with good modeling in applications. From that angle, an imbalance in $(\mathcal{P}_0)$ is apparent. The format demands thinking about feasibility in a structured way but offers no suggestions for representing structure also in the objective. A user is just expected to produce a finite family of functions and to designate which give the objective, the equations, and the inequalities.

History has a lot to do with this. The prototype in modern optimization was linear programming. In a linear programming problem, the constraint and objective functions are affine, and there is little more to be said. Problem structure in linear programming has typically been viewed in terms of the sparsity patterns of a coefficient matrix. Nonlinear programming started out as a sort of negation of linear programming, with functions merely no longer affine. In the early days no need was felt to be more detailed than that, and no other guidelines were developed.

People took for granted that, for practicality, all the functions in a problem should be smooth (i.e., continuously differentiable) to whatever degree might be helpful. This was a much more serious restriction than may have been realized, though, especially in applying the same standard to the objective as to the constraints. Certain kinds of nonsmoothness are common in expressing objective structure, and nowadays it's known how to handle them quite simply through composite terms which allow the numerical utilization of *underlying* smoothness in the problem data to go forward.

Even in a context of smoothness, it's easy to come up with applications where composite structure in the objective function provides special support in computation. In nonlinear least-squares problems, the expression to be minimized has the form

$$f_0(x) = \lambda_1 f_{01}(x)^2 + \cdots + \lambda_r f_{0r}(x)^2 \tag{1}$$

with positive weights λ_j. This function f_0 comes from composing a smooth mapping $x \mapsto (f_{01}(x), \dots, f_{0r}(x))$ with a convex function $(u_{01}, \dots, u_{0r}) \mapsto \lambda_1 u_{01}^2 + \cdots + \lambda_r u_{0r}^2$. The numerical methodology revolves around linearizations of the mapping.

For nonsmooth objective structures, a smooth mapping is often in the background as well, but the conventional format in $(\mathcal{P}_0)$ doesn't permit it to be accessed directly. *Nonsmoothness of functions has to be recast as nonsmoothness of sets by way of inequality constraints.* A typical example is "min-max" optimization, where the goal is to minimize an expression like

$$f_0(x) = \max\{f_{01}(x), \dots, f_{0r}(x)\}. \tag{2}$$

If f_0 has this form in $(\mathcal{P}_0)$, an additional variable x_0 has to be introduced. The task is translated into the minimization of $\tilde{f}_0(x_0, x) = x_0$ subject to $0 \geq \tilde{f}_{0j}(x_0, x) = f_{0j}(x) - x_0$ for $j = 1, \dots, r$ and the other constraints already in $(\mathcal{P}_0)$. Note that f_0 in (2) is obtained by composing a smooth mapping $x \mapsto (f_{01}(x), \dots, f_{0r}(x))$ with the convex, piecewise linear function $(u_{01}, \dots, u_{0r}) \mapsto \max\{u_{01}, \dots, u_{0r}\}$.

A similar maneuver would be triggered by a seemingly innocent shift of squares to absolute values in (1). If the expression to be minimized in $(\mathcal{P}_0)$ is

$$f_0(x) = \lambda_1 |f_{01}(x)| + \cdots + \lambda_r |f_{0r}(x)|, \tag{3}$$

one is obliged to introduce extra variables x_{0j} and minimize $\tilde{f}_0(x_{01}, \ldots, x_{0r}, x) := \lambda_1 x_{01} + \cdots + \lambda_r x_{0r}$ subject to the given constraints on x and the additional conditions $f_{0j}(x) - x_{0j} \leq 0$ and $-f_{0j}(x) - x_{0j} \leq 0$ for $j = 1, \ldots, r$. But why should such a drastic reformulation be required for the f_0 in (3), where none was needed for the f_0 in (1)? The two objective functions have almost the same form, except that in (3) the smooth mapping $x \mapsto (f_{01}(x), \ldots, f_{0r}(x))$ is composed with a convex function $(u_{01}, \ldots, u_{0r}) \mapsto \lambda_1 |u_{01}| + \cdots \lambda_r |u_{0r}|$ that's piecewise linear.

Penalty expressions offer further illustration. What if we wish to modify $(\mathcal{P}_0)$ by replacing the constraint $f_1(x) \leq 0$ by a objective term, say a "linear" penalty term,

$$f_1(x) \leq 0 \qquad \longrightarrow \qquad \rho_1 \max\{0, f_1(x)\} \tag{4}$$

with penalty parameter $\rho_1 > 0$? Again, this doesn't fit the conventional format directly. One has to resort to introducing a new variable ξ_1 and passing to the problem of minimizing $\tilde{f}_0(x, \xi_1) = f_0(x) + \rho_1 \xi_1$ subject to $f_1(x) - \xi_1 \leq 0$, $-\xi_1 \leq 0$, and the remaining constraints in $(\mathcal{P}_0)$ for $i = 2, \ldots, m$. Objective structure has once more been converted to constraint structure in order to conform to the model that people have gotten accustomed to taking as standard. But the penalty expression in (4) is simply the composition of a smooth function with a convex, piecewise linear function, so one might hope for an approach that's less disruptive.

Penalties need not be just "linear," of course. Expressions with nonlinear pieces often come up, and the nonsmoothness may then be just of second order. For instance, an equation $f_i(x) = 0$ in $(\mathcal{P}_0)$ could be replaced by an objective term $\theta_i(f_i(x))$ in which θ_i is a convex penalty function that's quadratic on an interval around 0 but switches to affine expressions outside of that interval—without any jumps in the first derivative, only in the second derivative. Such composite terms arise in particular in "robust statistical" modifications of least-squares problems which aim at reducing the influence of outliers. Observe, though, that in this situation it's not evident any more how to squeeze the problem into the mold of $(\mathcal{P}_0)$ with functions that are all $\mathcal{C}^2$, if the given f_i's are all $\mathcal{C}^2$.

These difficulties aren't only technical. They have important consequences for the mathematical modeling that goes into optimization. People are turned away from problem formulations that might be very good at representing the right structure and are forced instead into something less apt and perhaps inappropriate. This is an issue that deserves more attention from the optimization community than it has so far received.

The distorting effect of a narrow format is most serious perhaps in the emphasis of $(\mathcal{P}_0)$ on constraints that have to be satisfied with no room for penalty or reward.

In practice, the constraints encountered in optimization models can be divided into two categories, "hard" and "soft." In the "hard" category are constraints that record immutable requirements on the variables involved, such as nonnegativity and defining relationships. In the "soft" category are constraints that express properties deemed desirable, but which might be subject eventually to trade-off with other properties, or with costs. There are parameters in such constraints that are assigned wishful values, which might later be altered if the results of optimization aren't satisfactory. In many circumstances it might be better to replace soft constraints by objective terms that directly model a trade-off or gray areas between having an inequality simply be fulfilled or not fulfilled. The statement of $(\mathcal{P}_0)$ discourages that, however.

Stochastic programming gives a very telling example. Beginners who have grown up with $(\mathcal{P}_0)$ tend to set up a stochastic programming model with exact constraints, not realizing how costly that can be. Because of uncertainty, a constraint that's supposed to be satisfied at a future stage, but is influenced by present decisions, may be handled much better by introducing penalties for its violation than by insisting that earlier actions be foolproof.

