SIAM J. Optimization
Vol. ???, No. ???, pp. ???–???, ??? 1992

PRIMAL-DUAL PROJECTED GRADIENT ALGORITHMS FOR EXTENDED LINEAR-QUADRATIC PROGRAMMING*

CIYOU ZHU† and R. T. ROCKAFELLAR‡

Abstract. Many large-scale problems in dynamic and stochastic optimization can be modeled with extended linear-quadratic programming, which admits penalty terms and treats them through duality. In general the objective functions in such problems are only piecewise smooth and must be minimized or maximized relative to polyhedral sets of high dimensionality. This paper proposes a new class of numerical methods for "fully quadratic" problems within this framework, which exhibit second-order nonsmoothness. These methods, combining the idea of finite-envelope representation with that of modified gradient projection, work with local structure in the primal and dual problems simultaneously, feeding information back and forth to trigger advantageous restarts.

Versions resembling steepest descent methods and conjugate gradient methods are presented. When a positive threshold of ε-optimality is specified, both methods converge in a finite number of iterations. With threshold 0, it is shown under mild assumptions that the steepest descent version converges linearly, while the conjugate gradient version still has a finite termination property. The algorithms are designed to exploit features of primal and dual decomposability of the Lagrangian, which are typically available in a large-scale setting, and they are open to considerable parallelization.

Key words. Extended linear-quadratic programming, large-scale numerical optimization, finite-envelope representation, gradient projection, primal-dual methods, steepest descent methods, conjugate gradient methods.

AMS(MOS) subject classifications. 65K05, 65K10, 90C20

1. Introduction.

A number of recent papers have described "extended linear-quadratic programming" as a modeling scheme that is much more flexible for problems of optimization than conventional quadratic programming and seems especially suited to large-scale applications, in particular because of way penalty terms can be incorporated. Rockafellar and Wets in [1], [2], first used the concept in two-stage stochastic programming, where the primal dimension is low but the dual dimension is high. It was developed further in its own right in Rockafellar [3], [4], and carried in the latter paper to the context of continuous-time optimal control. Discrete-time problems of optimal control, both deterministic and stochastic (i.e., multistage stochastic programming) were analyzed as extended linear-quadratic programming problems in Rockafellar and Wets [5] and shown to have a remarkable property of Lagrangian decomposability in the primal and dual arguments, both of which can be high dimensional. These models raise new computational challenges and possibilities.

A foundation for numerical schemes in large-scale extended linear-quadratic programming has been laid in Rockafellar [6] and elaborated for problems in multistage format in Rockafellar [7]. The emphasis in [6] is on basic *finite-envelope methods*, which use duality in generating envelope approximations to the primal and dual objective functions through a finite sequence of separate minimizations or maximizations of the Lagrangian. These methods generalize the one originally proposed in [1] for two-stage stochastic programming and implemented by King [8] and Wagner [9]. They center

*Received by the editors December ???, 1990; accepted for publication (in revised form) ??? ???, 1992. This work was supported in part by AFOSR grants 87-0281 and 89-0081 and NSF grants DMS-8701768 and DMS-8819586.

†Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD 21218.

‡Department of Applied Mathematics, FS-20, University of Washington, Seattle, WA 98195.

on the "fully quadratic" case, where strong convexity is present in both the primal and dual objectives, relying on exterior schemes such as the proximal point algorithm to create such strong convexity iteratively when it might otherwise be lacking.

Here we propose new algorithms which for fully quadratic problems combine the idea of finite-envelope representation with that of nonlinear gradient projection. In these methods the envelope approximations are utilized in a sort of steepest descent format or conjugate gradient format in the primal and dual problems simultaneously. A type of feedback is introduced between primal and dual that takes advantage of information jointly uncovered in computations, which in practice greatly speeds convergence. Both algorithms fit into a fundamental scheme for which global convergence is established. Under a weak geometric assumption akin to strict complementary slackness at optimality, the steepest descent version is shown to converge at a linear rate, while the conjugate gradient version has a finite termination property.

Both versions differ significantly from their traditional namesakes not only through the incorporation of a primal-dual scheme of gradient projection, but also in handling objective functions that generally could involve a complicated polyhedral "cell" structure not conducive to explicit description by linear equations and inequalities. They treat the underlying constraints without resorting to an active set strategy, which would not be suitable for problems having high dimensionality in both primal and dual.

An important feature is that the computations are not carried out in terms of a large, sparse matrix, such as might in principle serve in part to specify the two problems, but through subroutines for separate minimization and maximization of the Lagrangian in its primal and dual arguments. This framework appears much better adapted to the special structure available in dynamic and stochastic applications, and it supports extensive parallelization. To make this point clearer, and to introduce facts and notation that will later be needed, we discuss briefly the nature of extended linear-quadratic programming and the way it differs from ordinary quadratic programming.

From the Lagrangian point of view, extended linear-quadratic programming is directed toward finding a saddle point $(\bar{u}, \bar{v})$ of a function

$$L(u,v) = p\cdot u + \tfrac{1}{2}u\cdot Pu + q\cdot v - \tfrac{1}{2}v\cdot Qv - v\cdot Ru \quad \text{over} \quad U \times V, \tag{1.1}$$

where U and V are nonempty polyhedral (convex) sets in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively, and the matrices $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric and positive semidefinite. (One has $p \in \mathbb{R}^n$, $q \in \mathbb{R}^m$, and $R \in \mathbb{R}^{m\times n}$.) Associated with L, U and V are the primal and dual problems

$$(\mathcal{P}) \qquad \text{minimize } f(u) \text{ over all } u \in U, \text{ where } f(u) := \sup_{v\in V} L(u,v),$$

$$(\mathcal{Q}) \qquad \text{maximize } g(v) \text{ over all } v \in V, \text{ where } g(v) := \inf_{u\in U} L(u,v).$$

We speak of the *fully quadratic* case of $(\mathcal{P})$ and $(\mathcal{Q})$ when both of the matrices P and Q are actually positive definite.

Standard quadratic programming would correspond to $Q = 0$ and $V = \mathbb{R}^{m_1}_+ \times \mathbb{R}^{m_2}$. Then f would consist of a quadratic function plus the indicator of a system of m_1 linear inequality constraints and m_2 linear equations, the indicator being the function which assigns an infinite penalty whenever these constraints are violated. Other choices of Q and V yield *finite* penalty expressions of various kinds. This is explained in [4, Secs. 2 and 3] with many examples. For sound modeling in large-scale

applications with dynamics and stochastics such as in [1], [2] and [5], it appears wise to use finite rather than infinite penalties whenever constraints are "soft." Extended linear-quadratic programming makes this option conveniently available. To the extent that constraints in the primal problem are "hard," they can be handled either by placing them in the definition of the polyhedron U or through an augmented Lagrangian technique which corresponds to an exterior scheme of iterations of the proximal point algorithm, as already mentioned.

THEOREM 1.1. [4] (Properties of the objective functions.) *The objective functions f in $(\mathcal{P})$ and g in $(\mathcal{Q})$ are piecewise linear-quadratic: in each case the space can be partitioned in principle into a finite collection of polyhedral cells, relative to which the function has a linear or quadratic formula. Moreover, f is convex while g is concave. In the fully quadratic case of $(\mathcal{P})$ and $(\mathcal{Q})$, f is strongly convex and g is strongly concave, both functions having continuous first derivatives.*

THEOREM 1.2. [4], [1] (Duality and optimality.)

(a) *If either of the optimal values* $\inf(\mathcal{P})$ *or* $\sup(\mathcal{Q})$ *is finite, then both are finite and equal, in which event optimal solutions $\bar{u}$ and $\bar{v}$ exist for the two problems. In the fully quadratic case in particular, the optimal values* $\inf(\mathcal{P})$ *and* $\sup(\mathcal{Q})$ *are finite and equal; then, moreover, the optimal solutions $\bar{u}$ and $\bar{v}$ are unique.*

(b) *A pair $(\bar{u}, \bar{v})$ is a saddle point of $L(u, v)$ over $U \times V$ if and only if $\bar{u}$ solves $(\mathcal{P})$ and $\bar{v}$ solves $(\mathcal{Q})$, or equivalently, $f(\bar{u}) = g(\bar{v})$.*

Current numerical methods in standard quadratic programming, and the somewhat more general area of linear complementarity problems [10], where $U = \mathbb{R}^n_+$, $V = \mathbb{R}^m_+$, and Q is not necessarily the zero matrix, are surveyed by Lin and Pang [11]. Other efforts in recent times have been made by Ye and Tse [12], Monteiro and Adler [13], Goldfarb and Liu [14].

None of these approaches is consonant with the large-scale applications that attract our interest, because the structure in such applications is not well served by the wholesale reformulations that would be required when penalty expressions are much involved. Although any problem of extended linear-quadratic programming can in principle be recast as a standard problem in quadratic programming, as established in [1, Theorem 1], there is a substantial price to be paid in dimensionality and loss of symmetry, as well as in potential ill-conditioning. If the original problem had n primal and m dual variables, and the expression of U and V involved m' and n' constraints beyond nonnegavity of variables, then the reformulated problem in standard format would generally have $n + n' + m$ primal and $m + m'$ dual variables, and its full constraint system would tend to degeneracy (see [1, proof of Theorem 1]). The dual problem would be quite different in its theoretical properties from the primal problem, so that computational ideas developed for the one could not be applied to the other.

Any problem of extended linear-quadratic programming can alternatively be posed in terms of solving a certain linear variational inequality (generalized equation) as explained in [6, Theorem 2.3], and from that one could pass to a linear complementarity model. Symmetry and the meaningful representation of dynamic and stochastic structure could be maintained to a larger extent in this manner. But linear complementarity algorithms tend to be less robust than methods utilizing objective function values, and an increase in dimensionality would still be required in handling constraints, even if these are simply upper and lower bounds on the variables. Furthermore, such algorithms typically have to be carried to completion. They do not generate sequences of primal-feasible and dual-feasible solutions along with estimates of how far these are from being optimal, as is highly desirable when problem size borders on the difficult.

While much could be said about the special problem structure in dynamic and stochastic applications [5], [7], it can be summarized for present purposes in the assertion that such problems, when formulated with care, satisfy the *double decomposability assumption* [6]. This means that for any fixed $u \in U$ it is relatively easy to maximize $L(u, v)$ over $v \in V$, and likewise, for any fixed $v \in V$ it is relatively easy to minimize $L(u, v)$ over $u \in U$, usually because of separability when either of the Lagrangian arguments is considered by itself. These subproblems of maximization and minimization calculate not only the objective values $f(u)$ and $g(v)$ but also, in the fully quadratic case where L is strongly convex-concave, the uniquely determined vectors

$$F(u) = \operatorname*{argmax}_{v \in V} L(u, v) \quad \text{and} \quad G(v) = \operatorname*{argmin}_{u \in U} L(u, v). \tag{1.2}$$

The issue is how to make use of such information in the design of numerical methods. Some proposals have already been made in Rockafellar [6]. Other ideas, which involve splitting algorithms, have been explored by Tseng [15], [16]. Here we aim at adapting classical descent algorithms with help from convex analysis [17].

In this paper we make the blanket assumption of double decomposability, taking it as license also for exact *line searchability* [6]: the supposition that it is possible to minimize $f(u)$ over any line segment joining two points in U, and likewise, to maximize $g(v)$ over any line segment joining two points in V. We focus on the fully quadratic case, even though standard quadratic programming is thereby excluded and a direct comparison with other computational approaches, apart from the finite-envelope methods in [6], becomes difficult. Our attention to that case is justified by its own potential in mathematical modeling (cf. [2], [4]) and because strong convexity-concavity of the Lagrangian can be created, if need be, through some outer implementation of the proximal point algorithm [18], [19], as carried out in [1] and [8]. The questions concerning such an outer algorithm are best handled elsewhere, since they have a different character and relate to a host of primal-dual procedures in extended linear-quadratic programming besides the ones developed here, cf. [1], [2], [6]. In particular, such questions are taken up in Zhu [20].

The supposition that line searches can be carried out exactly is an expedient to allow us to concentrate on more important matters for now. It is also in keeping with the exploration of finite termination properties of the kind usually associated with conjugate gradient-like algorithms, which is part of our agenda. One may observe also that because of the piecewise linear-quadratic nature of the objective functions in Theorem 1.1, line searches in our context are of a special kind where "exactness" is not far-fetched.

A common sort of problem structure which fits with double decomposability is the *box-diagonal case*, where P and Q are diagonal matrices,

$$P = \operatorname{diag}[\alpha_1, \ldots, \alpha_n] \quad \text{and} \quad Q = \operatorname{diag}[\beta_1, \ldots, \beta_m], \tag{1.3}$$

the entries α_j and β_i being positive (for fully quadratic problems), while U and V are boxes representing upper and lower bounds (not necessarily finite) on the components of $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_m)$:

$$U = [u_1^-, u_1^+] \times \cdots \times [u_n^-, u_n^+] \quad \text{and} \quad V = [v_1^-, v_1^+] \times \cdots \times [v_m^-, v_m^+]. \tag{1.4}$$

In this case, we have for each $u \in U$ that the problem of maximizing $L(u, v)$ over $v \in V$ to obtain $f(u)$ and $F(u)$ decomposes into separate *one-dimensional* subproblems in

the individual coordinates: for $i = 1, \ldots, m$

$$\text{maximize } \left[q_i - \sum_{j=1}^{n} r_{ij}u_j\right]\cdot v_i - \tfrac{1}{2}\beta_i v_i^2 \text{ subject to } v_i^- \le v_i \le v_i^+. \tag{1.5}$$

Likewise, the problem of minimizing $L(u, v)$ over $u \in U$ for given $v \in V$, so as to calculate $g(v)$ and $G(v)$, reduces to the separate problems

$$\text{minimize } \left[p_j - \sum_{i=1}^{m} v_i r_{ij}\right]\cdot u_i + \tfrac{1}{2}\alpha_j u_j^2 \text{ subject to } u_j^- \le u_j \le u_j^+. \tag{1.6}$$

Clearly, there exist very simple *closed-form* solutions to these one-dimensional subproblems. No actual minimization or maximization routine needs to be invoked. Often there are ways also of obtaining the answers without explicitly introducing the r_{ij}'s.