By now, it's well understood by theorists that a modeling distinction between objective and constraints is artificial. While one can take the position that any optimization problem comes down to minimizing some function over some set, one can equally well take the position that any optimization problem in n real variables consists, in principle, just of minimizing some function f over $\mathbb{R}^n$—if f is allowed to take on ∞. There's no real difference between including the condition $f_1(x) \leq 0$ in $(\mathcal{P}_0)$ as an *explicit* constraint or representing it *implicitly* in the objective by a penalty term $\theta_1(f_1(x))$ in which $\theta_1(u_1) = 0$ for $u_1 \leq 0$ but $\theta_1(u_1) = \infty$ for $u_1 > 0$.

To promote good modeling, we ought to explain optimization to the world in a framework in which transitions between such infinite penalties and alternative finite penalties are easy and attractive. Of course, the framework should also be one that's just as conducive, or even more conducive, to capturing and utilizing smoothness as the conventional one. And, as long as a list is being made, we might ask for the framework to be better than $(\mathcal{P}_0)$ at making use of what's now known about dualization and Lagrange multipliers. For instance, it should be able to handle a box constraint $x \in X$ without conversion to linear inequality constraints. It should furnish a way around the current obstacles to dualizing quadratic programming problems.

All this is possible with surprisingly little adjustment. By accepting the notion of composite terms as a basic modeling tool, one can arrive at a problem format that's much more flexible than $(\mathcal{P}_0)$ and much richer in ways of indicating structure, yet with essentially no sacrifice in computational practicality. Through only a small investment in learning how some common composite terms are dualized, in order to associate them with Lagrange multipliers, one can achieve a new level of standardization in which problems can be inputted to software without first reformulating them. An automatic interface can be supplied for invoking numerical packages as they now exist, but the door is opened also to further algorithmic developments, designed to take full advantage of the special structures that might be represented.

2 Format with Composite Modeling

As a first idea of the kind of extension that might be made—but one which we will end up simplifying—it could be proposed to replace $(\mathcal{P}_0)$ by

$$(\mathcal{P}_1) \qquad \text{minimize } \theta_0(F_0(x)) + \theta_1(F_1(x)) + \cdots + \theta_q(F_q(x)) \text{ over } x \in X,$$

where X is a subset of $I\!R^n$ and each composite term involves a smooth mapping $F_l : I\!R^n \to I\!R^{d_l}$ and a function θ_l on $I\!R^{d_l}$ that *might take on* ∞. The effective domains $D_l := \{u_l \in I\!R^{d_l} \mid \theta_l(u_l) < \infty$ would give constraints that are implicit in $(\mathcal{P}_1)$: a point x is feasible if and only if $x \in X$ and $F_l(x) \in D_l$ for $l = 1, \ldots, q$. As a special case, a particular θ_l might be just the *indicator* of D_l, i.e., the function that vanishes on D_l but has the value ∞ everywhere outside of D_l.

Much could be done with this fully composite model, but it seems too great a leap into inscrutability to be sold easily to users of optimization. It's too far from the conventional model for instant comparisons, and it appears to insist on too many things being specified.

What we propose instead, therefore, is a focus on the following problem format, which we speak of as *extended nonlinear programming*, or ENLP:

$$(\mathcal{P}) \qquad \text{minimize } f_0(x) + \theta(f_1(x), \ldots, f_m(x)) \text{ over } x \in X.$$

The feasible solutions to $(\mathcal{P})$ are the vectors $x \in X$ with $(f_1(x), \ldots, f_m(x)) \in D$, where

$$D = \{u \in I\!R^m \mid \theta(u) < \infty\}. \tag{5}$$

This covers the conventional NLP format in $(\mathcal{P}_0)$ as the case of $X = I\!R^n$ and

$$\theta(f_1(x), \ldots, f_m(x)) = \theta_1(f_1(x)) + \cdots + \theta_m(f_m(x)), \tag{6}$$

with the functions $\theta_1, \ldots, \theta_s$ being the indicator of $(-\infty, 0]$ but $\theta_{s+1}, \ldots, \theta_m$ being the indicator of $\{0\}$. It also has the virtue of indicating at once how exact constraints can be replaced by expressions imposing penalties or rewards, since all that's needed is the replacement of indicator functions in (6) by other functions θ_i.

In fact, $(\mathcal{P})$ also encompasses the general composite model $(\mathcal{P}_1)$ and thus all the examples that we've been discussing. This is because the functions $f_0, f_1, \ldots, f_m$ don't *have* to be identified one-by-one with the functions having the same symbol in $(\mathcal{P}_0)$, if that's not convenient. In $(\mathcal{P})$, we could choose to take $f_0 \equiv 0$, and then the entire objective is a composite expression $\theta(f_1(x), \ldots, f_m(x))$. It's a small step from that to specializing into a sum of composite terms as in $(\mathcal{P}_1)$.

For an illustration at an intermediate level, suppose the aim is to minimize a "max" expression like the one in (2) subject to equations and inequalities like those in $(\mathcal{P}_0)$. This amounts to minimizing

$$\theta_0(f_{01}(x),\ldots,f_{0r}(x)) + \theta_1(f_1(x)) + \cdots + \theta_m(f_m(x)) \tag{7}$$

over $X = I\!R^n$ with $\theta_0(u_{01},\ldots,u_{0r}) = \max\{u_{01},\ldots,u_{0r}\}$ and the other θ_i's indicators or $(-\infty,0]$ of $\{0\}$, as already seen. We then have an extended nonlinear programming problem of elementary type in which the f_0 in $(\mathcal{P})$ (as opposed to the f_0 in (2)) is just the constant function 0.

In like manner, nonlinear programming problems $(\mathcal{P}_0)$ with objective functions of the types in (1) or (3) can be handled as extended nonlinear programming problems with objective functions as in (7) but through different choices of θ_0.

We haven't yet been clear about the assumptions that should go along with the designation of $(\mathcal{P})$ as a problem of extended nonlinear programming. As in conventional nonlinear programming, we want the functions f_i to be smooth, of course, even with continuous second derivatives if needed. That's no longer a real restriction, but rather a modeling choice. The philosophy is that $f_0, f_1, \ldots, f_m$ furnish all the smoothness we wish to build into the problem for the purpose of generating Taylor expansions and other classical approximations. Anything nonsmooth is to be captured through the specification of θ.

What should assumed, though, about θ and for that matter about the set X in $(\mathcal{P})$? Optimization theory is capable now of treating very general θ and X, but our goal here isn't in that direction. Instead, we want a workable compromise between generality and simplicity, moreover one that concentrates on a systematic and elementary way of specifying θ and X. That turns out to be much easier than might be expected, although the solution requires a bit of explaining.

Assumptions. *In an extended nonlinear programming problem $(\mathcal{P})$, it will be supposed that*

(A1) *the functions $f_0, f_1, \ldots, f_m$ are smooth,*

(A2) *the set X in $I\!R^n$ is nonempty and polyhedral (convex), and*

(A3) *the function θ on $I\!R^m$ is convex, proper and lower semicontinuous, furthermore representable in the form*

$$\theta(u) = \sup_{y\in Y}\{y{\cdot}u - k(y)\} \tag{8}$$

by means of a nonempty polyhedral set Y in $I\!R^m$ and a smooth function k that is convex on Y.

The smoothness of the functions f_i has already been addressed. The polyhedral convexity of X may seem unnecessarily limiting, but it's good enough for many purposes. It covers nonnegativity constraints, upper and lower bounds on variables, and also situations where one wishes to minimize over a linear subspace of $I\!R^n$, with $I\!R^n$ itself as a special case, since linear subspaces are polyhedral sets in particular. Also covered are cases of linear equations where certain variables have to add to 1, or where defining relationships are expressed between various quantities.