In notation, we shall refer consistently to

$$\begin{aligned} \bar{u} &= \text{ the unique optimal solution to } (\mathcal{P}), \\ \bar{v} &= \text{ the unique optimal solution to } (\mathcal{Q}), \end{aligned} \tag{1.7}$$

these properties meaning by Theorem 1.2 that

$$(\bar{u}, \bar{v}) = \text{ the unique saddle point of } L \text{ on } U \times V, \tag{1.8}$$

or equivalently in terms of the mappings F and G that

$$\bar{v} = F(\bar{u}) \quad \text{and} \quad \bar{u} = G(\bar{v}). \tag{1.9}$$

Furthermore, we shall write

$$\begin{aligned} \|u\|_P = [u{\cdot}Pu]^{\frac{1}{2}} \quad &\text{and} \quad \|v\|_Q = [v{\cdot}Qv]^{\frac{1}{2}}, \\ \langle w, u\rangle_P = w{\cdot}Pu \quad &\text{and} \quad \langle z, v\rangle_Q = z{\cdot}Qv \end{aligned} \tag{1.10}$$

for the norms and inner products corresponding to the positive definite matrices P and Q. It is these norms and inner products, rather than the canonical ones, that intrinsically underlie the analysis of our problems, and it is well to bear this in mind. Just as the function f, if it is $\mathcal{C}^2$ around a point u, can be expanded as

$$f(u') = f(u) + \big\langle \nabla f(u),\, u' - u\big\rangle + \tfrac{1}{2}\big\langle u' - u,\, \nabla^2 f(u)(u' - u)\big\rangle + o\big(\|u' - u\|^2\big),$$

it can also be expanded as

$$f(u') = f(u) + \big\langle \nabla_P f(u),\, u' - u\big\rangle_P + \tfrac{1}{2}\big\langle u' - u,\, \nabla_P^2 f(u)(u' - u)\big\rangle_P + o\big(\|u' - u\|_P^2\big)$$

for a certain vector $\nabla_P f(u)$ and a certain matrix $\nabla_P^2 f(u)$; similarly for g in terms of $\nabla_Q g(v)$ and $\nabla_Q^2 g(v)$. Clearly,

$$\begin{aligned} \nabla_P f(u) &= P^{-1}\nabla f(u), \qquad \nabla_P^2 f(u) = P^{-1}\nabla^2 f(u), \\ \nabla_Q g(v) &= Q^{-1}\nabla g(v), \qquad \nabla_Q^2 g(v) = Q^{-1}\nabla^2 g(v). \end{aligned} \tag{1.11}$$

In appealing to this symbolism we shall better be able to bring out the basic structure and convergence properties of the proposed algorithms.

We cite now from [6] several fundamental properties on which the algorithmic developments in this paper will depend.

PROPOSITION 1.3. [6, p. 459] (Optimality estimates.) *Suppose $\hat{u}$ and $\hat{v}$ are elements of U and V satisfying $f(\hat{u})-g(\hat{v}) \le \varepsilon$, where $\varepsilon \ge 0$. Then $\hat{u}$ and $\hat{v}$ are ε-optimal in the sense that $|f(\hat{u}) - f(\bar{u})| \le \varepsilon$ and $|g(\hat{v}) - g(\bar{v})| \le \varepsilon$. Moreover, $\|\hat{u} - \bar{u}\|_P \le \sqrt{2\varepsilon}$ and $\|\hat{v} - \bar{v}\|_Q \le \sqrt{2\varepsilon}$.*

PROPOSITION 1.4. [6, pp. 438, 469] (Regularity properties.) *The functions f and g are continuously differentiable everywhere, and the mappings F and G are Lipschitz continuous:*

$$\begin{aligned} \nabla f(u) &= \nabla_u L(u, F(u)) = p + Pu - R^T F(u), \\ \nabla g(v) &= \nabla_v L(G(v), v) = q - Qv - RG(v), \end{aligned} \tag{1.12}$$

where in terms of the constant

$$\gamma(P, Q, R) := \|Q^{-\frac{1}{2}} R P^{-\frac{1}{2}}\| \tag{1.13}$$

one has

$$\begin{aligned} &\|F(u') - F(u)\|_Q \le \gamma(P, Q, R)\|u' - u\|_P \text{ for all } u \text{ and } u', \\ &\|G(v') - G(v)\|_P \le \gamma(P, Q, R)\|v' - v\|_Q \text{ for all } v \text{ and } v'. \end{aligned} \tag{1.14}$$

The finite-envelope idea enters through repeated application of the mappings F and G. The rationale is discussed at length in [6], but the main facts needed here are in the next two propositions.

PROPOSITION 1.5. [6, p. 460] (Envelope properties.) *For arbitrary $u_0 \in U$ and $v_0 \in V$, let $v_1 = F(u_0)$ and $u_1 = G(v_0)$, followed by $v_2 = F(u_1)$ and $u_2 = G(v_1)$. Then in the primal problem*

(1.15)
$$\begin{aligned} &f(u) \ge L(u, v_1) \text{ for all } u, \text{ with } L(u_0, v_1) = f(u_0) \text{ and } \nabla_u L(u_0, v_1) = \nabla f(u_0), \\ &f(u) \ge L(u, v_2) \text{ for all } u, \text{ with } L(u_1, v_2) = f(u_1) \text{ and } \nabla_u L(u_1, v_2) = \nabla f(u_1), \end{aligned}$$

while in the dual problem

(1.16)
$$\begin{aligned} &g(v) \le L(u_1, v) \text{ for all } v, \text{ with } L(u_1, v_0) = g(v_0) \text{ and } \nabla_v L(u_1, v_0) = \nabla g(v_0), \\ &g(v) \le L(u_2, v) \text{ for all } v, \text{ with } L(u_2, v_1) = g(v_1) \text{ and } \nabla_v L(u_2, v_1) = \nabla g(v_1). \end{aligned}$$

PROPOSITION 1.6. [6, p. 470] (Modified gradient projection.) *For arbitrary $u_0 \in U$ and $v_0 \in V$, let $v_1 = F(u_0)$ and $u_1 = G(v_0)$, followed by $v_2 = F(u_1)$ and $u_2 = G(v_1)$. Then*

$$\begin{aligned} L(u, v_1) &= f(u_0) + \nabla f(u_0)\cdot(u - u_0) + \tfrac{1}{2}(u - u_0)\cdot P(u - u_0) \\ &= f(u_0) + \langle \nabla_P f(u_0), u - u_0 \rangle_P + \tfrac{1}{2}\|u - u_0\|_P^2, \\ &= \tfrac{1}{2}\big\|(u - u_0) + \nabla_P f(u_0)\big\|_P^2 + \text{ const.} \end{aligned} \tag{1.17}$$

$$\begin{aligned} L(u_1, v) &= g(v_0) + \nabla g(v_0)\cdot(v - v_0) - \tfrac{1}{2}(v - v_0)\cdot Q(v - v_0) \\ &= g(v_0) + \langle \nabla_Q g(v_0), v - v_0 \rangle_Q - \tfrac{1}{2}\|v - v_0\|_Q^2, \\ &= -\tfrac{1}{2}\big\|(v - v_0) - \nabla_Q g(v_0)\big\|_Q^2 + \text{ const.} \end{aligned} \tag{1.18}$$

so from the definition of u_2 and v_2 one has that

$$\begin{aligned} u_2 - u_0 &= \ P\text{-projection of } \ -\nabla_P f(u_0) \ \text{ on } U - u_0, \\ v_2 - v_0 &= \ Q\text{-projection of } \ \nabla_Q g(v_0) \ \text{ on } V - v_0. \end{aligned} \tag{1.19}$$

Proof. The first equation in (1.17) expands $L(\cdot, v_1)$ at u_0 in accordance with (1.15), and the rest of (1.17) re-expresses this via (1.10) and (1.11). Since $u_2 := \operatorname{argmin}_{u\in U} L(u, v_1)$, u_2 is thus the $\|\cdot\|_P$-nearest point of U to $u_0 - \nabla_P f(u_0)$, so $u_2 - u_0$ is the $\|\cdot\|_P$-projection of $-\nabla_P f(u_0)$ on $U - u_0$. The assertions in the v-argument are verified similarly. □

The formulas in (1.19) give the precise form of (nonlinear) *gradient projection* that is available through our assumed ability to calculate $F(u)$ and $G(v)$ whenever we please. It is this form, therefore, that we shall incorporate in our algorithms. The reader should note this carefully, or a crucial feature of our approach, in its applicability to large-scale problems, will be missed. Although the gradients of f and g exist and are expressed by the formulas in Proposition 1.4, we do not have to calculate them through these formulas, much less apply a subroutine for gradient projection. In particular, *it is not necessary to generate or store the potentially huge or dense matrix R.* To execute our algorithms, one only needs to be able to generate the points u_1, u_2, v_1 and v_2 from a given pair u_0 and v_0. As explained, this can be done through subroutines which minimize or maximize the Lagrangian individually in the primal or dual argument, cf. (1.2). For multistage, possibly stochastic, optimization problems expressed in the format of [1], [2], and [6], such subroutines can easily be written in terms of the underlying data structure (without ever introducing R!).

In obtaining our results about local rates of convergence, a mild condition on the optimal solutions $\bar{u}$ and $\bar{v}$ will eventually be required. To formulate it, we introduce the sets

$$U_0 := \operatorname*{argmin}_{u\in U} \nabla_u L(\bar{u}, \bar{v})\cdot u = \operatorname*{argmin}_{u\in U} \nabla f(\bar{u})\cdot u = \operatorname*{argmin}_{u\in U} \big\langle \nabla_P f(\bar{u}), u \big\rangle_P, \tag{1.20}$$

$$V_0 := \operatorname*{argmax}_{v\in V} \nabla_v L(\bar{u}, \bar{v})\cdot v = \operatorname*{argmax}_{v\in V} \nabla g(\bar{v})\cdot v = \operatorname*{argmax}_{v\in V} \big\langle \nabla_Q g(\bar{v}), v \big\rangle_Q, \tag{1.21}$$

which are called the *critical faces* of U and V in $(\mathcal{P})$ and $(\mathcal{Q})$ [6]. They are closed faces of the polyhedral sets U and V, and they contain the optimal solutions $\bar{u}$ and $\bar{v}$, respectively, by virtue of the elementary conditions for the minimum of a smooth convex function (or the maximum of a smooth concave function).

DEFINITION 1.7. (*Critical face condition.*) The *critical face* condition will be said to be satisfied at the optimal solutions $\bar{u}$ and $\bar{v}$ if $\bar{u} \in \operatorname{ri} U_0$ and $\bar{v} \in \operatorname{ri} V_0$ (where "ri" denotes relative interior in the sense of convex analysis).

We do not add this condition as a standing assumption, but it will be invoked several times in connection with the following property of the envelope mappings F and G, which is implicit in [6, Theorem 5.4] in its proof, but is stated here explicitly.

PROPOSITION 1.8. (Envelope behavior near the critical faces.) *There exist neighborhoods of $\bar{u}$ and $\bar{v}$ with the property that if the points $u_0 \in U$ and $v_0 \in V$ belong to these neighborhoods, then the points*

$$v_1 = F(u_0), \qquad u_1 = G(v_0), \qquad v_2 = F(u_1), \qquad u_2 = G(v_1),$$

will be such that u_1 and u_2 belong to the primal critical face U_0, while v_1 and v_2 belong to the dual critical face V_0. Under the critical face condition, the neighborhoods can be chosen so that u_1 and u_2 actually belong to $\operatorname{ri} U_0$, while v_1 and v_2 belong to $\operatorname{ri} V_0$.

Proof. We adapt the argument given for [6, Theorem 5.4]. From (1.9) and the continuity of F and G in Proposition 1.4, we know that by making u_0 and v_0 close to $\bar{u}$ and $\bar{v}$ we will make u_1 and u_2 close to $\bar{u}$ and v_1 and v_2 close to $\bar{v}$. For each vector $w \in \mathbb{R}^n$, let $M(w)$ be the closed face of the polyhedron U on which the function $u \mapsto w \cdot u$ achieves its minimum. This could be empty for some choices of w, but in the case of $\bar{w} = \nabla_u L(\bar{u}, \bar{v})$ it is U_0, which contains $\bar{u}$. The graph of M as a set-valued mapping is closed (as can be verified directly or through the observation that M is the subdifferential of the support function of U, cf. [17, Secs. 13, 23]), and M has only finitely many values (since U has only finitely many faces). It follows that $M(w) \subset M(\bar{w}) = U_0$ when w is in some neighborhood of $\bar{w}$. We can apply this in particular to $w = \nabla_u L(u_1, v_0)$, noting that this vector will be close to $\bar{w}$ when u_0 and v_0 are sufficiently close to $\bar{u}$ and $\bar{v}$. The point u_1 minimizes $L(u, v_0)$ over $u \in U$ and therefore has the property that $\nabla_u L(u_1, v_0) \cdot (u - u_1) \leq 0$ for all $u \in U$, which means $u_1 \in M(w)$. Therefore $u_1 \in U_0$ when u_0 and v_0 are sufficiently close to $\bar{u}$ and $\bar{v}$.

Parallel reasoning demonstrates that $v_1 \in V_0$ under such circumstances. If the critical face condition holds, then as u_1 and v_1 approach $\bar{u}$ and $\bar{v}$ they must actually enter the relative interiors $\operatorname{ri} U_0$ and $\operatorname{ri} V_0$. The same argument can be applied now to reach these conclusions for u_2 and v_2. □

2. Formulation of the Algorithms. The new methods for the fully quadratic case of problems $(\mathcal{P})$ and $(\mathcal{Q})$ will be formulated as conceptual algorithms involving line search. The convergence analysis will be undertaken in Sections 3, 4, and 5, and the numerical test results will be given in Section 6.

In what follows, we use $[w_1, w_2]$ to denote the line segment between two points w_1 and w_2, and we use ν as the running index for iterations.

The main characteristic of the new methods is the coupling of line search procedures in the primal and dual problems with interactive restarts. To assist the reader in understanding this, we first formulate the method analogous to steepest descent, where there are fewer parameters and the algorithmic logic is simpler.

Algorithm 1. (*Primal-Dual Steepest Descent Algorithm*, PDSD.) Construct primal and dual sequences $\{u_0^\nu\} \subset U$ and $\{v_0^\nu\} \subset V$ as follows.

Step 0 (initialization). Choose a real value for the parameter $\varepsilon \geq 0$ (optimality threshold). Set $\nu := 0$ (iteration counter). Specify starting points $\hat{u}_0^0 \in U$ and $\hat{v}_0^0 \in V$ for the sequences $\{\hat{u}_0^\nu\} \subset U$ and $\{\hat{v}_0^\nu\} \subset V$ that will be generated along with $\{u_0^\nu\}$ and $\{v_0^\nu\}$.

Step 1 (evaluation). Calculate

$$\begin{cases} f(\hat{u}_0^\nu),\ g(\hat{v}_0^\nu), & \text{obtaining as by-products } \hat{v}_1^\nu = F(\hat{u}_0^\nu),\ \hat{u}_1^\nu = G(\hat{v}_0^\nu), \\ g(\hat{v}_1^\nu),\ f(\hat{u}_1^\nu), & \text{obtaining as by-products } \hat{u}_2^\nu = G(\hat{v}_1^\nu),\ \hat{v}_2^\nu = F(\hat{u}_1^\nu). \end{cases}$$

Step 2 (interactive restarts). Take

$$\begin{cases} u_0^\nu := \hat{u}_0^\nu, v_1^\nu := \hat{v}_1^\nu, u_2^\nu := \hat{u}_2^\nu & \text{if } f(\hat{u}_0^\nu) \leq f(\hat{u}_1^\nu), \\ u_0^\nu := \hat{u}_1^\nu, v_1^\nu := \hat{v}_2^\nu, u_2^\nu := G(v_1^\nu) & \text{otherwise (this is an interactive primal restart).} \end{cases}$$

$$\begin{cases} v_0^\nu := \hat{v}_0^\nu, u_1^\nu := \hat{u}_1^\nu, v_2^\nu := \hat{v}_2^\nu & \text{if } g(\hat{v}_0^\nu) \geq g(\hat{v}_1^\nu), \\ v_0^\nu := \hat{v}_1^\nu, u_1^\nu := \hat{u}_2^\nu, v_2^\nu := F(u_1^\nu) & \text{otherwise (this is an interactive dual restart).} \end{cases}$$

(In an interactive primal restart, the calculation of $G(v_1^\nu)$ yields the new $g(v_1^\nu)$. Likewise, in an interactive dual restart, the calculation of $F(u_1^\nu)$ yields the new $f(u_1^\nu)$.)