There's no controversy over how to specify a polyhedral set. Everyone knows how to do that in more than one way, as convenience dictates. Anything nonpolyhedral could of course be handled in $(\mathcal{P})$ by other means.

It's the assumption on θ that may seem mysterious. The mystery will dissipate with an appeal to elementary convex analysis.

Taking θ to be convex and having ∞ as a possible value isn't itself controversial. (An extended-real-valued function is *proper* if it doesn't take on $-\infty$ and isn't just the constant function ∞.) In all the examples brought out so far, the outer function in the composite term was convex. One can go a very long way with that. Lower semicontinuity of θ is equivalent to the epigraph of θ being closed and is a minor technical requirement. The rest of (A3) is what raises eyebrows.

An expression of θ as in assumption (A3) will be called a *dualizing representation*. It will soon be seen that, in all the examples we've encountered and a vast array of others, such an expression of θ is indeed available. Dualizing representations will play a big role with respect to Lagrange multipliers. There's much more to them than that, however.

Many functions θ of interest are only piecewise smooth and therefore difficult to describe directly, but a dualizing representation (8) furnishes an alternative, if indirect, description that's actually quite simple and easy to work with, all the more so once the implications of it are understood. In order to specify θ through a dualizing representation, all one has to specify is a polyhedral set Y and a smooth function k that's convex on Y. As it turns out, it would be enough most of the time to have k be quadratic, hence specified in terms of an $m \times m$ positive-semidefinite matrix (in some cases just the 0 matrix).

This is a crucial observation because it reveals that the structural features so essential to the nonsmooth functions θ one wants to use in modeling don't have to be an impediment in practice. *To specify* $(\mathcal{P})$*, all one has to do is specify the smooth functions* $f_0, f_1, \ldots, f_m$ *on* $I\!\!R^n$ *and* k *on* $I\!\!R^m$ *along with two polyhedral sets* $X \subset I\!\!R^n$ *and* $Y \subset I\!\!R^m$*.* That's no harder than specifying a conventional nonlinear programming problem $(\mathcal{P})$, but it does require an appreciation of how θ corresponds to Y and k. The results coming next address that concern, first in general terms and then through examples.

Proposition 1. *If a function θ has a dualizing representation as in (A3), it automatically satisfies the requirements of being convex, proper and lower semicontinuous. Moreover Y and k, at least in its restriction to Y, can be recovered then from θ by*

$$Y = \{y \in I\!\!R^m \,|\, \theta^*(y) < \infty\}, \qquad\qquad k(y) = \theta^*(y) \;\text{ for }\; y \in Y,$$

where $\theta^*(y) = \sup_u \{u{\cdot}y - \theta(y)\}$. (This θ^* is the convex function conjugate to θ.)

Proof. Define $\psi(y)$ to equal $k(y)$ for $y \in Y$ but ∞ for $y \notin Y$. Then ψ is convex function on $I\!\!R^m$ that's proper and lower semicontinuous. According to (8), the conjugate convex function ψ^* is θ. It follows then that $\theta^* = \psi^{**} = \psi$, as is well known in convex analysis. □

On the basis of Proposition 1 there is a *one-to-one* correspondence between the functions θ admitted in assumption (A3) and the pairs (Y, k) described there, except that only the values of k on Y count. We show now how this correspondence can be broken down to a term by term description when θ is to some degree separable.

Proposition 2. *Suppose that $\theta(u) = \theta_1(u_1) + \cdots + \theta_q(u_q)$ for $u = (u_1, \ldots, u_q)$ with $u_l \in I\!R^{d_l}$. Let each θ_l have a dualizing representation in terms of a polyhedral set $Y_l \subset I\!R^{d_l}$ and a smooth function k_l that is convex on Y_l. Then θ has a dualizing representation with respect to $y = (y_1, \ldots, y_q)$ and $y_l \in I\!R^{d_l}$ in terms of*

$$Y = Y_1 \times \cdots \times Y_q, \qquad k(y) = k_1(y_1) + \cdots + k_q(y_q). \tag{9}$$

Proof. This is just of matter of recognizing that

$$\sum_{l=1}^{q} \sup_{y_l \in Y_l} \{u_l{\cdot}y_l - k_l(y_l)\} = \sup_{y \in Y} \{u{\cdot}y - k(y)\}$$

under (9). Note that the components u_l and y_l can be vectors or, in the case of dimension $d_l = 1$, merely scalars. □

Let's proceed now with some examples of dualizing representations, taking the cue from Proposition 2 that it's enough to consider individual terms. One-dimensional terms are a good place to begin.

We've seen that inequality constraints $f_i(x) \le 0$ correspond to objective terms $\theta_i(f(x))$ in which $\theta_i(u_i)$ is 0 when $u_i \le 0$ but ∞ when $u_i > 0$. The dualizing representation for such θ is obtained by taking $k_i \equiv 0$ on $Y_i = [0, \infty)$. For equality constraints $f_i(x) = 0$, we instead have $\theta_i(u_i)$ equal to 0 for $u_i = 0$ but ∞ everywhere else, and then $k_i \equiv 0$ on $Y_i = (-\infty, \infty)$. A linear penalty term as in (4) with $\theta_i(u_i) = \rho_i \max\{0, u_i\}$ comes out, however, as corresponding to $k_i \equiv 0$ on $Y_i = [0, \rho_i]$. Likewise, such a penalty term for an equality constraint, with $\theta_i = \rho_i |u_i|$, has $k_i \equiv 0$ on $Y_i = [-\rho_i, \rho_i]$.

It's valuable to observe that in these cases, where the function k_i doesn't really enter, the effect of replacing a classical constraint by a linear penalty term is to replace an unbounded interval Y_i by a truncated one. We'll see later that this corresponds to introducing bounds on Lagrange multipliers. In general, for $k_i \equiv 0$ on $Y_i = [\sigma_i, \rho_i]$ one gets $\theta_i(u_i) = \rho_i u_i$ for $u_i \ge 0$, and $\theta_i(u_i) = \sigma_i u_i$ for $u_i \le 0$, regardless of the signs of σ_i and ρ_i.

What can happen with $k_i \not\equiv 0$? A simple case is a term like those in the least-squares setting of (1), with $\theta_{0j}(u_{0j}) = \lambda_j u_{0j}^2$, $\lambda_j > 0$. The dualizing representation is obtained then with $k_{0j}(y_{0j}) = (1/4\lambda_j) y_{0j}^2$ on $Y_{0j} = (-\infty, \infty)$. If we kept the same function $k_{0j}(y_{0j})$ but truncated the interval to $Y_{0j} = [-\rho_j, \rho_j]$, however, we would get the piecewise linear-quadratic function θ_{0j} that has the formula $\lambda_j u_{0j}^2$ on the interval of u_{0j} values where the derivative of this term is between $-\rho_j$ and ρ_j, i.e., where $\alpha_j \leq u_{0j} \leq \beta_j$ with $\alpha_j = -\rho_j/2\lambda_j$ and $\beta_j = \rho_j/2\lambda_j$, but is extrapolated linearly outside that interval, with formula $\lambda_j\alpha_j^2 - \rho_j[u_{0j} - \alpha_j]$ to the left and $\lambda_j\beta_j^2 + \rho_j[u_{0j} - \beta_j]$ to the right. (These affine pieces take off as tangents from the original quadratic graph.) Note that this is just the kind of function of interest in "robust statistics." The expression for θ_{0j} is readily generalized to the case of $k_{0j}(y_{0j}) = (1/4\lambda_j)y_{0j}^2$ on an arbitrary interval $Y_{0j} = [\sigma_j, \rho_j] \subset (-\infty, \infty)$. Terms of such type come up in augmented Lagrangians.

Just with these very simple, one-dimensional dualizing representations, we have already taken care of all the composite terms mentioned in earlier examples except for the "max" term in (2). That requires an appeal to higher dimensions: we get

$$\begin{array}{l} \theta(u_{01}, \ldots, u_{0r}) = \max\{u_{01}, \ldots, u_{0r}) \;\text{ for }\; k \equiv 0 \;\text{ on} \\ Y = \{(y_{01}, \ldots, y_{0r}) \,|\, y_{0j} \geq 0,\;\; y_{01} + \cdots + y_{0r} = 1\}. \end{array} \tag{10}$$

The set Y is polyhedral, as stipulated in (A3). In another example, if Y is any polyhedral cone in $I\!R^m$ and $k(y) = \frac{1}{2}|y|^2$ for the Euclidean norm $|y|$, then one has $\theta(u) = \frac{1}{2}d(u, K)^2$ for the polar cone $K = Y^*$, with $d(u, K)$ denoting the distance of u from K.

All the examples so far fit the pattern of k being a purely quadratic convex function, perhaps identically 0. As seen from Proposition 2, if that holds for the individual terms in θ, whatever they might be, it also holds for θ as a whole. This case is particularly deserving of attention, and we give it special notation:

$$\theta = \theta_{YQ} \;\text{ when }\; \theta(u) = \sup_{y \in Y} \{u{\cdot}y - \tfrac{1}{2}y{\cdot}Qy\} \tag{11}$$

for a polyhedral set $Y \subset I\!R^m$ and a symmetric, positive-semidefinite matrix $Q \in I\!R^{m \times m}$ (possibly $Q = 0$). When Y is a box and Q is diagonal, one has a decomposition of θ_{YQ} into one-dimensional terms.

Note from Proposition 1 that although Y and the values of the quadratic form $y{\cdot}Qy$ for $y \in Y$ can be recovered from θ in (11), this wouldn't be enough to pin down Q uniquely unless Y has nonempty interior. Situations where Y has empty interior do arise, as for instance in (10).

Proposition 3. *Any function θ of the form θ_{YQ} in (11) is piecewise linear-quadratic, in the sense that its effective domain D in (5) is polyhedral (in particular closed and convex) and can be partitioned into a finite collection of polyhedral subsets, with respect to each of which the formula for θ is a polynomial of degree at most 2.*

Indeed, D is a polyhedral cone which is polar to $\{y \,|\, Qy = 0,\; Y + y \subset Y\}$. For D to be the whole space (so that θ is finite everywhere), it is necessary and sufficient therefore that the latter cone (likewise polyhedral) contain no $y \neq 0$.

Proof. For the first assertion, see Theorem 11.14(b) of [1]. The rest was proved in Proposition 2.4 of [2]. □

It follows for instance that when Y is bounded or Q is positive-definite, the feasible set in $(\mathcal{P})$ is the polyhedral set X. Otherwise the condition $(f_1(x), \ldots, f_m(x)) \in D$ could come into play in determining feasibility. Because D is a polyhedral cone, this condition can in theory be expressed by constraints $0 \geq \sum_{i=1}^m \alpha_{ki} f_i(x) = g_k(x)$ for a collection of coefficient vectors $(\alpha_{k1}, \ldots, \alpha_{km})$ (chosen to generate the cone polar to D in Proposition 3). This sheds light on the nature of the feasible set in an extended nonlinear programming problem $(\mathcal{P})$, revealing that it's neither more nor less general in principle than the kind of feasible set in a conventional problem $(\mathcal{P}_0)$—but that's also somewhat beside the point. The focus isn't merely on constraints any more, and because of the possibility of D having to be broken down into many pieces to get a direct description of θ, one may have to rely anyway on the dual description of θ furnished by Y and Q.

If we concentrate on functions θ of form θ_{YQ} in (11) and at the same time restrict f_0 to be quadratic and $f_1, \ldots, f_m$ to be affine in $(\mathcal{P})$ we get *extended linear-quadratic programming*, or ELQP for short:

$$(\mathcal{Q}) \qquad \text{minimize } \; c{\cdot}x + \tfrac{1}{2} x{\cdot}Px + \theta_{YQ}(b - Ax) \; \text{ over } \; x \in X,$$

where $A \in \mathbb{R}^{m \times n}$ and $P \in \mathbb{R}^{n \times n}$ (symmetric). This kind of model goes back to Rockafellar and Wets [3], where it was introduced for the sake of penalty modeling and algorithm development in stochastic programming. The topic was expanded in [2], where many special cases of ELQP were worked out and applications were made to continuous-time optimal control. Other aspects of ELQP methodology and applications can be found in [4], [5], [6], along with [1].

The general nature of $(\mathcal{Q})$, in contrast to conventional quadratic programming, can be perceived from the case where Y is a box and Q is a diagonal matrix, so that

$$\theta_{YQ}(b - Ax) = \theta_1(b_1 - a_1{\cdot}x) + \cdots + \theta_m(b_m - a_m{\cdot}x)$$

for the components b_i of b and the vectors a_i giving the rows of A. Our discussion of such one-dimensional terms shows that they can stand for piecewise linear-quadratic penalty expressions as well as standard linear constraints. Of course, "max" expressions can also be represented, by following the pattern in (10). When $P = 0$ and $Q = 0$ in $(\mathcal{Q})$, one has *extended linear programming*, ELP, where X can impose upper or lower bounds on the variables while θ allows for linear penalty.

An important advantage of ELQP over ordinary QP is that problem $(\mathcal{Q})$ can be dualized without difficulty to get another ELQP problem, as long as P is positive-semidefinite, so that convexity prevails in $(\mathcal{Q})$. Similarly, an ELP problem dualizes to an ELP problem. This will be explained near the end of the next section.

3 Extended Lagrangian and Multiplier Rule

The dualizing representation in assumption (A3) is the route to the *Lagrangian function* we associate with an extended nonlinear programming function $(\mathcal{P})$, namely

$$L(x,y) = f_0(x) + y_1 f_1(x) + \cdots + y_m f_m(x) - k(y) \text{ for } x \in X,\ y \in Y. \tag{12}$$

The sets X and Y are regarded as an integral part of the specification of L. Obviously $(\mathcal{P})$ is completely determined by its Lagrangian in this sense, because the expression being minimized over X in $(\mathcal{P})$ is

$$f(x) = \sup_{y \in Y} L(x,y) = f_0(x) + \theta(f_1(x) \ldots, f_m(x)). \tag{13}$$

The main difference between this extended Lagrangian for $(\mathcal{P})$ and the one associated with a conventional nonlinear programming problem $(\mathcal{P}_0)$ is the presence of the term $-k(y)$ (which could however be 0) and a general polyhedral set Y instead of just the special cone $I\!R^s_+ \times I\!R^{m-s}$ that classically expresses the sign requirements on Lagrange multipliers for constraints $f_i(x) \leq 0$ or $f_i(x) = 0$. Here, Y will express requirements on Lagrange multipliers more broadly.

There's also a difference in the presence of a set X, where for $(\mathcal{P}_0)$ one would have $X = I\!R^n$. This is less important, though, because a set X has often been brought into discussions of Lagrangians. If one goes to the original paper of Kuhn and Tucker [7], say, one finds a treatment of how full or partial nonnegativity of x can be represented in that way, and what it means for the statement of first-order optimality conditions. That simple modification has also been common in linear and quadratic programming.