Step 3 (optimality test). Let

$$\hat{u} := \begin{cases} u_0^\nu & \text{if } f(u_0^\nu) \le f(u_1^\nu), \\ u_1^\nu & \text{if } f(u_0^\nu) > f(u_1^\nu), \end{cases} \qquad \text{and} \qquad \hat{v} := \begin{cases} v_0^\nu & \text{if } g(v_0^\nu) \ge g(v_1^\nu), \\ v_1^\nu & \text{if } g(v_0^\nu) < g(v_1^\nu). \end{cases}$$

If $f(\hat{u}) - g(\hat{v}) \le \varepsilon$, terminate with $\hat{u}$ and $\hat{v}$ being ε-optimal solutions to $(\mathcal{P})$ and $(\mathcal{Q})$.

Step 4 (line segment search). Search for

$$\hat{u}_0^{\nu+1} := \operatorname*{argmin}_{u \in [u_0^\nu, u_2^\nu]} f(u) \qquad \text{and} \qquad \hat{v}_0^{\nu+1} := \operatorname*{argmax}_{v \in [v_0^\nu, v_2^\nu]} g(v).$$

Return then to Step 1 with the counter ν increased by 1.

Basically, the idea in this method is that if the point $\hat{u}_1^\nu$ calculated as a by-product of finding the projected gradient (1.19) in the dual problem gives a better value to the objective in the primal problem than does the current primal point $\hat{u}_0^\nu$, we take it instead as the current primal point (and accordingly recalculate the projected gradient in the primal problem). Likewise, if the point $\hat{v}_1^\nu$ calculated as a by-product of finding the projected gradient (1.19) in the primal problem happens to give a better value to the objective in the primal problem than the current dual point $\hat{v}_0^\nu$, we take it instead as the current dual point (and accordingly recalculate the projected gradient in the dual problem). Here it may be recalled that $\hat{u}_1^\nu$ minimizes over U the convex quadratic function $L(\cdot, \hat{v}_0^\nu)$, which is a lower approximant to the objective function f in $(\mathcal{P})$ that would have the same minimum value as f over U if $\hat{v}_0^\nu$ were dual optimal. By the same token, $\hat{v}_1^\nu$ maximizes over V the concave quadratic function $L(\hat{u}_0^\nu, \cdot)$, which is an upper approximant to the objective function g in $(\mathcal{Q})$ that would have the same maximum value as g over V is $\hat{u}_0^\nu$ were primal optimal.

Once the issue of triggering a primal or dual interactive restart (or both) settles down in a given iteration, we perform line searches in the directions indicated by the projected gradients in the two problems. If U were the whole space $\mathbb{R}^n$, the primal search direction would be the true direction of steepest descent for f (relative to the geometry induced by the Euclidean norm $\|\cdot\|_P$ on $\mathbb{R}^n$). Similarly, if V were the whole space $\mathbb{R}^m$, the dual search direction would be the true direction of steepest ascent for g (relative to the geometry of the Euclidean norm $\|\cdot\|_Q$ on $\mathbb{R}^m$). However, even in this unconstrained case there would be a difference in the way the searches are carried out, in comparison with classical steepest descent, because instead of looking along an entire half-line we only optimize along a line segment whose length is that of the gradient, i.e., we restrict the step size to be at most 1. (Also, we call for an "exact" optimum because the objective is piecewise strictly quadratic with only finitely many pieces. Clearly, this requirement could be loosened, but the issue is minor and we do not wish to be distracted by it here.)

The restriction to a line segment instead of a half-line is motivated in part by the fact that the line segment is known to lie entirely in the feasible set. A search over a half-line would have to cope with detecting the feasibility boundary in the search parameter, which could be a disadvantage in a high-dimensional setting, although this topic could be explored further. Heuristic motivation for the restriction comes also from evidence of second-order effects induced by the primal-dual feedback, as discussed below. It turns out that under mild assumptions the optimal step sizes along a half-line would eventually be no greater than 1 anyway.

The interactive restarts may seem like a merely opportunistic feature of Algorithm 1, but they have a marked effect, as the numerical tests in Section 6 will reveal. When interactive restarts are blocked, so that the algorithm reverts to two

independent procedures in the primal and dual settings (through a sort of computational "lobotomy"), the performance is slowed down to what one might expect from a steepest-descent-like algorithm. On the other hand, when the interactions are permitted the performance in practice is quite comparable to that of more complicated procedures which attempt to exploit second-order properties. The feedback between primal and dual appears able to supply some such information to the calculations.

In order to develop a broader range of interactive-restart methods, analogous not only to steepest descent but to conjugate gradients, we next formulate as Algorithm 0 a bare-bones procedure which will serve in establishing convergence properties of such methods, including Algorithm 1. The chief complication in Algorithm 0 beyond what has already been seen in Algorithm 1 comes through the introduction of *cycles* for primal and dual restarts. With respect to these cycles an additional threshold parameter is introduced as a technical safeguard against interactive restarts being triggered too freely, without assurance of adequate progress.

Algorithm 0. (*General Primal-Dual Projected Gradient Algorithm*, PDPG.) Construct primal and dual sequences $\{u_0^\nu\} \subset U$ and $\{v_0^\nu\} \subset V$ as follows.

Step 0 (initialization). Choose an integer value for the parameter $k > 0$ (cycle size) and real values for the parameters $\varepsilon \geq 0$ (optimality threshold) and $\delta > 0$ (progress threshold). Set $\nu := 0$ (iteration counter), $k_p := 0$ (primal restart counter), and $k_d := 0$ (dual restart counter). Specify starting points $\hat{u}_0^0 \in U$ and $\hat{v}_0^0 \in V$ for the sequences $\{\hat{u}_0^\nu\} \subset U$ and $\{\hat{v}_0^\nu\} \subset V$ that will be generated along with $\{u_0^\nu\}$ and $\{v_0^\nu\}$.

Step 1 (evaluation). Calculate

$$\begin{cases} f(\hat{u}_0^\nu),\ g(\hat{v}_0^\nu), & \text{obtaining as by-products } \hat{v}_1^\nu = F(\hat{u}_0^\nu),\ \hat{u}_1^\nu = G(\hat{v}_0^\nu), \\ g(\hat{v}_1^\nu),\ f(\hat{u}_1^\nu), & \text{obtaining as by-products } \hat{u}_2^\nu = G(\hat{v}_1^\nu),\ \hat{v}_2^\nu = F(\hat{u}_1^\nu). \end{cases}$$

Step 2 (interactive restarts). Take

$$\begin{cases} u_0^\nu := \hat{u}_0^\nu, v_1^\nu := \hat{v}_1^\nu, u_2^\nu := \hat{u}_2^\nu & \text{if } f(\hat{u}_0^\nu) \leq f(\hat{u}_1^\nu), \text{ or } f(\hat{u}_0^\nu) < f(\hat{u}_1^\nu)+\delta \text{ and } k_p < k, \\ u_0^\nu := \hat{u}_1^\nu, v_1^\nu := \hat{v}_2^\nu, u_2^\nu := G(v_1^\nu) & \text{otherwise (this is an interactive primal restart).} \end{cases}$$

$$\begin{cases} v_0^\nu := \hat{v}_0^\nu, u_1^\nu := \hat{u}_1^\nu, v_2^\nu := \hat{v}_2^\nu & \text{if } g(\hat{v}_0^\nu) \geq g(\hat{v}_1^\nu), \text{ or } g(\hat{v}_0^\nu) > g(\hat{v}_1^\nu) - \delta \text{ and } k_d < k, \\ v_0^\nu := \hat{v}_1^\nu, u_1^\nu := \hat{u}_2^\nu, v_2^\nu := F(u_1^\nu) & \text{otherwise (this is an interactive dual restart).} \end{cases}$$

(In an interactive primal restart the calculation of $G(v_1^\nu)$ yields the new $g(v_1^\nu)$. Likewise, in an interactive dual restart the calculation of $F(u_1^\nu)$ yields the new $f(u_1^\nu)$.) Set

$$\begin{cases} k_p := 0 & \text{if an interactive primal restart occurred in this step,} \\ k_d := 0 & \text{if an interactive dual restart occurred in this step.} \end{cases}$$

Step 3 (optimality test). Let

$$\hat{u} := \begin{cases} u_0^\nu & \text{if } f(u_0^\nu) \leq f(u_1^\nu), \\ u_1^\nu & \text{if } f(u_0^\nu) > f(u_1^\nu), \end{cases} \quad \text{and} \quad \hat{v} := \begin{cases} v_0^\nu & \text{if } g(v_0^\nu) \geq g(v_1^\nu), \\ v_1^\nu & \text{if } g(v_0^\nu) < g(v_1^\nu). \end{cases}$$

If $f(\hat{u}) - g(\hat{v}) \leq \varepsilon$, terminate with $\hat{u}$ and $\hat{v}$ being ε-optimal solutions to $(\mathcal{P})$ and $(\mathcal{Q})$.

Step 4 (search endpoint generation). Take

$$\begin{cases} u_e^\nu := u_2^\nu & \text{if } k_p \equiv 0 (\mathrm{mod}\, k), \\ u_e^\nu \in U \text{ according to an auxiliary rule} & \text{otherwise.} \end{cases}$$

$$\begin{cases} v_e^\nu := v_2^\nu & \text{if } k_d \equiv 0 (\mathrm{mod}\, k), \\ v_e^\nu \in V \text{ according an auxiliary rule} & \text{otherwise.} \end{cases}$$

Step 5 (line segment search). Search for

$$\hat{u}_0^{\nu+1} := \operatorname*{argmin}_{u\in[u_0^\nu,u_e^\nu]} f(u) \quad \text{and} \quad \hat{v}_0^{\nu+1} := \operatorname*{argmax}_{v\in[v_0^\nu,v_e^\nu]} g(v).$$

Return then to Step 1 with the counters ν, k_p and k_d increased by 1.

By specifying the auxiliary rules in Step 4 for generating the search interval endpoints u_e^ν and v_e^ν in iterations where k_p or k_d is not a multiple of k, we obtain particular *realizations* of Algorithm 0. An attractive case in which these rules correspond to a "conjugate gradient" approach with cycle size k will be developed presently as Algorithm 2. Before proceeding, however, we want to emphasize for theoretical purposes that Algorithm 1 is itself a particular realization of Algorithm 0.

PROPOSITION 2.1. *Algorithm 0 reduces to Algorithm 1 when the cycle size is* $k = 1$ *(except for a slight difference in iteration* $\nu = 0$*).*

Proof. In returning from Step 4 of Algorithm 0 to Step 1, the counters k_p and k_d are always at least 1. It follows that if $k = 1$ the condition in Step 2 with progress threshold δ will never come into play after such a return. Thus, the only possible effect of this threshold will be in iteration $\nu = 0$, where a restart will be avoided unless it improves the objective by at least δ. In Step 4, k_p and k_d will always be multiples of k, so we will always have $u_e^\nu = u_2^\nu$ and $v_e^\nu = v_2^\nu$. Thus the counters k_p and k_d become redundant and the auxiliary rules moot. □

In Algorithm 0 in general, k_p counts iterations in the primal problem from the start or the most recent interactive primal restart. An iteration that begins with k_p being a positive multiple of k is said to be one in which an *ordinary primal restart* takes place (whether or not an interactive primal restart also takes place), because it marks the completion of a cycle of k iterations not cut short by an interactive primal restart. Every iteration involving an ordinary or interactive primal restart ends by searching the line segment $[u_0^\nu, u_2^\nu]$, where $u_2^\nu - u_0^\nu$ is the negative of the current projected gradient of f in (1.19). The dual situation is parallel in terms of the counter k_d and the notion of an *ordinary dual restart*.

The role of the parameter $\delta > 0$ is to control the extent to which the algorithm forgoes interactive restarts and insists on waiting for ordinary restarts. Interactive restarts are always accepted if they improve the corresponding objective value by the amount δ or more, but there can only be finitely many iterations with this size of improvement, due to the finiteness of the joint optimal value in $(\mathcal{P})$ and $(\mathcal{Q})$ (Theorem 1.1). When such improvement is no longer possible, interactive restarts are blocked in the primal until an ordinary restart has again intervened, unless one is already occurring in the same iteration; the same holds in the dual. This feature ensures that full cycles of k iterations will continue to be performed in the primal and dual as long as the algorithm keeps running, which is important in establishing certain properties of convergence.

Recall that the point u_2^ν minimizes over U the lower envelope function $L(u, v_1)$ as a representation of $f(u)$ at u_0^ν (Proposition 1.5), which has $\nabla_u L(u_0^\nu, v_1^\nu) = \nabla f(u_0^\nu)$. Even apart from the projected gradient interpretation, therefore, there is motivation in searching the line segment $[u_0^\nu, u_2^\nu]$ in order to reduce the objective value $f(u)$ in primal. The same motivation exists for searching $[v_0^\nu, v_2^\nu]$ in the dual.

As a matter of fact, we shall prove in Proposition 5.1 that on exiting from Step 5 (line segment search) of Algorithm 0, the point $\hat{u}_1^{\nu+1} = G(\hat{v}_0^{\nu+1})$ will be the minimum point relative to U for the envelope function

$$f^\nu(u) := \max_{v\in[v_0^\nu,v_2^\nu]} L(u, v) \le \max_{v\in V} L(u, v) = f(u).$$

When the algorithm reaches Step 2 in the iteration, it will compare the point $\hat{u}_0^{\nu+1}$ resulting from the just-completed line search in the primal with the point $\hat{u}_1^{\nu+1}$ resulting from minimizing the lower envelope function $f^\nu(u)$, and it will take the "better" of the two as the next primal iterate. In the dual procedure there are corresponding comparisons between $\hat{v}_0^{\nu+1}$ and $\hat{v}_1^{\nu+1}$.

We focus now on a specialization of Algorithm 0 in which, in contrast to Algorithm 1, the cycle provisions are crucial and the auxiliary rules nontrivial. The rules emulate those of the classical conjugate gradient method (Hestenes-Stiefel).

Algorithm 2. (*Primal-Dual Conjugate Gradient Method*, PDCG.) In the implementation of Algorithm 0, choose a cycle size $k > 1$ and use the following auxiliary rules to get the search intervals in Step 4. Unless $k_p \equiv 0 (\mathrm{mod}\, k)$, set

$$(2.1)\quad w_p^\nu := \nabla_P f(u_0^\nu) - \nabla_P f(u_0^{\nu-1}),$$

$$(2.2)\quad \beta_p^\nu := \begin{cases} \max\{0, \langle w_p^\nu,\ u_0^\nu - u_2^\nu\rangle_P\}/\langle w_p^\nu,\ u_e^{\nu-1} - u_0^\nu\rangle_P & \text{if } \langle w_p^\nu,\ u_e^{\nu-1} - u_0^\nu\rangle_P > 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$(2.3)\quad u_{cg}^\nu := (u_2^\nu + \beta_p^\nu u_e^{\nu-1})/(1+\beta_p^\nu),$$

$$(2.4)\quad [u_0^\nu, u_e^\nu] := \begin{cases} [u_0^\nu, u_{cg}^\nu] & \text{if } \|u_{cg}^\nu - u_0^\nu\|_P \geq 1, \\ L_p^\nu \cap U & \text{otherwise,} \end{cases}$$

where $L_p^\nu = \big\{u \in \mathbb{R}^n \mid u = u_0^\nu + \lambda(u_{cg}^\nu - u_0^\nu),\ 0 \leq \lambda \leq \|u_{cg}^\nu - u_0^\nu\|_P^{-1}\big\}$. Similarly, unless $k_d \equiv 0 (\mathrm{mod}\, k)$, set

$$(2.5)\quad w_d^\nu := -\nabla_Q g(v_0^\nu) + \nabla_Q g(v_0^{\nu-1}),$$

$$(2.6)\quad \beta_d^\nu := \begin{cases} \max\{0, \langle w_d^\nu,\ v_0^\nu - v_2^\nu\rangle_Q\}/\langle w_d^\nu,\ v_e^{\nu-1} - v_0^\nu\rangle_Q & \text{if } \langle w_d^\nu,\ v_e^{\nu-1} - v_0^\nu\rangle_Q > 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$(2.7)\quad v_{cg}^\nu := (v_2^\nu + \beta_d^\nu v_e^{\nu-1})/(1+\beta_d^\nu),$$

$$(2.8)\quad [v_0^\nu, v_e^\nu] := \begin{cases} [v_0^\nu, v_{cg}^\nu] & \text{if } \|v_{cg}^\nu - v_0^\nu\|_Q \geq 1, \\ L_d^\nu \cap V & \text{otherwise,} \end{cases}$$

where $L_d^\nu = \big\{v \in \mathbb{R}^m \mid v = v_0^\nu + \lambda(v_{cg}^\nu - v_0^\nu),\ 0 \leq \lambda \leq \|v_{cg}^\nu - v_0^\nu\|_Q^{-1}\big\}$.

Note that because the auxiliary rules are never invoked in iteration $\nu = 0$ (where $k_p = 0$ and $k_d = 0$), the points indexed with $\nu - 1$ in the statement of Algorithm 2 are all well defined. Another thing to observe is the fact that in (2.2) and (2.6) we actually have

$$(2.9)\qquad \langle w_p^\nu,\ u_e^{\nu-1} - u_0^\nu\rangle_P \geq 0 \quad \text{and} \quad \langle w_d^\nu,\ v_e^{\nu-1} - v_0^\nu\rangle_Q \geq 0.$$

These inequalities follow from (2.1) and (2.5) and the monotonicity of gradient mappings of convex functions. In Proposition 4.4 we shall prove that under the critical face condition the inequalities in (2.9) hold *strictly* in a vicinity of the optimal solution if the critical faces are reached by the corresponding iterates.

On the other hand, it is apparent from (2.3) and (2.7) that

$$(2.10)\qquad u_{cg}^\nu - u_0^\nu = \frac{u_2^\nu - u_0^\nu + \beta_p^\nu(u_e^{\nu-1} - u_0^\nu)}{(1+\beta_p^\nu)} \text{ and } v_{cg}^\nu - v_0^\nu = \frac{v_2^\nu - v_0^\nu + \beta_d^\nu(u_e^{\nu-1} - v_0^\nu)}{(1+\beta_d^\nu)}.$$

Hence, the search direction vector in the primal is, in fact, a convex combination of the P-projection of $-\nabla_P f(u_0)$ and the search direction vector in the previous primal

iteration. Similarly, the search direction vector in the dual is a convex combination of the Q-projection of $\nabla_Q g(v_0)$ and the search direction vector in the previous dual iteration.

We shall prove in Theorem 4.5 that under the critical face condition, the primal iterations in Algorithm 2 reduce in a vicinity of the optimal solution to $(\mathcal{P})$ to the execution of the Hestenes-Stiefel conjugate gradient method if the critical face U_0 is eventually reached by the primal iterates, and similarly for the dual iterations. From this we will obtain a termination property for Algorithm 2, which will be invoked by an interactive restart of the algorithm.

Algorithm 2 departs a bit from the philosophy of Algorithm 1 in utilizing unprojected gradients in (2.1) and (2.5) instead of just projected gradients. These unprojected gradients are available through (1.11) and (1.12) (also (1.15) or (1.16)), and for multistage optimization problems in the pattern laid out in [7] they can still be calculated without having to invoke the gigantic R matrix. An earlier version of Algorithm 2 that we worked with did use the projected gradients exclusively, and it performed similarly, but there were technical difficulties in establishing a finite termination property. Future research may shed more light on this issue. The same can be said of another small departure in Algorithm 2 from the philosophy one might hope maintain in a "conjugate gradient" method: the introduction on occasion of step sizes possibly greater than 1 relative to $[u_0^\nu, u_{cg}^\nu]$ or $[v_0^\nu, v_{cg}^\nu]$ (although not, of course, relative to the designated intervals $[u_0^\nu, u_e^\nu]$ or $[v_0^\nu, v_e^\nu]$) through the second alternatives in (2.4) or (2.8).

3. Global Convergence and Local Quadratic Structure. This section establishes some basic convergence properties of Algorithms 0, 1 and 2. It also reveals the special quadratic structure in $(\mathcal{P})$ and $(\mathcal{Q})$ around the optimal solutions $\bar{u}$ and $\bar{v}$ in the case where the critical face condition is satisfied, which will be utilized in further convergence analysis in Section 5.

PROPOSITION 3.1. (Feasible descent and ascent.)

(a) *In Algorithm 0 (hence also in Algorithms 1 and 2) the vector $u_2^\nu - u_0^\nu$ gives a feasible descent direction for the primal objective function f at u_0^ν (unless $u_2^\nu - u_0^\nu = 0$, in which case $u_0^\nu = \bar{u}$). Similarly, the vector $v_2^\nu - v_0^\nu$ gives a feasible ascent direction for the dual objective function g at v_0^ν (unless $v_2^\nu - v_0^\nu = 0$, in which case $v_0^\nu = \bar{v}$).*

(b) *In Algorithm 2, the vector $u_{cg}^\nu - u_0^\nu$ gives a feasible descent direction for the primal objective f at u_0^ν unless $u_0^\nu = \bar{u}$. Similarly, the vector $v_{cg}^\nu - v_0^\nu$ gives a feasible ascent direction for the dual objective g at v_0^ν unless $v_0^\nu = \bar{v}$. Thus, Algorithm 2 is well defined in the sense that, regardless of the type of iteration, as long as it does not terminate in optimality, the vector $u_e^\nu - u_0^\nu$ gives a feasible descent direction at u_0^ν in the primal while the vector $v_e^\nu - v_0^\nu$ gives a feasible ascent direction at v_0^ν in the dual.*

Proof. (a) We know that u_2^ν minimizes $L(u, v_1^\nu)$ over $u \in U$, where $L(u, v_1^\nu)$ is given by formula (1.17). We obtain from this formula that unless $u_2^\nu = u_0^\nu$, implying u_0^ν is optimal for the primal, we must have $\nabla f(u_0^\nu)\cdot(u_2^\nu - u_0^\nu) < 0$. Descent in this direction is feasible because the line segment $[u_0^\nu, u_2^\nu]$ is included in U by convexity. The proof of the dual part is parallel.

(b) The argument is by induction. From the optimality test in Step 3 we see that the algorithm will terminate at $(\bar{u}, \bar{v})$ if either $u_0^\nu = \bar{u}$ in the primal or $v_0^\nu = \bar{v}$ in the dual. (For instance, if $u_0^\nu = \bar{u}$, then $v_1^\nu = \bar{v}$, so that $f(\hat{u}) - g(\hat{v}) = 0$.) Suppose neither u_0^ν nor v_0^ν is optimal. Proposition 3.1(a) covers our claims for the initial iteration of each primal or dual cycle. Suppose that the claims are true for iteration $l-1$ of a primal cycle, $0 < l < k$, this corresponding to iteration $\nu - 1$ of the algorithm as a

whole. We have $(u_2^\nu - u_0^\nu)\cdot\nabla f(u_0^\nu) < 0$ by part (a) and $(u_e^{\nu-1} - u_0^\nu)\cdot\nabla f(u_0^\nu) \le 0$ through the line search. (Note that we get this inequality instead of an equation because the search is over a segment rather than a half-line; the minimizing point could be at the end of the segment.) Hence

$$(u_{cg}^\nu - u_0^\nu)\cdot\nabla f(u_0^\nu) = \frac{(u_2^\nu - u_0^\nu)\cdot\nabla f(u_0^\nu) + \beta_p^\nu (u_e^{\nu-1} - u_0^\nu)\cdot\nabla f(u_0^\nu)}{1+\beta_p^\nu} < 0.$$

Therefore, the vector $u_{cg}^\nu - u_0^\nu \neq 0$ gives a descent direction, so the segment L_p^ν in (2.4) is nontrivial. From (2.3), we see further that u_{cg}^ν is a convex combination of two feasible points $u_2^\nu \in U$ and $u_e^{\nu-1} \in U$. Hence the point u_{cg}^ν is feasible, i.e., $u_{cg}^\nu \in U$, and the direction of $u_{cg}^\nu - u_0^\nu$ is a feasible direction in the primal at u_0^ν. The vector $u_e^\nu - u_0^\nu$ therefore gives a feasible descent direction for f at u_0^ν, since it results from a scaling of the vector $u_{cg}^\nu - u_0^\nu$. Iteration l of the primal cycle thus again satisfies the claim. The case of dual cycles is handled similarly. □

THEOREM 3.2. (Global convergence.) *In Algorithm 0 (hence also in Algorithms 1 and 2) with optimality threshold $\varepsilon > 0$, termination must come with ε-optimal solutions $\hat{u}$ and $\hat{v}$ in just a finite number of iterations. With $\varepsilon = 0$, unless the procedure happens to terminate with the exact optimal solutions $\bar{u}$ and $\bar{v}$ in a finite number of iterations, the sequences generated will be such that $u_0^\nu \to \bar{u}$ and $v_0^\nu \to \bar{v}$ as $\nu \to \infty$. Furthermore, then $u_1^\nu \to \bar{u}$ and $u_2^\nu \to \bar{u}$, as well as $v_1^\nu \to \bar{v}$ and $v_2^\nu \to \bar{v}$.*

Proof. Consider first the case where $\varepsilon = 0$. From Proposition 1.4, the point $u_2 = G\big(F(u_0)\big)$ depends continuously on u_0. Denote by $\mathcal{D}$ the continuous mapping $u_0 \mapsto (u_0, u_2 - u_0)$ from U to $U \times \mathbb{R}^n$. Let $\mathcal{M} : U \times \mathbb{R}^n \to U$ be the line search mapping defined by

$$\mathcal{M}(u_0, d) = \operatorname*{argmin}_{u \in [u_0, u_0 + d]} f(u).$$

The mapping $\mathcal{M}$ is closed at the point (u_0, d) with $d \neq 0$, cf. [21, Theorem 8.3.1]. Now by Proposition 3.1(a), $u_2 - u_0 \neq 0$ for $u_0 \neq \bar{u}$. Hence the composite mapping $\mathcal{M}\circ\mathcal{D}$ is closed on $U \setminus \{\bar{u}\}$, cf. [21, Theorem 7.3.2]. Define

$$\mathcal{A} = \mathcal{B}\circ\mathcal{M}\circ\mathcal{D},$$

where $\mathcal{B} : U \rightrightarrows U$ is the point-to-set mapping $\mathcal{B}(u) = \{u' \in U \mid f(u') \le f(u)\}$. Note that the sequence $\{f(u_0^\nu)\}$ is nonincreasing. Now let $\mathcal{K}_p$ be the sequence consists of the indices of those iterations in which a line search on $[u_0^\nu, u_2^\nu]$ is performed for the primal objective function. Then $\mathcal{K}_p$ is an infinite subsequence of $\{\nu\}$ unless the procedure happens to terminate with the exact optimal solutions $\bar{u}$ and $\bar{v}$ in a finite number of iterations. Let ν'' and ν' be two consecutive elements in $\mathcal{K}_p$ with $\nu'' > \nu'$. Then we can write

$$u_0^{\nu''} \in \mathcal{A}(u_0^{\nu'}).$$

By Proposition 3.1, moreover, the vector $u_2 - u_0$ is a descent direction for the primal objective $f(u)$ at u_0 unless u_0 is already optimal. Since we are in the fully quadratic case, the set $\{u \in \mathbb{R}^n \mid f(u) \le f(u_0^0)\}$ is compact, and the optimal solution $\bar{u}$ for problem $(\mathcal{P})$ is unique. It follows then that $u_0^\nu \to \bar{u}$ as $\nu \to \infty$, $\nu \in \mathcal{K}_p$, cf. [21, Theorem 7.3.4]. Therefore $f(u_0^\nu) \to f(\bar{u})$ as $\nu \to \infty$, which in turn implies $u_0^\nu \to \bar{u}$ as $\nu \to \infty$ since f is strongly convex (Theorem 1.1).

For analogous reasons, $v_0^\nu \to \bar{v}$. Then since $u_1^\nu = G(v_0^\nu)$ and $\bar{u} = G(\bar{v})$ with the mapping G continuous (Proposition 1.4), we have $u_1^\nu \to \bar{u}$. Likewise, $v_1^\nu \to \bar{v}$. The

argument can be applied then again: we have $u_2^\nu = G(v_1^\nu)$, so $u_2^\nu \to \bar{u}$ and in parallel fashion $v_2^\nu \to \bar{v}$.

In particular, we have $f(u_0^\nu) - g(v_0^\nu) \to f(\bar{u}) - g(\bar{v}) = 0$ because f and g are continuous (Theorems 1.1 and 1.2(a)). In the case where $\varepsilon > 0$, this guarantees termination in finitely many iterations. □

COROLLARY 3.3. (Points in the critical faces.) *The sequences generated by Algorithm 0 have the property that eventually u_1^ν and u_2^ν belong to the primal critical face U_0, while v_1^ν and v_2^ν belong to the dual critical face V_0.*

Proof. This follows via Proposition 1.8. □

COROLLARY 3.4. (A special case of finite termination.) *If $\varepsilon = 0$ and either of the critical faces U_0 or V_0 consists of just a single point, Algorithm 0 (and therefore also Algorithms 1 and 2) will terminate at the optimal solution pair $(\bar{u}, \bar{v})$ after a finite number of iterations.*

Proof. When U_0 consists of the single point $\bar{u}$, we have by Corollary 3.3 that $u_2^\nu = \bar{u}$ for all sufficiently large ν. Once this is the situation, the line search in the first iteration of the next primal cycle will yield $\bar{u}$. On returning to Step 1 for the succeeding iteration, $\bar{v}$ will be generated as $F(\bar{u})$, and termination must then come in Step 3. The situation is analogous when V_0 consists of just $\bar{v}$. □

A companion result to Corollary 3.3 is the following.