Beyond such special instances of X, little has been made of how to adapt Lagrange multiplier rules to a constraint $x \in X$, at least within the optimization community at large, although theorists have long had answers. This is true even though the case of a box X is very common in numerical optimization. Actually the adaptation is quite easy, and it's worth looking at because the same ideas are need for understanding how to adapt to multiplier vectors $y \in Y$ for nonclassical Y.

The only notion that's needed is that of the *normal cone* $N_X(\bar{x})$ to a convex set X at one of its points $\bar{x}$, as introduced in convex analysis [8], [1]:

$$v \in N_X(\bar{x}) \iff \bar{x} \in X \text{ and } v \cdot [x - \bar{x}] \leq 0 \text{ for all } x \in X. \tag{13}$$

This is polar to the *tangent cone* $T_X(\bar{x})$, which for polyhedral X consists of the origin and all rays of the form $\{\tau[x - \bar{x}] \,|\, \tau \geq 0\}$ generated by points $x \neq \bar{x}$ in X. For a box X, the description of $N_X(\bar{x})$ is especially simple. Then X is a product of closed intervals X_j which constrain the components x_j of x, and we have

$$\begin{aligned} &(v_1, \ldots, v_n) \in N_X(\bar{x}_1, \ldots, \bar{x}_n) \text{ for } X = X_1 \times \cdots \times X_n \\ &\iff \begin{cases} v_j = 0 & \text{when } \bar{x}_j \text{ is an interior point of } X_j, \\ v_j \geq 0 & \text{when } \bar{x}_j \text{ is the right endpoint (only) of } X_j, \\ v_j \leq 0 & \text{when } \bar{x}_j \text{ is the left endpoint (only) of } X_j. \end{cases} \end{aligned} \tag{14}$$

When $\bar{x}_j$ is both the right and left endpoint of X_j, i.e., X_j is a one-point interval, there's no restriction on v_j; it can be any number in $(-\infty,\infty)$. All this holds also for Y with only a change of notation. In particular, when Y is a box we have

$$\begin{aligned}&(u_1,\ldots,u_m)\in N_Y(\bar{y}_1,\ldots,\bar{y}_m)\ \text{ for } Y=Y_1\times\cdots\times Y_m\\ \iff &\begin{cases}u_i=0 & \text{when } \bar{y}_i \text{ is an interior point of } Y_i,\\ u_i\geq 0 & \text{when } \bar{y}_i \text{ is the right endpoint (only) of } Y_i,\\ u_i\leq 0 & \text{when } \bar{y}_i \text{ is the left endpoint (only) of } Y_i.\end{cases}\end{aligned}\tag{15}$$

Theorem 1. *In an extended nonlinear programming problem $(\mathcal{P})$, let $\bar{x}$ be a locally optimal solution. Suppose that the following constraint qualification is satisfied:*

$$\nexists\, \bar{y}\in Y,\ \bar{y}\neq 0,\ \textit{ with } \begin{cases}-[\bar{y}_1\nabla f_1(\bar{x})+\cdots+\bar{y}_m\nabla f_m(\bar{x})]\in N_X(\bar{x}),\\ \quad(\bar{y}_1,\ldots,\bar{y}_m)\in N_D(f_1(\bar{x}),\ldots,f_m(\bar{x})).\end{cases}\tag{16}$$

Then necessarily

$$\exists\, \bar{y}\in Y\ \textit{ with } \begin{cases}-[f_0(\bar{x})+\bar{y}_1\nabla f_1(\bar{x})+\cdots+\bar{y}_m\nabla f_m(\bar{x})]\in N_X(\bar{x}),\\ (\bar{u}_1,\ldots,\bar{u}_m)\in N_Y(\bar{y}_1,\ldots,\bar{y}_m)\ \textit{ for } \bar{u}_i=f_i(\bar{x})-(\partial k/\partial y_i)(\bar{y}).\end{cases}\tag{17}$$

When $(\mathcal{P})$ is an extended linear-quadratic programming problem $(\mathcal{Q})$, the constraint qualification need not be invoked.

Proof. See Rockafellar [9] for ENLP and [2] for ELQP. □

The constraint qualification in (16) is equivalent to the Mangasarian-Fromovitz constraint qualification when $(\mathcal{P})$ is a conventional problem $(\mathcal{P}_0)$. It can be viewed therefore as an apt extension of that well known condition to our context. When $X=I\!R^n$, the first normality condition in (16) reduces to $\bar{y}_1\nabla f_1(\bar{x})+\cdots+\bar{y}_m\nabla f_m(\bar{x})=0$. On the other hand, $(\mathcal{P}_0)$ has D equal to the product of intervals $D_i=(-\infty,0]$ for $i=1,\ldots,s$ and $D_i=\{0\}$ for $i=s+1,\ldots,m$. In that case, by (14) as applied to D, the second normality condition in (16) requires $f_i(\bar{x})=0$ for inequality constraints with $\bar{y}_i>0$ but allows $f_i(\bar{x})\leq 0$ for inequality constraints with $\bar{y}_i=0$; for equality constraints, one must of course have $f_i(\bar{x})=0$. In other words, we get the usual conditions of *complementary slackness* associated with multipliers in $(\mathcal{P}_0)$.

The interpretation of the multiplier rule in (17) is similar. If $\bar{x}$ is an interior point of X, as for instance when $X=I\!R^n$, the cone $N_X(\bar{x})$ is just the zero cone $\{0\}$. In asking the gradient expression $-[\nabla f_0(\bar{x})+\bar{y}_1\nabla f_1(\bar{x})+\cdots+\bar{y}_m\nabla f_m(\bar{x})]$ to belong to $N_X(\bar{x})$, one is asking it to be 0. When $\bar{x}$ is a boundary point of X, however, the gradient expression is required to have a certain relationship to $\bar{x}$. What this might be can easily be seen for instance when X is a box. Then, according to (14), the requirement is for the jth component of the gradient sum to be positive, negative or zero according to the location of $\bar{x}_j$ within the jth interval X_j.

The meaning of the requirement $(f_1(\bar{x}),\ldots,f_m(\bar{x}))\in N_Y(\bar{y}_1,\ldots,\bar{y}_m)$ in (17) is simplest when Y is a box $Y_1\times\cdots\times Y_m$. In that case it places a sign restriction on $f_i(\bar{x})$ that's tied to the location of the multiplier $\bar{y}_i$ within Y_i, as seen from (15). Again we have a generalization of the complementary slackness conditions in a nonlinear programming problem ($\mathcal{P}_0$). When $Y_i = [0, \infty)$, the requirement comes out as $\bar{u}_i \leq 0$ if $\bar{y}_i = 0$ but $\bar{u}_i = 0$ if $\bar{y}_i > 0$. When $Y_i = (-\infty, \infty)$, it's just $\bar{u}_i = 0$. For general intervals Y_i, (15) describes a broader version of complementary slackness.

First-order optimality conditions for ($\mathcal{P}$) as in Theorem 1 were first developed in [9]. The second normality condition in (17) can equivalently be stated in terms of subgradients of θ, namely as

$$(\bar{y}_1, \ldots, \bar{y}_m) \in \partial\theta(f_1(\bar{x}), \ldots, f_m(\bar{x})). \tag{18}$$

When θ is separable as in (6), this comes down to $\bar{y}_i \in \partial\theta_i(f_i(\bar{x}))$ for $i = 1, \ldots, m$. An advantage of the version in (17), however, is a connection with variational inequalities.