PROPOSITION 3.5. (Convergence onto critical faces.) *Let $\{u_0^\nu\}$ and $\{v_0^\nu\}$ be sequences generated by Algorithm 1 or Algorithm 2. Then for the primal critical face U_0, we have either $u_0^\nu \in U_0$ for all sufficiently large ν or $u_0^\nu \notin U_0$ for all sufficiently large ν. Similarly, for the dual critical face V_0 we have either $v_0^\nu \in V_0$ for all sufficiently large ν or $v_0^\nu \notin V_0$ for all sufficiently large ν.*

Proof. We prove the primal part. The proof of the dual part is similar. Observe that $\hat{v}_0^\nu \to \bar{v}$ as $v_0^\nu \to \bar{v}$ in the algorithm. Hence by Proposition 1.8, we have $\hat{u}_1^\nu \in U_0$ as well as $u_2^\nu \in U_0$ for sufficiently large ν. Then in Algorithm 1 we have

$$u_0^\nu \in U_0 \quad \Rightarrow \quad [u_0^\nu, u_2^\nu] \subset U_0 \quad \Rightarrow \quad \hat{u}_0^{\nu+1} \in U_0 \quad \Rightarrow \quad u_0^{\nu+1} \in U_0$$

since $u_0^{\nu+1}$ is defined either as $\hat{u}_0^{\nu+1}$ or as $\hat{u}_1^{\nu+1}$. From this it is apparent that our assertion is valid in the case of sequences generated by Algorithm 1.

For Algorithm 2, we claim that for sufficiently large ν we have $u_e^\nu \in U_0$ when $u_0^\nu \in U_0$. For if $u_e^\nu = u_2^\nu$, we certainly have $u_e^\nu = u_2^\nu \in U_0$. If $u_e^\nu \neq u_2^\nu$, then u_{cg}^ν is a convex combination of $u_e^{\nu-1}$ and $u_2^\nu \in U_0$, and there is no interactive restart in iteration ν, i.e., $\hat{u}_0^\nu = u_0^\nu \in U_0$. Now $\hat{u}_0^\nu \neq u_0^{\nu-1}$ by Proposition 3.1(b). Hence we have either $\hat{u}_0^\nu = u_e^{\nu-1}$ which implies $u_e^{\nu-1} \in U_0$, or $\hat{u}_0^\nu \in \mathrm{ri}[u_0^{\nu-1}, u_e^{\nu-1}]$, which also implies $u_e^{\nu-1} \in U_0$ since U_0 is a face of U. Then $u_{cg}^\nu \in U_0$, and by the definition of u_e^ν in the algorithm we have $u_e^\nu \in U_0$. Therefore

$$u_0^\nu \in U_0 \quad \Rightarrow \quad [u_0^\nu, u_e^\nu] \subset U_0 \quad \Rightarrow \quad \hat{u}_0^{\nu+1} \in U_0 \quad \Rightarrow \quad u_0^{\nu+1} \in U_0$$

for sufficiently large ν. Thus, our assertion is valid also in the case of sequences generated by Algorithm 2. □

REMARK. With the aid of the concept of an *ultimate quadratic region* introduced later in Definition 3.7, it will be seen that when the critical face condition is satisfied, the assertion of the proposition can be written as follows: after the sequences $\{u_0^\nu\}$ and $\{v_0^\nu\}$ have entered an ultimate quadratic region, once $u_0^{\nu'} \in U_0$ for some ν', then $u_0^\nu \in U_0$ for all $\nu \geq \nu'$; and similarly once $v_0^{\nu''} \in V_0$ for some ν'', then $v_0^\nu \in V_0$ for all $\nu \geq \nu''$.

For Algorithm 2, broader results on finite termination than the one in Corollary 3.4 will be obtained when the critical face condition is satisfied through reduction to a simpler quadratic structure which is identified as governing in a neighborhood of the solution. This local structure will also be the basis for developing convergence rates for Algorithms 1 and 2 in cases without finite termination. In developing it in the next theorem, we recall the notion of the affine hull $\operatorname{aff} C$ of a convex set C: this is the smallest affine set that includes C, or equivalently, the intersection of all the hyperplanes that include C [17].

THEOREM 3.6. (Quadratic structure near optimality.) *Suppose the critical face condition is satisfied. Then f is quadratic in some neighborhood of $\bar{u}$, while g is quadratic in some neighborhood of $\bar{v}$. Furthermore, for points $u_0 \in U$ and $v_0 \in V$ sufficiently close to $\bar{u}$ and $\bar{v}$, the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is the same as that on $\operatorname{aff} U_0 - u_0$, while the Q-projection of $\nabla_Q g(v_0)$ on $V - v_0$ is the same as that on $\operatorname{aff} V_0 - v_0$.*

Proof. Since by Proposition 1.8 the point $v_1 = F(u_0)$ lies in the critical face V_0 when u_0 is sufficiently close to $\bar{u}$, we have

$$\max_{v\in V}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\} = \max_{v\in V_0}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\}. \tag{3.1}$$

The mapping F is continuous (Proposition 1.4) and $\bar{v} \in \operatorname{ri} V_0$ by assumption, so we have $v_1 \in \operatorname{ri} V_0$ when u_0 is sufficiently close to $\bar{u}$. Then (3.1) can further be written instead as

$$\max_{v\in V}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\} = \max_{v\in \operatorname{aff} V_0}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\}.$$

Locally, therefore,

$$f(u) = p\cdot u + \tfrac{1}{2}u\cdot Pu + \max_{v\in \operatorname{aff} V_0}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\}. \tag{3.2}$$

Similarly, for v in some neighborhood of $\bar{v}$ we have

$$g(v) = q\cdot v - \tfrac{1}{2}v\cdot Qv + \min_{u\in \operatorname{aff} U_0}\{u\cdot(p - R^T v) + \tfrac{1}{2}u\cdot Pu\}. \tag{3.3}$$

The set $\operatorname{aff} V_0$, because it is affine and contains $\bar{v}$, has the form $\bar{v} + S$ for a certain subspace S of $\mathbb{R}^m$, which in turn can be written as the set of all vectors of the form $v' = Dw$ for a certain $m \times d$ matrix D of rank d (the dimension of S). In substituting $v = \bar{v} + Dw$ in (3.2) and taking the maximum instead over all $w \in \mathbb{R}^d$, we see through elementary calculus and linear algebra that the maximum value is a quadratic function of u. This establishes that $f(u)$ is quadratic in u on a neighborhood of $\bar{u}$. The same argument can be pursued in (3.3) to verify that $g(v)$ is quadratic around $\bar{v}$.

Next we consider the projected gradients. According to Proposition 1.6, the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is the vector $u_2 - u_0$, where $u_2 = G\big(F(u_0)\big)$. When u_0 is close enough to $\bar{u}$ in U, u_2 belongs by Proposition 1.8 to $\operatorname{ri} U_0$, which is the interior of U_0 relative to $\operatorname{aff} U_0$. Thus, for u_0 in some neighborhood of $\bar{u}$ in U_0 the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ belongs to the relatively open convex subset $\operatorname{ri} U_0 - u_0$ of $U - u_0$ and must be the same as the projection on this subset or on $U_0 - u_0$ itself. When the nearest point of a convex set C belongs to $\operatorname{ri} C$, it is the same the nearest point of $\operatorname{aff} C$. The P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is therefore the

same as the P-projection of $-\nabla_P f(u_0)$ on $\operatorname{aff} U_0 - u_0$. The Q-projection of $\nabla_Q g(v_0)$ on $V - v_0$ is analyzed in parallel fashion. □

Theorem 3.6 together with Proposition 1.8 makes it possible for us to concentrate our analysis of the terminal behavior of our algorithms, in the case of optimality threshold $\varepsilon = 0$, on regions around $(\bar{u}, \bar{v})$ of the following special kind.

DEFINITION 3.7. (*Ultimate quadratic regions.*) By an *ultimate quadratic region* for problems $(\mathcal{P})$ and $(\mathcal{Q})$ when the critical face condition is satisfied, we shall mean an open convex neighborhood $U^* \times V^*$ of $(\bar{u}, \bar{v})$ with the properties that

(a) $U^* \cap U_0 = U^* \cap \operatorname{ri} U_0$ and $V^* \cap V_0 = V^* \cap \operatorname{ri} V_0$,

(b) f is quadratic on U^* and g is quadratic on V^*,

(c) for all $u_0 \in U^* \cap U$ the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is that on $(\operatorname{aff} U_0) - u_0$, while for all $v_0 \in V^* \cap V$ the Q-projection of $\nabla_Q g(v_0)$ on $V - v_0$ is that on $(\operatorname{aff} V_0) - v_0$,

(d) for all $u_0 \in U^* \cap U$ and $v_0 \in V^* \cap V$ the points $u_1 = G(v_0)$, $v_1 = F(u_0)$, $u_2 = G(v_1)$ and $v_2 = F(u_1)$ are such that u_1 and u_2 belong to $\operatorname{ri} U_0$, while v_1 and v_2 belong to $\operatorname{ri} V_0$.

Here we recognize that the affine sets $\operatorname{aff} U_0$ and $\operatorname{aff} V_0$ are translates of certain subspaces, which in fact are the sets $(\operatorname{aff} U_0) - \bar{u}$ and $(\operatorname{aff} V_0) - \bar{v}$. The projections in (c) of this definition can be described also in terms of these subspaces. Let

$$\begin{aligned} S_p &= P\text{-projection mapping onto the subspace } (\operatorname{aff} U_0) - \bar{u}, \\ S_d &= Q\text{-projection mapping onto the subspace } (\operatorname{aff} V_0) - \bar{v}, \\ S_p^\perp &= I - S_p, \qquad S_d^\perp = I - S_d. \end{aligned} \tag{3.4}$$

The mapping $S_p^\perp$ projects onto the subspace of $\mathbb{R}^n$ that is orthogonally complementary to $(\operatorname{aff} U_0) - \bar{u}$ with respect to the P-inner product in (1.10), while the mapping $S_d^\perp$ projects onto the subspace of $\mathbb{R}^m$ that is orthogonally complementary to $(\operatorname{aff} V_0) - \bar{v}$ with respect to the Q-inner product. All these projections are linear transformations, of course.

PROPOSITION 3.8. (Projection decomposition.) *For (u_0, v_0) in an ultimate quadratic region $U^* \times V^*$, one has for $u_2 := G(F(u_0))$ and $v_2 := F(G(v_0))$ that*

$$\begin{aligned} u_2 - u_0 &= S_p\big(-\nabla_P f(u_0)\big) - S_p^\perp(u_0 - \bar{u}) = -S_p\big(\nabla_P^2 f(\bar{u})(u_0 - \bar{u})\big) - S_p^\perp(u_0 - \bar{u}), \\ v_2 - v_0 &= S_d\big(\nabla_Q g(v_0)\big) - S_d^\perp(v_0 - \bar{v}) = S_d\big(\nabla_Q^2 g(\bar{v})(v_0 - \bar{v})\big) - S_d^\perp(v_0 - \bar{v}). \end{aligned}$$

Proof. The P-projection of $-\nabla_P f(u_0)$ on $(\operatorname{aff} U_0) - u_0$ can be realized by taking the P-projection of $-\nabla_P f(u_0) + (u_0 - \bar{u})$ on the set $(\operatorname{aff} U_0) - u_0 + (u_0 - \bar{u})$ and then subtracting $(u_0 - \bar{u})$. Therefore, in a region with property (c) of Definition 3.7 we have by (1.17) in Proposition 1.6 that

$$u_2 - u_0 = S_p\big(-\nabla_P f(u_0) + (u_0 - \bar{u})\big) - (u_0 - \bar{u}) = S_p\big(-\nabla_P f(u_0)\big) - (I - S_p)(u_0 - \bar{u}),$$

which is the first equality asserted. The second equality comes from having $\nabla_P f(u_0) = \nabla_P f(\bar{u}) + \nabla_P^2 f(\bar{u})(u_0 - \bar{u})$ (since f is quadratic in the region in question), and $S_p\big(\nabla_P f(\bar{u})\big) = 0$ by the optimality of $\bar{u}$. The proof of the dual equalities is along the same lines. □

4. Rate of Convergence. In taking advantage of the existence of an ultimate quadratic region, we shall utilize in our technical arguments a change of variables that will make a number of basic properties clearer. This change of variables amounts

to the introduction of orthonormal coordinate systems relative to the inner products naturally associated with our problems, namely $\langle\cdot,\cdot\rangle_P$ on $\mathbb{R}^n$ and $\langle\cdot,\cdot\rangle_Q$ on $\mathbb{R}^m$, as given in (1.10). The coordinate systems are introduced in such a way that the subspaces $(\operatorname{aff} U_0)-\bar{u}$ and $(\operatorname{aff} V_0)-\bar{v}$ for the projections in (3.4) and Proposition 3.8 take a very simple form.

Let W be an $n\times n$ orthogonal matrix and Z an $m\times m$ orthogonal matrix. Our shift will be from u and v to $\tilde{u}=WP^{\frac{1}{2}}u$ and $\tilde{v}=ZQ^{\frac{1}{2}}v$. In these variables and with

$$\tilde{U}=WP^{\frac{1}{2}}U,\quad \tilde{V}=ZQ^{\frac{1}{2}}V,$$

our primal and dual problems take the form

$$(\tilde{\mathcal{P}})\qquad \text{minimize } \tilde{f}(\tilde{u}) \text{ over all } \tilde{u}\in\tilde{U},$$
$$(\tilde{\mathcal{Q}})\qquad \text{maximize } \tilde{g}(\tilde{v}) \text{ over all } \tilde{v}\in\tilde{V},$$

where we have

$$\tilde{f}(\tilde{u})=\sup_{\tilde{v}\in\tilde{V}}\tilde{L}(\tilde{u},\tilde{v})\quad\text{and}\quad \tilde{g}(\tilde{v})=\inf_{\tilde{u}\in\tilde{U}}\tilde{L}(\tilde{u},\tilde{v}),\tag{4.1}$$

$$\tilde{F}(\tilde{u})=\operatorname*{argmax}_{\tilde{v}\in\tilde{V}}\tilde{L}(\tilde{u},\tilde{v})\quad\text{and}\quad \tilde{G}(\tilde{v})=\operatorname*{argmin}_{\tilde{u}\in\tilde{U}}\tilde{L}(\tilde{u},\tilde{v}),\tag{4.2}$$

in the notation that

$$\tilde{L}(\tilde{u},\tilde{v})=\tilde{p}\cdot\tilde{u}+\tfrac{1}{2}\|\tilde{u}\|^2+\tilde{q}\cdot\tilde{v}-\tfrac{1}{2}\|\tilde{v}\|^2-\tilde{v}\cdot\tilde{R}\tilde{u}\ \text{ on }\ \tilde{U}\times\tilde{V},\tag{4.3}$$

$$\tilde{p}=WP^{-\frac{1}{2}}p,\qquad \tilde{q}=ZQ^{-\frac{1}{2}}q,\qquad \tilde{R}=ZQ^{-\frac{1}{2}}RP^{-\frac{1}{2}}W^T.\tag{4.4}$$

The optimal solutions $\bar{u}$ and $\bar{u}$ to $(\mathcal{P})$ and $(\mathcal{Q})$ translate into optimal solutions $\bar{\tilde{u}}$ and $\bar{\tilde{v}}$ to $(\tilde{\mathcal{P}})$ and $(\tilde{\mathcal{Q}})$, namely

$$\bar{\tilde{u}}=WP^{\frac{1}{2}}\bar{u}\ \text{ and }\ \bar{\tilde{v}}=ZQ^{\frac{1}{2}}\bar{v}.\tag{4.5}$$

Let d_1 be the dimension of the subspace $(\operatorname{aff} U_0)-\bar{u}$ and d_2 the dimension of the subspace $(\operatorname{aff} V_0)-\bar{v}$. We choose W such that, in the new coordinates corresponding to the components of $\tilde{u}$, the set $WP^{\frac{1}{2}}(\operatorname{aff} U_0-\bar{u})=\operatorname{aff}\tilde{U}_0-\bar{\tilde{u}}$ is the subspace spanned by the first d_1 columns of I_n. Likewise, we choose Z such that in the $\tilde{v}$ coordinates the set $ZQ^{\frac{1}{2}}(\operatorname{aff} V_0-\bar{v})=\operatorname{aff}\tilde{V}_0-\bar{\tilde{v}}$ is the subspace spanned by the first d_2 columns of I_m. We partition the vectors $\tilde{u}\in\mathbb{R}^n$ and $\tilde{v}\in\mathbb{R}^m$ into

$$\tilde{u}=\begin{pmatrix}\tilde{u}_f\\ \tilde{u}_r\end{pmatrix}\ \text{ and }\ \tilde{v}=\begin{pmatrix}\tilde{v}_f\\ \tilde{v}_r\end{pmatrix},\tag{4.6}$$

where $\tilde{u}_f$ consists first d_1 components of $\tilde{u}$ and $\tilde{v}_f$ consists first d_2 components of $\tilde{v}$. (Here u_f is the "free" part of $\tilde{u}$, relative to $(\operatorname{aff} U_0)-\bar{u}$ being the subspace that indicates the remaining degrees of freedom in the tail of our convergence analysis when

the critical face condition is satisfied, whereas u_r is the "restricted" part of $\tilde{u}$.) The projection mappings S_p, $S_p^\perp$, S_d, and $S_d^\perp$ reduce in this way to the simple projections

$$\begin{aligned} &\tilde{S}_p : \begin{pmatrix} \tilde{u}_f \\ \tilde{u}_r \end{pmatrix} \mapsto \begin{pmatrix} \tilde{u}_f \\ 0 \end{pmatrix}, \qquad \tilde{S}_p^\perp : \begin{pmatrix} \tilde{u}_f \\ \tilde{u}_r \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ \tilde{u}_r \end{pmatrix}, \\ &\tilde{S}_d : \begin{pmatrix} \tilde{v}_f \\ \tilde{v}_r \end{pmatrix} \mapsto \begin{pmatrix} \tilde{v}_f \\ 0 \end{pmatrix}, \qquad \tilde{S}_d^\perp : \begin{pmatrix} \tilde{v}_f \\ \tilde{v}_r \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ \tilde{v}_r \end{pmatrix}. \end{aligned} \tag{4.7}$$

We partition the columns of the matrix $\tilde{R}$ in accordance with $\tilde{u}$ and the rows in accordance with $\tilde{v}$. Thus,

$$\tilde{R} = \begin{pmatrix} \tilde{R}_{ff} & \tilde{R}_{fr} \\ \tilde{R}_{rf} & \tilde{R}_{rr} \end{pmatrix}. \tag{4.8}$$

In this notation the primal objective function in the transformed problem $(\tilde{\mathcal{P}})$ takes, in an ultimate quadratic region, the simple form

$$\begin{aligned} &\tilde{f}(\tilde{u}) = \tfrac{1}{2}(\tilde{u} - \tilde{u}^*)\cdot A(\tilde{u} - \tilde{u}^*) + \text{const.} \quad \text{for some } \tilde{u}^*, \text{ where} \\ &A := I + \begin{pmatrix} \tilde{R}_{ff} & \tilde{R}_{fr} \end{pmatrix}^T \begin{pmatrix} \tilde{R}_{ff} & \tilde{R}_{fr} \end{pmatrix}, \end{aligned} \tag{4.9}$$

while in the dual problem one similarly has

$$\begin{aligned} &\tilde{g}(\tilde{v}) = -\tfrac{1}{2}(\tilde{v} - v^*)\cdot B(\tilde{v} - \tilde{v}^*) + \text{const.} \quad \text{for some } \tilde{v}^*, \text{ where} \\ &B := I + \begin{pmatrix} \tilde{R}_{ff} \\ \tilde{R}_{rf} \end{pmatrix} \begin{pmatrix} \tilde{R}_{ff} \\ \tilde{R}_{rf} \end{pmatrix}^T. \end{aligned} \tag{4.10}$$

In fact, in the notation (4.5) and with $\tilde{U}_0$ and $\tilde{V}_0$ denoting the critical faces $WP^{\frac{1}{2}}U_0$ and $ZQ^{\frac{1}{2}}V_0$ in the transformed problems, one has the expansions

$$\tilde{f}(\tilde{u}) = \tilde{f}(\bar{\tilde{u}}) + \tfrac{1}{2}(\tilde{u}_f - \bar{\tilde{u}}_f)\cdot\Big(I + \tilde{R}_{ff}^T\tilde{R}_{ff}\Big)(\tilde{u}_f - \bar{\tilde{u}}_f) \text{ for } \tilde{u} \in \operatorname{aff}\tilde{U}_0, \tag{4.11}$$

$$\tilde{g}(\tilde{v}) = \tilde{g}(\bar{\tilde{v}}) - \tfrac{1}{2}(\tilde{v}_f - \bar{\tilde{v}}_f)\cdot\Big(I + \tilde{R}_{ff}\tilde{R}_{ff}^T\Big)(v_f - \bar{\tilde{v}}_f) \text{ for } \tilde{v} \in \operatorname{aff}\tilde{V}_0. \tag{4.12}$$

It will be helpful to write the Hessian matrices A and B in (4.9) and (4.10) as

$$A = \begin{bmatrix} A_{ff} & A_{fr} \\ A_{rf} & A_{rr} \end{bmatrix} = \begin{bmatrix} I + \tilde{R}_{ff}^T\tilde{R}_{ff} & \tilde{R}_{ff}^T\tilde{R}_{fr}, \\ \tilde{R}_{fr}^T\tilde{R}_{ff} & I + \tilde{R}_{fr}^T\tilde{R}_{fr} \end{bmatrix}, \tag{4.13}$$

$$B = \begin{bmatrix} B_{ff} & B_{fr} \\ B_{rf} & B_{rr} \end{bmatrix} = \begin{bmatrix} I + \tilde{R}_{ff}\tilde{R}_{ff}^T & \tilde{R}_{ff}\tilde{R}_{rf}^T \\ \tilde{R}_{rf}\tilde{R}_{ff}^T & I + \tilde{R}_{rf}\tilde{R}_{rf}^T \end{bmatrix}. \tag{4.14}$$

A crucial property of our change of variables $\tilde{u} = WP^{\frac{1}{2}}u$ and $\tilde{v} = ZQ^{\frac{1}{2}}v$ is that

$$\|\tilde{u}\| = \|u\|_P \text{ and } \|\tilde{v}\| = \|v\|_Q,$$

and accordingly

$$\|\nabla \tilde{f}(\tilde{u})\| = \|\nabla_P f(u)\|_P \text{ and } \|\nabla \tilde{g}(\tilde{v})\| = \|\nabla_Q g(v)\|_Q,$$
$$\|\nabla^2 \tilde{f}(\tilde{u})\| = \|\nabla^2_P f(u)\|_P \text{ and } \|\nabla^2 \tilde{g}(\tilde{v})\| = \|\nabla^2_Q g(v)\|_Q.$$

The following result is a strengthening of Proposition 3.1 in the sense that it gives a quantitative estimate for the relationship between $\|u_0 - u_2\|_P$ and $\|u_0 - \bar{u}\|_P$ in primal, and between $\|v_0 - v_2\|_Q$ and $\|v_0 - \bar{v}\|_Q$ in the dual.

PROPOSITION 4.1. (Norm estimates.) *Suppose the critical face condition is satisfied. Then for u_0 and v_0 in an ultimate quadratic region for problems $(\mathcal{P})$ and $(\mathcal{Q})$, and with $u_2 := G(F(u_0))$ and $v_2 := F(G(v_0))$, one has*

$$(4.15)\quad (5+4\|\nabla^2_P f(\bar{u})\|^2_P)^{-\frac{1}{2}}\|u_0 - \bar{u}\|_P \le \|u_0 - u_2\|_P \le (1+\|\nabla^2_P f(\bar{u})\|^2_P)^{\frac{1}{2}}\|u_0 - \bar{u}\|_P,$$

$$(4.16)\quad (5+4\|\nabla^2_Q g(\bar{v})\|^2_Q)^{-\frac{1}{2}}\|v_0 - \bar{v}\|_Q \le \|v_0 - v_2\|_Q \le (1+\|\nabla^2_Q g(\bar{v})\|^2_Q)^{\frac{1}{2}}\|v_0 - \bar{v}\|_Q.$$

Proof. In the transformed coordinates the first equation in Proposition 3.8 gives us $\tilde{u}_2 - \tilde{u}_0 = -\tilde{S}_p\big(\nabla^2 \tilde{f}(\bar{\tilde{u}})(\tilde{u}_0 - \bar{\tilde{u}})\big) - \tilde{S}_P^{\perp}(\tilde{u}_0 - \bar{\tilde{u}})$. In the notation (4.13) for $\nabla^2 f(\bar{\tilde{u}})$ this gives

$$\begin{aligned}\|\tilde{u}_0 - \tilde{u}_2\|^2 &= \|\tilde{S}_p\big(A(\tilde{u}_0 - \bar{\tilde{u}})\big)\|^2 + \|\tilde{S}_p^{\perp}(\tilde{u}_0 - \bar{\tilde{u}})\|^2 \\ &\le \|A(\tilde{u}_0 - \bar{\tilde{u}})\|^2 + \|\tilde{u}_0 - \bar{\tilde{u}}\|^2 \le (\|A\|^2 + 1)\,\|\tilde{u}_0 - \bar{\tilde{u}}\|^2.\end{aligned}$$

This gives the right half of (4.15). To get the left half, decompose $\tilde{u}_0 - \bar{\tilde{u}}$ into $\mu_1\xi + \mu_2\eta$, where ξ is a unit vector in the null space of $(A_{ff}\ A_{fr})$ while η is a unit vector in the orthogonal complement of that null space, and the direction of η is so chosen that $\mu_2 > 0$. Partition ξ and η as well:

$$\xi = \begin{pmatrix}\xi_f \\ \xi_r\end{pmatrix}, \qquad \eta = \begin{pmatrix}\eta_f \\ \eta_r\end{pmatrix}.$$

It follows from $(A_{ff}\ A_{fr})\xi = A_{ff}\xi_f + A_{fr}\xi_r = 0$ that $\xi_f = -A_{ff}^{-1}A_{fr}\xi_r$ and

$$\|\xi_f\|^2 \le \|A_{ff}^{-1}\|^2\|A_{fr}\|^2\|\xi_r\|^2 \le \|A_{fr}\|^2\|\xi_r\|^2,$$

because the smallest eigenvalue of A_{ff} is no less than 1. Therefore

$$\|\xi\|^2 = \|\xi_f\|^2 + \|\xi_r\|^2 \le (1 + \|A_{fr}\|^2)\|\xi_r\|^2 \quad \Rightarrow \quad \|\xi_r\|^2 \ge \frac{1}{1+\|A_{fr}\|^2} \ge \frac{1}{1+\|A\|^2}.$$

Denote $\|\tilde{u}_0 - \bar{\tilde{u}}\|$ by κ. We get

$$\|\mu_1\xi_r + \mu_2\eta_r\| \ge \mu_1\|\xi_r\| - \mu_2\|\eta_r\| \ge (\kappa^2 - (\mu_2)^2)^{1/2}(1+\|A\|^2)^{-1/2} - \mu_2.$$

Recalling that all the eigenvalues of A_{ff} are no less than 1, we obtain

$$\begin{aligned}\|\tilde{u}_0 - \tilde{u}_2\|^2 &= \|(A_{ff}\ A_{fr})\mu_2\eta\|^2 + \|\mu_1\xi_r + \mu_2\eta_r\|^2 \\ &\ge \mu_2^2 + \big(\max\{0, (\kappa^2 - \mu_2^2)^{1/2}(1+\|A\|^2)^{-1/2} - \mu_2\}\big)^2.\end{aligned}$$

But the term $(\kappa^2-\mu_2^2)^{1/2}(1+\|A\|^2)^{-1/2}-\mu_2$ decreases monotonically as μ_2 increases from 0. This term equals $\bar{\mu}_2 := (5+4\|A\|^2)^{-1/2}\kappa$ when $\mu_2=\bar{\mu}_2$. Therefore $\|\tilde{u}_0-\tilde{u}_2\|^2 \geq (\bar{\mu}_2)^2$, from which the left half of (4.15) follows. The proof of (4.16) is similar. □

THEOREM 4.2. (Rate of convergence of PDSD.) *Consider Algorithm 1 in the case of threshold $\varepsilon=0$, and suppose the critical face condition is satisfied. In terms of $\gamma := \gamma(P,Q,R) := \|Q^{-\frac{1}{2}}RP^{-\frac{1}{2}}\|$, let*

$$c_1 := 1 - \frac{1}{(1+\gamma^2)\big[2+5(1+\gamma^2)+4(1+\gamma^2)^3\big]} < 1, \tag{4.17}$$

$$c_2 := \left(1-\frac{1}{1+\gamma^2/2}\right)^2 < 1. \tag{4.18}$$

Unless the algorithm actually terminates in a finite number of iterations with $(\hat{u},\hat{v})=(\bar{u},\bar{v})$, the sequences $\{f(u_0^\nu)\}$ and $\{g(v_0^\nu)\}$ generated by it converge linearly to the common optimal value $f(\bar{u})=g(\bar{v})$ in the sense that

$$\limsup_{\nu\to\infty}\frac{f(u_0^{\nu+1})-f(\bar{u})}{f(u_0^\nu)-f(\bar{u})} \leq c_1 \;\; \textit{and} \;\; \limsup_{\nu\to\infty}\frac{g(v_0^{\nu+1})-g(\bar{v})}{g(v_0^\nu)-g(\bar{v})} \leq c_1. \tag{4.19}$$

Moreover let $\bar{\nu}$ be an iteration number such that for $\nu\geq\bar{\nu}$ all the points u_0^ν, u_2^ν and v_0^ν, v_2^ν are in an ultimate quadratic region in Definition 3.7. Then once $u_0^{\nu'}\in U_0$ for some $\nu'\geq\bar{\nu}$ (as is sure to happen in an interactive primal restart at that stage) one has

$$\frac{f(u_0^{\nu+1})-f(\bar{u})}{f(u_0^\nu)-f(\bar{u})} \leq c_2 \quad \forall \nu\geq\nu', \tag{4.20}$$

and similarly, once $v_0^{\nu''}\in V_0$ for some $\nu''\geq\bar{\nu}$ (as is sure to happen in an interactive dual restart at that stage) one has

$$\frac{g(v_0^{\nu+1})-g(\bar{v})}{g(v_0^\nu)-g(\bar{v})} \leq c_2 \quad \forall \nu\geq\nu''. \tag{4.21}$$

Proof. Under the assumption that the algorithm does not terminate after a finite number of iterations at $(\bar{u},\bar{v})$, neither u_0^ν nor v_0^ν is optimal, as we have shown in the proof of Proposition 3.1(b).