Theorem 2. *The normality conditions in the multiplier rule (17) in Theorem 1 can be equivalently be expressed in the form*

$$-\nabla_x L(\bar{x}, \bar{y}) \in N_X(\bar{x}), \qquad \nabla_y L(\bar{x}, \bar{y}) \in N_Y(\bar{y}). \tag{19}$$

In terms of $z = (x, y)$, $Z = X \times Y$ and $F : (x, y) \mapsto (\nabla_x L(x, y), -\nabla_y L(x, y))$, this is the variational inequality $F(\bar{z}){\cdot}[z - \bar{z}] \geq 0$ for all $z \in Z$, the same as

$$F(\bar{z}) + N_Z(\bar{z}) \ni 0. \tag{20}$$

Proof. The equivalence of these various statements is immediate from the definition of L and the fact that $N_Z(\bar{z}) = N_X(\bar{x}) \times N_Y(\bar{y})$ for $\bar{z} = (\bar{x}, \bar{y})$. □

The set Z in the variational inequality of Theorem 2 is, like X and Y, polyhedral. It's a box when X and Y are boxes. For variational inequalities over polyhedral sets, unusually powerful results are available; see e.g. [10].

To explore convexity and duality in extended linear programming, we introduce now the notion of ($\mathcal{P}$) being a problem of *extended convex programming.* By that we'll mean that the data elements $f_0, f_1, \ldots, f_m$, X and Y in ($\mathcal{P}$) have the property that, for each $y \in Y$, $L(x, y)$ is convex in x relative to X.

Theorem 3. *Suppose ($\mathcal{P}$) is an extended convex programming problem in the general sense just defined. Then the expression $f_0(x) + \theta(f_1(x), \ldots, f_m(x))$ being minimized in ($\mathcal{P}$) is convex relative to X, and, the normal cone conditions in (17), or the equivalent versions of them in Theorem 2, are sufficient for $\bar{x}$ to be a globally optimal solution, without regard to the constraint qualification (16). Indeed, these conditions mean that the pair $(\bar{x}, \bar{y}) \in X \times Y$ gives a saddle point of L on $X \times Y$:*

$$L(x, \bar{y}) \geq L(\bar{x}, \bar{y}) \geq L(\bar{x}, y) \text{ for all } x \in X,\ y \in Y. \tag{21}$$

Furthermore, in this case the mapping F in the variational inequality in Theorem 2 is monotone on $Z = X \times Y$, i.e., one has

$$[F(z') - F(z)]{\cdot}[z' - z] \geq 0 \text{ for all } z, z' \in Z. \tag{22}$$

Proof. The convexity of the function being minimized over X follows from (13) and the assumed convexity of $L(x, y)$ in $x \in X$, since the supremum of any collection of convex functions is convex.

To say that $L(x, \bar{y}) \geq L(\bar{x}, \bar{y})$ for all $x \in X$ is to say that for every choice of $x \in X$ the function $\varphi(\tau) = L((1-\tau)\bar{x} + \tau x, \bar{y})$ has $\varphi(\tau) \geq \varphi(0)$ for $\tau \in [0, 1]$. The convexity of $L(\cdot, \bar{y})$ relative to X implies that φ is convex relative to $[0, 1]$, so that this inequality on φ holds if and only if $\varphi'(0) \geq 0$. But $\varphi'(0) = \nabla_x L(\bar{x}, \bar{y}) \cdot [x - \bar{x}]$. Hence we have $L(x, \bar{y}) \geq L(\bar{x}, \bar{y})$ for all $x \in X$ if and only if $\nabla_x L(\bar{x}, \bar{y}) \cdot [x - \bar{x}] \geq 0$ for all $x \in X$, or in other words, $-\nabla_x L(\bar{x}, \bar{y}) \in N_X(\bar{x})$.

We always have $L(\bar{x}, y)$ concave in $y \in Y$ on the basis of the formula for L in (12) and the convexity of k on Y that was assumed in (A3). By a parallel argument, therefore, we have $L(\bar{x}, y) \leq L(\bar{x}, \bar{y})$ for all $y \in Y$ if and only if $\nabla_y L(\bar{x}, \bar{y}) \in N_Y(\bar{y})$.

A minor extension of these two arguments brings out the fact that actually, for arbitrary choices of $x, x' \in X$ and $y, y' \in Y$, we have

$$\begin{aligned}
L(x', y) &\geq L(x, y) + \nabla_x L(x, y) \cdot [x' - x], \\
L(x, y') &\leq L(x, y) + \nabla_y L(x, y) \cdot [y' - y], \\
L(x, y') &\geq L(x', y') + \nabla_x L(x', y') \cdot [x - x'], \\
L(x', y) &\leq L(x', y') + \nabla_y L(x', y') \cdot [y - y'].
\end{aligned}$$

In multiplying the first and third inequalities by -1 and then adding all four together, we get $0 \geq [\nabla_x L(x, y) - \nabla x L(x', y')] \cdot [x' - x] - [\nabla_y L(x, y) - \nabla y L(x', y')] \cdot [y' - y]$, which comes out as (22) for $z = (x, y)$ and $z' = (x', y')$. □

In the light of Theorem 3, the vectors $\bar{y}$ paired with $\bar{x}$ in the first-order optimality conditions for an extended *convex* programming problem $(\mathcal{P})$ solve the associated *dual problem*

$$(\mathcal{P}^*) \qquad \text{maximize } g(y) \text{ over } y \in Y, \text{ where } g(y) = \inf_{x \in X} L(x, y).$$

It's especially interesting to see how this kind of duality works out in extended linear-quadratic programming.

Theorem 4. *In the case of an extended linear-quadratic programming problem* $(\mathcal{Q})$, *the Lagrangian is*

$$L(x, y) = c \cdot x + \tfrac{1}{2} x \cdot Px + b \cdot y - \tfrac{1}{2} y \cdot Qy - y \cdot Ax \quad \text{on} \quad X \times Y \tag{23}$$

and the first-order optimality conditions take the form

$$-c - P\bar{x} + A^T \bar{y} \in N_X(\bar{x}), \qquad b - A\bar{x} - Q\bar{y} \in N_Y(\bar{y}), \tag{24}$$

thus corresponding to a variational inequality (20) with F monotone and affine.

If P is positive-semidefinite along with Q, these conditions are equivalent to having $(\bar{x}, \bar{y})$ be a saddle point of L on $X \times Y$, and they hold if and only if $\bar{x}$ is optimal in $(\mathcal{Q})$

while $\bar{y}$ is optimal in the problem dual to ($\mathcal{Q}$). Moreover, that dual problem belongs again to extended linear-quadratic programming and has the form

$$(\mathcal{Q}^*) \qquad \text{maximize } b{\cdot}y - \tfrac{1}{2}y{\cdot}Qy - \theta_{XP}(A^T y - c) \text{ over } y \in Y.$$

Proof. In ($\mathcal{Q}$) we have $\theta = \theta_{YQ}$ and therefore the Lagrangian in (23). Everything then follows from Theorem 3 except for the particular form of ($\mathcal{Q}^*$). That emerges from the description of the general dual problem ($\mathcal{P}^*$) through the fact that

$$\begin{aligned}\inf_{x\in X}\{c{\cdot}x + \tfrac{1}{2}x{\cdot}Px + b{\cdot}y - \tfrac{1}{2}y{\cdot}Qy - y{\cdot}Ax\}\\ = b{\cdot}y - \tfrac{1}{2}y{\cdot}Qy - \sup_{x\in X}\{y{\cdot}Ax - c{\cdot}x - \tfrac{1}{2}x{\cdot}Px\}.\end{aligned}$$

Here $y{\cdot}Ax = x{\cdot}A^T y$, so the supremum is $\theta_{XP}(A^T y - c)$ by the definition of the function θ_{XP} (in parallel to that of θ_{YQ} in (11)). □

For the conclusions of Theorem 4 to be valid, it's not really essential that P and Q be positive semidefinite, but just that the expressions $x{\cdot}Px$ and $y{\cdot}Qy$ be convex with respect to $x \in X$ and $y \in Y$, respectively. In that more subtle case, though, the mapping F may only be monotone relative to $Z = X \times Y$ rather than the whole space $\mathbb{R}^n \times \mathbb{R}^m$. The same extra bit of generality is available also in the following duality theorem.

Theorem 5. *In the case of extended linear-quadratic programming with both the matrix P and the matrix Q positive-semidefinite, one has*

$$\min(\mathcal{Q}) = \max(Q^*) \ \textit{(with the existence of optimal solutions)}$$

in any of the following circumstances, which in fact are equivalent:
(a) *the optimal value in ($\mathcal{Q}$) is finite;*
(b) *the optimal value in ($\mathcal{Q}^*$) is finite;*
(c) *feasible solutions exist for both ($\mathcal{Q}$) and ($\mathcal{Q}^*$).*
This remains valid even if P and Q are not positive-semidefinite, as long as the expressions $x{\cdot}Px$ and $y{\cdot}Qy$ are convex with respect to $x \in X$ and $y \in Y$, respectively.

Proof. For positive-definite P and Q, this theorem was first proved in [3]. It has subsequently presented in that mode also in [2] and [1]. The following argument confirms that it holds true also under the weaker conditions of relative convexity.

Without loss of generality it can be supposed, for the purpose at hand, that $0 \in X$ that $0 \in Y$. This just amounts to a change of variables: for any choice of $x_0 \in X$ and $y_0 \in Y$, we can rewrite everything in terms of $x' \in X_0 = X - x_0$ and $y' \in Y_0 = Y - y_0$. The shifted Lagrangian $L_0(x', y') = L(x_0 + x', y_0 + y')$ on X_0 and Y_0 gives rise to primal and dual problems ($\mathcal{Q}_0$) and ($\mathcal{Q}_0^*$) in x' and y' that are equivalent to ($\mathcal{Q}$) and ($\mathcal{Q}^*$), as readily can be checked.

Once we have $0 \in X$ and $0 \in Y$, we know that the affine hulls of X and Y are subspaces of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Through coordinate transformations in these spaces, we can identify these affine hulls with spaces $I\!R^{n_0}$ and $I\!R^{m_0}$ of perhaps lower dimension. Since X and Y, being convex sets, have nonempty interior relative to their affine hulls, we can in this way reduce the whole issue to the case where X and Y have nonempty interior. In that case, however, the convexity of $x{\cdot}Px$ and $y{\cdot}Qy$ with respect to $x \in X$ and $y \in Y$ implies that P and Q are positive-semidefinite. The earlier result can then be invoked, and its conclusion can be brought over. □

4 Solution Methodology

Many of the approaches to solving nonlinear programming problems can be explained in terms of their Lagrangians and approximations that can be made of them. That's true for techniques of the sequential quadratic programming variety and even steepest descent, as well as for algorithms based on successive approximations to Karush-Kuhn-Tucker conditions and the techniques for handling variational inequalities. Because the extended Lagrangian in (12) is so close to the classical one, it's tempting to think that virtually all of those approaches may work also for extended nonlinear programming with appropriate adjustments in concept and formulation.

For example, around any pair of points $\hat{x} \in X$ and $\hat{y} \in Y$, the Lagrangian L for problem $(\mathcal{P})$ has a second-order expansion

$$\begin{aligned}\widehat{L}(x,y) \;=\;& L(\hat{x},\hat{y}) + \nabla_x L(\hat{x},\hat{y}){\cdot}[x-\hat{x}] + \nabla_y L(\hat{x},\hat{y}){\cdot}[y-\hat{y}]\\ &+\tfrac{1}{2}[x-\hat{x}]{\cdot}\nabla^2_{xx}L(\hat{x},\hat{y})[x-\hat{x}] + \tfrac{1}{2}[y-\hat{y}]{\cdot}\nabla^2_{yy}L(\hat{x},\hat{y})[y-\hat{y}]\\ &+[y-\hat{y}]{\cdot}\nabla^2_{yx}L(\hat{x},\hat{y})[x-\hat{x}],\end{aligned} \tag{25}$$

which reduces algebraically to

$$\widehat{L}(x,y) = \hat{d} + \hat{c}{\cdot}x + \tfrac{1}{2}x{\cdot}\hat{P}x + \hat{b}{\cdot}y - \tfrac{1}{2}y{\cdot}\hat{Q}y - y{\cdot}\hat{A}x \tag{26}$$

for a certain choice of constant $\hat{d}$, vectors $\hat{c}$ and $\hat{b}$, and matrices $\hat{P}$, $\hat{Q}$, and $\hat{A}$, with $\hat{P}$ and $\hat{Q}$ symmetric and the expression $y{\cdot}\hat{Q}y$ convex for $y \in Y$. (The latter holds because $\hat{Q} = \nabla^2 k(\hat{y})$ and the function k is assumed to be convex relative to Y.) Associated with $\widehat{L}$ and the sets X and Y is an *extended linear-quadratic* programming problem

$$(\widehat{\mathcal{Q}}) \qquad \text{minimize } \; \hat{c}{\cdot}x + \tfrac{1}{2}x{\cdot}\hat{P}x + \theta_{Y\hat{Q}}(\hat{b} - \hat{A}x) \; \text{ over } \; x \in X,$$

(where the constant $\hat{d}$ has been dropped as unessential). This might be exploited as an approximation of $(\mathcal{P})$ in some iterative scheme. Moreover X and Y might be replaced in $(\widehat{Q})$ by $\widehat{X} = X \cap X_0$ and $\widehat{Y} = Y \cap Y_0$ for polyhedral neighborhoods X_0 of $\hat{x}$ and Y_0 of $\hat{y}$, so that such an approximation would have a trust region character, primally and even dually.

Such ideas offer stimulating prospects for research that could be rewarding for optimization in practice. The extended problem format could be matched with numerical methods able to take direct advantage of the extra structure that modelers could bring out in it. Progress could be made in many directions.

Here, we can't go further than speculation about such possibilities. In the meantime, it's important to know that there's no need to hold back from using the extended format. We'll now demonstrate that in the absence of computer codes tailored to extended nonlinear programming, a technical trick can applied to convert a problem $(\mathcal{P})$ with θ of type θ_{YQ} into a conventional nonlinear programming problem. Not only a solution $\bar{x}$ but also an associated multiplier vector $\bar{y}$ for it in $(\mathcal{P})$ can be obtained then by calling on existing software to solve the converted problem. Moreover the conversion can be achieved in terms of standardized representations of X, Y and Q so that, if desirable, it can be carried out automatically in a programming interface which the user doesn't even have to be aware of. That wouldn't be as efficient presumably as a more direct approach, but could be helpful nevertheless.