Again we work in the transformed coordinates. Consider $\nu\geq\bar{\nu}$, i.e., the sequences $\{\tilde{u}_0^\nu\}$, $\{\tilde{u}_2^\nu\}$ and $\{\tilde{v}_0^\nu\}$, $\{\tilde{v}_2^\nu\}$ have entered the ultimate quadratic region. With respect to the direction vector $d^\nu := \tilde{u}_2^\nu-\tilde{u}_0^\nu$, the optimal step length $\bar{\lambda}^\nu$ for $\tilde{u}=\tilde{u}_0^\nu+\lambda d^\nu$ to minimize the quadratic form (4.9) over all $\lambda\in[0,\infty)$ can be written as

$$\begin{aligned}\bar{\lambda}^\nu &= \frac{-d^\nu\cdot A(\tilde{u}_0^\nu-\tilde{u}^*)}{d^\nu\cdot Ad^\nu} \\ &= \frac{\big[\tilde{S}_pA(\tilde{u}_0^\nu-u^*)+\tilde{S}_P^\perp(\tilde{u}_0^\nu-\bar{\tilde{u}})\big]\cdot A(\tilde{u}_0^\nu-\tilde{u}^*)}{\big[\tilde{S}_pA(\tilde{u}_0^\nu-\tilde{u}^*)+\tilde{S}_P^\perp(\tilde{u}_0^\nu-\bar{\tilde{u}})\big]\cdot A\big[\tilde{S}_pA(\tilde{u}_0^\nu-\tilde{u}^*)+\tilde{S}_P^\perp(\tilde{u}_0^\nu-\bar{\tilde{u}})\big]}.\end{aligned} \tag{4.22}$$

In the following, we first show that $\bar{\lambda}^\nu\leq 1$. Then the search on $[\tilde{u}_0^\nu,\tilde{u}_2^\nu]$ in Step 5 of the algorithm is equivalent to a search on the corresponding half-line (or is "perfect," for short), and there exist easy ways to estimate progress in the line search step. By Proposition 3.5 (cf. also the remark afterward), we have

Case 1: there exists some $\nu' \geq \bar{\nu}$ such that $\tilde{u}_0^\nu \in \tilde{U}_0$ for all $\nu \geq \nu'$, or
Case 2: $\tilde{u}_0^\nu \notin \tilde{U}_0$ for all $\nu \geq \bar{\nu}$.

In Case 1 the equation $\tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}}) = 0$ holds for all $\nu \geq \nu'$. Then it follows from (4.22) that

$$\bar{\lambda}^\nu = \frac{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot A(\tilde{u}_0^\nu - \tilde{u}^*)}{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot A\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)} = \frac{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot \tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)}{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot A\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)} \leq 1,$$

because all the eigenvalues of A are at least 1. Now Step 5 of the algorithm must coincide with the steepest descent method for $\tilde{f}$ on $\operatorname{aff}\tilde{U}_0$ with "perfect" line search, since $[\tilde{u}_0^\nu, \tilde{u}_2^\nu]$ is in an ultimate quadratic region of the problem. Note that all the eigenvalues of the Hessian matrix A_{ff} are in the interval $[1, 1 + \|\tilde{R}_{ff}\|^2]$, where $\|\tilde{R}_{ff}\|^2 \leq \|\tilde{R}\|^2 = \gamma^2$. Hence by using the expression of $\tilde{f}$ in (4.11), we have [22]

$$\frac{\tilde{f}(\hat{\tilde{u}}_0^{\nu+1}) - \tilde{f}(\bar{\tilde{u}})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})} \leq \left(\frac{\|\tilde{R}_{ff}\|^2}{\|\tilde{R}_{ff}\|^2 + 2}\right)^2 \leq \left(1 - \frac{1}{1 + \frac{1}{2}\|\tilde{R}\|^2}\right)^2, \tag{4.23}$$

which yields (4.20) since $\tilde{f}(\tilde{u}_0^{\nu+1}) \leq f(\hat{\tilde{u}}_0^{\nu+1})$ in the algorithm.

In Case 2 we have $\bar{\lambda}^\nu < 1$ for all $\nu \geq \bar{\nu}$, since otherwise $\tilde{u}_2^\nu$ would be taken as the next point $\tilde{u}_0^{\nu+1}$ and the iteration would be on the critical face $\tilde{U}_0$ thereafter. Hence the line search restricted to $[\tilde{u}_0^\nu, \tilde{u}_2^\nu]$ is again "perfect." On exiting from the line search in Step 5, we have

$$\begin{aligned}
\frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\hat{\tilde{u}}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})} &= \frac{(\bar{\lambda}^\nu)^2 d^\nu \cdot A d^\nu}{2\big[\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})\big]} \\
&= \frac{\big[A(\tilde{u}_0^\nu - \tilde{u}^*)\cdot \tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big) + A(\tilde{u}_0^\nu - \tilde{u}^*)\cdot \tilde{S}_p^\perp(u_0^\nu - \bar{u})\big]^2}{(d^\nu \cdot A d^\nu)\big[(\tilde{u}_0^\nu - \tilde{u}^*)\cdot A(\tilde{u}_0^\nu - \tilde{u}^*) - (\bar{\tilde{u}} - \tilde{u}^*)\cdot A(\bar{\tilde{u}} - \tilde{u}^*)\big]} \\
&= \frac{\big[d^\nu \cdot d^\nu - \big(\tilde{u}_0^\nu - \bar{\tilde{u}} - A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]^2}{(d^\nu \cdot A d^\nu)\big[(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu - \bar{\tilde{u}}) + 2(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\bar{\tilde{u}} - \tilde{u}^*)\big]}.
\end{aligned}$$

Defining $b(\tilde{u}) := \tilde{u} - \bar{\tilde{u}} - A(\tilde{u} - \tilde{u}^*)$ and observing $\tilde{f}(\tilde{u}_0^{\nu+1}) \leq \tilde{f}(\hat{\tilde{u}}_0^{\nu+1})$, we obtain from the equation $\tilde{S}_p\big(A(\bar{\tilde{u}} - \tilde{u}^*)\big) = 0$ (which is based on the optimality of $\hat{\tilde{u}}$) that

$$\begin{aligned}
\frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})} &\geq \frac{\big[d^\nu \cdot d^\nu + b(u_0^\nu)\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]^2}{(d^\nu \cdot A d^\nu)\big[(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu - \bar{\tilde{u}}) - 2b(\bar{\tilde{u}})\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]} \\
&\geq \frac{(d^\nu \cdot d^\nu)^2 + \big[b(\tilde{u}_0^\nu)\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]^2}{(d^\nu \cdot A d^\nu)\big[(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu - \bar{\tilde{u}}) - 2b(\bar{\tilde{u}})\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]}.
\end{aligned}$$

By Theorem 3.2 the algorithm converges, hence for arbitrarily chosen $\tilde{\varepsilon} > 0$, we have $\|b(\tilde{u}_0^\nu) - b(\bar{\tilde{u}})\| \leq \tilde{\varepsilon}$ for sufficiently large ν. Then

$$\begin{aligned}
|b(\tilde{u}_0^\nu)\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})| &= |b(\bar{\tilde{u}})\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}}) + \big(b(\tilde{u}_0^\nu) - b(\bar{\tilde{u}})\big)\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})| \\
&\geq |b(\bar{\tilde{u}})\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})| - |\big(b(\tilde{u}_0^\nu) - b(\bar{\tilde{u}})\big)\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})| \\
&\geq |b(\bar{\tilde{u}})\cdot \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})| - \tilde{\varepsilon}\|\tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\|.
\end{aligned}$$

But $|b(\bar{\tilde{u}})\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})| = \|\nabla\tilde{f}(\bar{\tilde{u}})\|\cdot\|\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})\|$. Therefore

$$\frac{|b(\tilde{u}_0^\nu)\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})|}{|b(\bar{\tilde{u}})\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})|} \geq 1-\frac{\tilde{\varepsilon}}{\|\nabla\tilde{f}(\bar{\tilde{u}})\|} \tag{4.24}$$

where $\nabla\tilde{f}(\bar{\tilde{u}})\neq 0$, for otherwise $\tilde{U}_0=\tilde{U}$ and then $\tilde{u}_0^\nu\in\tilde{U}_0$ in contradiction to our assumption in Case 2.

Now, if $d^\nu\cdot d^\nu \geq -b(\tilde{u}_0^\nu)\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})$, we obtain from (4.15) that

$$\begin{aligned}\frac{\tilde{f}(\tilde{u}_0^\nu)-\tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu)-f(\bar{\tilde{u}})} &\geq \frac{(d^\nu\cdot d^\nu)^2}{(d^\nu\cdot Ad^\nu)\big[(\tilde{u}_0^\nu-\bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu-\bar{\tilde{u}})+2d^\nu\cdot d^\nu\big]}\\ &\geq \frac{1}{\|A\|\big[2+\|A\|(5+4\|A\|^2)\big]}.\end{aligned}$$

Otherwise $d^\nu\cdot d^\nu < -b(\tilde{u}_0^\nu)\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})$, and then

$$\begin{aligned}&\frac{\tilde{f}(\tilde{u}_0^\nu)-\tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu)-\tilde{f}(\bar{\tilde{u}})}\\ &\geq \frac{\big[b(\tilde{u}_0^\nu)\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})\big]^2}{\|A\|\big[b(\bar{\tilde{u}})\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})\big]\Big[\|A\|(5+4\|A\|^2)b(\bar{\tilde{u}})\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})+2b(\bar{\tilde{u}})\cdot\tilde{S}_P^{\perp}(\tilde{u}_0^\nu-\bar{\tilde{u}})\Big]}\\ &= \frac{1}{\|A\|\big(2+\|A\|(5+4\|A\|^2)\big)}\left(1-\frac{\tilde{\varepsilon}}{\|\nabla\tilde{f}(\bar{\tilde{u}})\|}\right)\end{aligned}$$

by (4.24). Thus, we have

$$\liminf_{\nu\to\infty}\frac{\tilde{f}(\tilde{u}_0^\nu)-\tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu)-\tilde{f}(\bar{\tilde{u}})} \geq \frac{1}{\|A\|\big(2+\|A\|(5+4\|A\|^2)\big)},$$

which can be written as

$$\limsup_{\nu\to\infty}\frac{\tilde{f}(\tilde{u}_0^{\nu+1})-\tilde{f}(\bar{\tilde{u}})}{\tilde{f}(\tilde{u}_0^\nu)-\tilde{f}(\bar{\tilde{u}})} \leq 1-\frac{1}{\|A\|\big(2+\|A\|(5+4\|A\|^2)\big)}. \tag{4.25}$$

Noting that $\|A\| = 1+\|(\tilde{R}_{ff}\ \tilde{R}_{fr})\|^2 \leq 1+\|\tilde{R}\|^2 = 1+\gamma^2$, we get the first inequality in (4.19), which is also true for Case 1 in view of (4.20) since $c_2<c_1$. The dual part has a parallel argument. □

Observe that the rates in (4.20) and (4.21) are much better than the ones in (4.19). The former will be effective if any interactive restarts occur for $\nu\geq\bar{\nu}$, as indicated in the theorem. This partially explains the effects of interactive restarts on the algorithm as observed in our numerical tests.

The role of the constant $\gamma=\gamma(P,Q,R)$ in the convergence rate in Theorem 4.2 has been borne out in our numerical tests, although because of the interactive restarts the method appears to work much better than one might expect from "steepest descent." We have definitely observed in small-scale problems where some idea of the size of γ is available that the convergence is faster with low γ than with high γ.

Although Theorem 4.2 centers on the specialization of Algorithm 0 to Algorithm 1, the argument has content also for Algorithm 2. Recall from the discussion after the

statement of Algorithm 0 in Section 2 that in every k iterations of Algorithm 0 (when implemented with cycle size $k > 1$) there is at least one primal line search on $[u_0^\nu, u_2^\nu]$ and at least one dual line search on $[v_0^\nu, v_2^\nu]$. This gives us the following result about Algorithm 2, which will be complemented by a finite termination result in Theorem 4.5.

COROLLARY 4.3. (Rate of convergence of PDCG.) *Suppose the critical face condition is satisfied. Then Algorithm 2 with $\varepsilon = 0$ converges at least k-step linearly in the sense that*

$$\limsup_{\nu\to\infty} \frac{f(u_0^{\nu+k}) - f(\bar{u})}{f(u_0^\nu) - f(\bar{u})} \le c_1 \ \text{ and } \ \limsup_{\nu\to\infty} \frac{g(v_0^{\nu+k}) - g(\bar{v})}{g(v_0^\nu) - g(\bar{v})} \le c_1, \tag{4.26}$$

where c_1 is the value defined in (4.17)*, unless the algorithm terminates after a finite number of iterations with $(\hat{u}, \hat{v}) = (\bar{u}, \bar{v})$.*

To derive a special finite termination property of Algorithm 2, we need the following.