Proposition 4. *Let $\theta = \theta_{YQ}$ as in (11) with Q positive-semidefinite and consider any representations of the form*

$$Y = \{y \,|\, S^T y \leq s\}, \qquad Q = DJ^{-1}D^T, \tag{27}$$

where J is symmetric and positive-definite (for instance $J = I$). Then

$$\theta(u) = \inf_{z \geq 0,\, w} \{s{\cdot}z + \tfrac{1}{2} w{\cdot}Jw \,|\, Sz + Dw = u\} \;\; \textit{for every } u. \tag{28}$$

Proof. For fixed u, let $(\mathcal{Q}_1)$ be the problem in (z, w) that underlies the right side of (28). This is the primal problem associated with the saddle point problem for

$$L_1(z, w; y) = s{\cdot}z + \tfrac{1}{2} w{\cdot}Jw + y{\cdot}[u - Sz - Dw]$$

with respect to minimizing in (z, w) with $z \geq 0$ but maximizing in $y \in I\!R^m$. The associated dual problem $(\mathcal{Q}_1)^*$, in the framework of extended linear-quadratic programming that's been presented here, is to maximize over all $y \in I\!R^m$ the expression

$$g(y) = \inf_{z \geq 0,\, w} L_1(z, w; y) = \inf_{z \geq 0,\, w} \{u{\cdot}y + z{\cdot}[s - S^T y] + \tfrac{1}{2} w{\cdot}Jw - w{\cdot}[D^T y]\,\}.$$

Since J is symmetric and positive definite, this calculates out to

$$g(y) = \begin{cases} u{\cdot}y - \frac{1}{2}[D^T y]{\cdot}J^{-1}[D^T y] & \text{if } y \in Y, \\ \infty & \text{if } y \notin Y. \end{cases}$$

Thus, the feasible solution set in $(\mathcal{Q}_1^*)$ is Y, which is nonempty, whereas the optimal value in $(\mathcal{Q}_1^*)$ is

$$\sup_{y \in Y} \{u{\cdot}y - \tfrac{1}{2}[D^T y]{\cdot}J^{-1}[D^T y]\,\} = \sup_{y \in Y} \{u{\cdot}y - \tfrac{1}{2} y{\cdot}[DJ^{-1}D^T]y\} = \theta_{YQ}(u).$$

It follows from the duality facts in Theorem 5, as specialized here, that this value and the one in (28) are equal; either both are finite, or both are ∞. □

Theorem 6. *Let $(\mathcal{P})$ be a problem of extended nonlinear programming with $\theta = \theta_{YQ}$ as in (11). Let Y and Q be represented as in (27) and express $X = \{x \mid Rx \leq r\}$ for some matrix R and vector r. Write $F(x) = (f_1(x), \ldots, f_m(x))$. Then the optimal solutions $\bar{x}$ to $(\mathcal{P})$ are the $\bar{x}$ components of the optimal solutions $(\bar{x}, \bar{z}, \bar{w})$ to*

$$(\overline{\mathcal{P}}_0) \qquad \begin{array}{l} \textit{minimize} \ \ f_0(x) + s{\cdot}z + \tfrac{1}{2}w{\cdot}Jw \ \ \textit{subject to} \\ Rx - r \leq 0, \ \ -z \leq 0, \ \ F(x) - Sz - Dw = 0, \end{array}$$

which belongs to conventional nonlinear programming. Moreover a multiplier vector $\bar{y}$ for the equation constraint in $(\overline{\mathcal{P}}_0)$ in the usual sense will be a multiplier vector associated with $\bar{x}$ in the sense of the extended Lagrangian for $(\mathcal{P})$.

Proof. The validity of the reformulation is immediate from the expression for θ achieved in the lemma. The claim about multiplier vectors $\bar{y}$ comes out of the duality developed in the proof of the lemma, according to which the multiplier vectors for the constraint $u - Sz - Dw = 0$ in $(\mathcal{Q}_1)$ must, on the basis of Theorem 5, be the optimal solutions to the dual problem $(\mathcal{Q}_1^*)$. Since the maximization in that dual problem expresses the conjugacy formula

$$\theta_{YQ}(u) = \sup_{y \in R^m} \{u{\cdot}y - \psi(y)\} \ \text{ for } \ \psi(y) = \begin{cases} \frac{1}{2}y{\cdot}Qy & \text{if } y \in Y, \\ \infty & \text{if } y \notin Y, \end{cases}$$

such vectors y are precisely the subgradients of $\theta = \theta_{YQ}$ at u. That means in the case of $\bar{u} = F(\bar{x})$ that they are the elements of $\partial\theta(F(\bar{x})$. But those vectors, as in (18), are known to be the multiplier vectors $\bar{y}$ associated with $\bar{x}$ in the sense of the optimality condition in Theorem 1 for $(\mathcal{P})$. □

Corollary. *In a problem $(\mathcal{Q})$ of extended linear-quadratic programming in which Y and Q are furnished expressions of the kind in (27), the optimal solutions $\bar{x}$ to $(\mathcal{Q})$ are the $\bar{x}$ components of the optimal solutions $(\bar{x}, \bar{z}, \bar{w})$ to*

$$(\overline{Q}_0) \qquad \begin{array}{c} \textit{minimize} \ \ c{\cdot}x + \tfrac{1}{2}x{\cdot}Px + s{\cdot}z + \tfrac{1}{2}w{\cdot}Jw \ \ \textit{subject to} \\ x \in X, \ \ , -z \leq 0, \ \ b - Ax - Sz - Dw = 0. \end{array}$$

Moreover the multiplier vectors $\bar{y}$ for the constraint $b - Ax - Sz - Dw = 0$ in this reformulated problem are identical to the ones associated with $\bar{x}$ in $(\mathcal{Q})$. Hence, if P is positive semidefinite, they are the optimal solutions to the dual problem of extended linear-quadratic programming, $(\mathcal{Q}^*)$.

References

[1] R.T. Rockafellar and R.J-B Wets (1998), *Variational Analysis*, Springer Verlag, Berlin.

[2] R.T. Rockafellar (1987), “Linear-quadratic programming and optimal control,” *SIAM J. Control Opt.* 25, 781–814.

[3] R.T. Rockafellar and R.J-B Wets (1986), “A Lagrangian finite-generation technique for solving linear-quadratic problems in stochastic programming,” *Math Programming Studies* 28, 63–93.

[4] R.T. Rockafellar and R.J-B Wets (1990), “Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time,” *SIAM J. Control Opt.* 28, 810–822.

[5] R.T. Rockafellar (1987), “Computational schemes for large-scale problems in extended linear-quadratic programming,” *Math. Programming* 48, 447–744.

[6] R.T. Rockafellar and C.-Y. Zhu (1987), “Primal-dual projected gradient algorithms for extended linear-quadratic programming,” *SIAM J. Optimization* 3, 751–761.

[7] H.W. Kuhn and A.W. Tucker (1951), “Nonlinear programming,” *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (J. Neyman, ed.), Univ. of California Press, Berkeley, California, 481–492.

[8] R.T. Rockafellar (1970), *Convex Analysis*, Princeton University Press, Princeton, New Jersey (available from 1997 also in paperback in the series Princeton Landmarks in Mathematics).

[9] R.T. Rockafellar (1993), “Lagrange multipliers and optimality,” *SIAM Review* 35, 183–238.

[10] A.L. Dontchev and R.T. Rockafellar (1996), “Characterizations of strong regularity for variational inequalities over polyhedral sets,” *SIAM J. Optimization* 6, 1087–1105.