PROPOSITION 4.4. (inequalities in PDCG.) *Suppose the critical face condition is satisfied. Let $\hat{\nu}$ be an iteration number such that for $\nu \ge \hat{\nu}$, all the points u_0^ν, u_2^ν and v_0^ν, v_2^ν are in an ultimate quadratic region $U^* \times V^*$ in Definition 3.7, where U^* is contained in the $\|\cdot\|_P$-ball around $\bar{u}$ of radius $\frac{1}{2}$, and likewise V^* is contained in the $\|\cdot\|_Q$-ball around $\bar{v}$ of radius $\frac{1}{2}$. If $u_0^{\nu'} \in U_0$ for some $\nu' \ge \hat{\nu}$, then in Algorithm 2 one has*

$$\langle w_p^\nu,\, u_e^{\nu-1} - u_0^\nu \rangle_P > 0 \tag{4.27}$$

whenever (2.1)–(2.4) *are used to generate u_e^ν for $\nu > \nu'$, and similarly if $v_0^{\nu''} \in V_0$ for some $\nu'' \ge \hat{\nu}$, then in Algorithm 2 one has*

$$\langle w_d^\nu,\, v_e^{\nu-1} - v_0^\nu \rangle_Q > 0 \tag{4.28}$$

whenever (2.5)–(2.8) *are used to generate v_e^ν for $\nu > \nu''$.*

Proof. It suffices once more to give the argument in the context of the transformed variables. Observe that the gradient mapping $\nabla\tilde{f}$ is strongly monotone, and that $\tilde{w}_p^\nu = \nabla\tilde{f}(\tilde{u}_0^\nu) - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})$ with $\tilde{u}_0^\nu \in [\tilde{u}_0^{\nu-1}, \tilde{u}_e^{\nu-1}]$ when (2.1)–(2.4) are used to generate u_e^ν in the primal. Hence the primal part of the assertion is true if $\tilde{u}_0^\nu \ne \tilde{u}_e^{\nu-1}$ for $\nu > \nu'$. According to Proposition 3.5 (cf. also the remark after it), one has $\tilde{u}_0^\nu \in \tilde{U}_0$ for all $\nu \ge \nu'$. We partition all vectors in conformity with the scheme in (4.6). Then $\tilde{u}_{0,r}^\nu = \bar{\tilde{u}}_r$ and $\tilde{u}_{2,r}^\nu = \bar{\tilde{u}}_r$.

If the $(\nu-1)$th iteration with $\nu > \nu'$ is the first iteration of a primal cycle, then the line search is performed on $[\tilde{u}_0^{\nu-1}, \tilde{u}_2^{\nu-1}]$. For the direction vector $d^{\nu-1} := \tilde{u}_2^{\nu-1} - \tilde{u}_0^{\nu-1}$, the optimal step length $\bar{\lambda}^\nu$ for $\tilde{u} = \tilde{u}_0^\nu + \lambda d^\nu$ to minimize the quadratic form (4.9) over all $\lambda \in [0, \infty)$ can be derived from the expression in (4.11) as

$$\bar{\lambda}^{\nu-1} = \frac{-d^{\nu-1}\cdot\nabla\tilde{f}(\tilde{u}_0^{\nu-1})}{d^{\nu-1}\cdot A d^{\nu-1}} = \frac{d_f^{\nu-1}\cdot d_f^{\nu-1}}{d_f^{\nu-1}\cdot A_{ff} d_f^{\nu-1}},$$

where the first equation in Proposition 3.8 has been used with $\nabla\tilde{f}(\tilde{u}_0^{\nu-1})$, and A_{ff} is the Hessian component in (4.13). Note that none of the eigenvalues of A_{ff} is less than 1. Hence $\bar{\lambda}^{\nu-1} \le 1$, and the equality holds only if $d_f^{\nu-1}$ is an eigenvector corresponding to 1 as an eigenvalue of A_{ff}, i.e., $A_{ff} d_f^{\nu-1} = d_f^{\nu-1}$. But it follows from (4.11) and the

first equation in Proposition 3.8 that we also have $A_{ff}(\bar{\tilde{u}}_f - \tilde{u}_{0,f}^{\nu-1}) = d_f^{\nu-1}$. Therefore $\bar{\lambda}^{\nu-1} = 1$ implies $\tilde{u}_{2,f}^{\nu-1} = \bar{\tilde{u}}_f$ and $\tilde{u}_2^{\nu-1} = \bar{\tilde{u}}$. And then $\tilde{u}_0^{\nu} = \bar{\tilde{u}}$, i.e., the iteration terminates at the primal optimal solution.

If the $(\nu-1)$th iteration with $\nu > \nu'$ is not the first iteration of a primal cycle, then formulas (2.1)–(2.4) are used to define $\tilde{u}_e^{\nu-1}$. In the proof of Proposition 3.5, we have actually shown that $\tilde{u}_e^{\nu-1} \in \tilde{U}_0$ for all $\nu > \nu'$. Hence $[\tilde{u}_0^{\nu-1}, \tilde{u}_e^{\nu-1}] \subset \tilde{U}_0$ for all $\nu > \nu'$. Then it follows from (2.4) that $\|\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu-1}\| \geq 1$ unless $\tilde{u}_e^{\nu-1}$ is on the relative boundary of $\tilde{U}_0$. In either case we have $\tilde{u}_0^{\nu} \neq \tilde{u}_e^{\nu-1}$ again, since $\tilde{u}_0^{\nu-1} \in \tilde{U}^*$ for $\nu > \nu'$ and $\tilde{U}^*$ is contained in the $\|\cdot\|_P$-ball around $\bar{u}$ of radius $\frac{1}{2}$. The dual claims can be verified similarly. □

THEOREM 4.5. (A finite termination property of PDCG.) *Assume that the critical face condition is satisfied. Suppose that the cycle size k chosen in Algorithm 2 is such that $k > \bar{k}$, where $\bar{k}$ denotes the rank of the linear transformation $u \mapsto S_d\big(RS_p(u)\big)$. (It suffices in this to have $k > \min\{m, n\}$.) Let $\hat{\nu}$ be an iteration number as defined in Proposition 4.4 and satisfying the conditions there. If $u_0^{\nu'} \in U_0$ for some $\nu' \geq \hat{\nu}$ (as is sure to happen in an interactive primal restart at that stage), then the algorithm will terminate in the next full primal cycle, if not earlier. Similarly, if $v_0^{\nu''} \in V_0$ for some $\nu'' \geq \hat{\nu}$ (as is sure to happen in an interactive dual restart at that stage), then the algorithm will terminate in the next full dual cycle, if not earlier.*

Proof. We concentrate on the primal part; the proof of the dual part is parallel. In the transformed variables, where we place the argument once more, $\bar{k}$ is the rank of the submatrix $\tilde{R}_{ff}$ of $\tilde{R}$ in (4.8). Note that for $\nu \geq \hat{\nu}$ the process is in an quadratic region of the problem as specified in Proposition 4.4. In the proof of Proposition 4.4, we have shown that for all $\nu \geq \nu'$, $[\tilde{u}_0^{\nu}, \tilde{u}_e^{\nu}] \subset \tilde{U}_0$, and that the line searches on $[\tilde{u}_0^{\nu}, \tilde{u}_e^{\nu}]$ are "perfect" in the sense that, on exiting Step 5 of iteration ν, $\tilde{u}_0^{\nu+1}$ minimizes $\tilde{f}$ on the half-line from $\tilde{u}_0^{\nu}$ in the direction of $\tilde{u}_e^{\nu} - \tilde{u}_0^{\nu}$. Observe there is no interactive primal restart in the first $k-1$ iterations of a full primal cycle, i.e., $\hat{\tilde{u}}_0^{\nu} = \tilde{u}_0^{\nu}$ for these iterations. We claim now that the search direction vectors $\tilde{u}_e^{\nu} - \tilde{u}_0^{\nu}$ and $\tilde{v}_e^{\nu} - \tilde{v}_0^{\nu}$ are the same as the ones that would be generated by a conjugate gradient algorithm on $\tilde{f}$ relative to $\operatorname{aff}\tilde{U}_0$. The finite termination property will be a consequence of observing that the Hessians of $\tilde{f}$ in an quadratic region of the problem (cf. (4.11)) have at most $\bar{k}+1$ different eigenvalues.

The proof of the claim will go by induction. We know from Proposition 3.8 that the claim is true for the first iteration of the full primal cycle in question. Suppose it is true for the $(\nu-1)$th iteration generating $\tilde{u}_0^{\nu}$ in that cycle, but $\tilde{u}_0^{\nu} \neq \bar{\tilde{u}}$. Then by (4.27) in Proposition 4.4, the first alternative of (2.2) will be used to generate β_p^{ν}. Hence it follows from (2.1)–(2.3) and Proposition 3.8 that

(4.29)
$$(\tilde{u}_{cg}^{\nu} - \tilde{u}_0^{\nu})_f = \frac{(\tilde{u}_2^{\nu} - \tilde{u}_0^{\nu})_f + \beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f}{1+\beta_p^{\nu}} = \frac{-\nabla\tilde{f}(\tilde{u}_0^{\nu})_f + \beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f}{1+\beta_p^{\nu}},$$
$$\beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f = \frac{\max\{0,\, (\nabla\tilde{f}(u_0^{\nu})_f - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})_f)\cdot\nabla\tilde{f}(\tilde{u}_0^{\nu})_f\}}{(\nabla\tilde{f}(\tilde{u}_0^{\nu})_f - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})_f)\cdot(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f,$$

where all the points $\tilde{u}_e^{\nu-1}$, $\tilde{u}_0^{\nu}$ and $\tilde{u}_2^{\nu}$ are on the critical face $\tilde{U}_0$. By the induction hypothesis, the directions of line search are the same as the ones generated by the conjugate gradient algorithm in all the previous iterations of the cycle. Hence

$$\nabla\tilde{f}(\tilde{u}_0^{\nu})_f\cdot\nabla\tilde{f}(\tilde{u}_0^{\nu-1})_f = 0$$

which implies $(\nabla\tilde{f}(\tilde{u}_0^\nu)_f - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})_f)\cdot\nabla\tilde{f}(\tilde{u}_0^\nu)_f \geq 0$. Therefore, by noting that $\tilde{u}_e^{\nu-1} - \tilde{u}_0^\nu$ is a positive multiple of $\tilde{u}_0^\nu - \tilde{u}_0^{\nu-1}$, we obtain

$$\beta_p^\nu(\tilde{u}_e^{\nu-1} - \tilde{u}_0^\nu)_f = \frac{(\nabla\tilde{f}(\tilde{u}_0^\nu)_f - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})_f)\cdot\nabla\tilde{f}(\tilde{u}_0^\nu)_f}{(\nabla\tilde{f}(\tilde{u}_0^\nu)_f - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})_f)\cdot(\tilde{u}_0^{\nu-1} - \tilde{u}_0^\nu)_f}(\tilde{u}_0^{\nu-1} - \tilde{u}_0^\nu)_f. \tag{4.30}$$

Comparing (4.29) and (4.30) with the conjugate gradient formulas of Hestenes and Stiefel [22], we see that the vector $\tilde{u}_{cg}^\nu - \tilde{u}_0^\nu$ is equivalent to the search direction vector in a standard conjugate gradient algorithm for $\tilde{f}$ relative to the free variables, i.e., over $\operatorname{aff}\tilde{U}_0$. □

Observe that the rank of linear transformation in Theorem 4.5 is bounded above by the ranks of the projection mappings S_p and S_d, which are $\dim U_0$ and $\dim V_0$. Hence

$$\bar{k} \leq \min\{\dim U_0, \dim V_0\}.$$

Therefore, even in the case that the original problems $(\mathcal{P})$ and $(\mathcal{Q})$ are of high dimension, the optimal solution can still be reached in a relatively short cycle after entering an ultimate quadratic regions for the problem if merely one of the critical faces U_0 and V_0 happens to be of low dimension, provided that at least one of the critical faces is eventually reached by the corresponding iterates. This condition will certainly be satisfied if any interactive restarts occur for $\nu \geq \hat{\nu}$, since all points $\hat{u}_1^\nu$ and $\hat{v}_1^\nu$ will be on the critical faces U_0 and V_0 by Proposition 1.8, and once u_0^ν or v_0^ν are on the critical faces, they will stay there (Proposition 3.5).

There are ways to force this condition to be satisfied, such as to insert at the beginning of each primal cycle a line search in the direction of the projection of $-\nabla f(u_0^\nu)$ on the tangent cone to U at u_0^ν, and similarly in the dual. (See Burke and Moré [23].) But even without such remedies, we often find in our test problems that the critical faces are identified in the tail of iteration, and that restarts do occur in most cases, after which the iteration terminates at the optimal solution in a few steps.

5. Envelope Properties. To finish off, we establish two results on the finite-envelope property of the points u_1^ν and v_1^ν in our algorithms.

PROPOSITION 5.1. (General saddle point property of iterates.) *On exiting from Step 5 of Algorithm 0 with $\hat{u}_0^{\nu+1}$ and $\hat{v}_0^{\nu+1}$, the elements $\hat{u}_1^{\nu+1} \in G(\hat{v}_0^{\nu+1})$ and $\hat{v}_1^{\nu+1} \in F(\hat{u}_0^{\nu+1})$ that will be calculated on return to Step 1 will be such that the pair $(\hat{u}_0^{\nu+1}, \hat{v}_1^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $[u_0^\nu, u_2^\nu] \times V$, while the pair $(\hat{u}_1^{\nu+1}, \hat{v}_0^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $U \times [v_0^\nu, v_2^\nu]$. In particular, $\hat{u}_1^{\nu+1}$ will be the unique minimizing point relative to U for the envelope function*

$$f^\nu(u) := \max_{v\in[v_0^\nu, v_2^\nu]} L(u,v) \leq \max_{v\in V} L(u,v) = f(u),$$

whereas $\hat{v}_1^{\nu+1}$ will be the unique maximizing point relative to V for the envelope function

$$g^\nu(u) := \min_{u\in[u_0^\nu, u_2^\nu]} L(u,v) \geq \min_{u\in U} L(u,v) = g(u).$$

Proof. Recall that because we are in the fully quadratic case, $L(u,v)$ and $f(u)$ are strictly convex in u, while $L(u,v)$ and $g(v)$ are strictly concave in v. In particular, $\hat{u}_0^{\nu+1}$ must be the unique solution to the problem in Step 5 of minimizing $f(u)$ over $u \in [u_0^\nu, u_2^\nu]$. This is the primal problem of extended linear-quadratic programming that corresponds to L on $[u_0^\nu, u_2^\nu] \times V$ instead of $U \times V$. Applying Theorem 1.1 to

it instead of to the original problem we deduce the existence of a vector v' such that $(\hat{u}_0^{\nu+1}, v')$ is a saddle point of L relative to $[u_0^\nu, u_2^\nu] \times V$. Then v' is the unique point maximizing $L(\hat{u}_0^{\nu+1}, v)$ with respect to $v \in V$ (by the strict concavity of $L(u,v)$ in v). Thus, v' is the unique element of $F(\hat{u}_0^{\nu+1})$, so $v' = \hat{v}_1^{\nu+1}$. It follows from Theorem 1.1 again that $(\hat{u}_0^{\nu+1}, \hat{v}_1^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $[u_0^\nu, u_2^\nu] \times V$, and $\hat{v}_1^{\nu+1}$ is the unique solution to the corresponding dual problem, which by definition consists of maximizing the function g^ν over V.

The rest of the assertions are true by a parallel argument in which Theorem 1.1 is applied to the primal and dual problems that correspond to L on $U \times [v_0^\nu, v_2^\nu]$. □

PROPOSITION 5.2. (Ultimate saddle point property of iterates.) *Suppose the critical face condition is satisfied. Let $\hat{\nu}$ be an iteration number as specified in Proposition 4.4 and satisfying the conditions there. If $\nu = r \geq \hat{\nu}$ is the first iteration of some primal cycle with $v_0^r \in U_0$, then for all $\nu \geq r$ in that cycle, on exiting from Step 5 of Algorithm 2 with $\hat{u}_0^{\nu+1}$ the element $\hat{v}_1^{\nu+1} \in F(\hat{u}_0^{\nu+1})$ that will be calculated on return to Step 1 will be such that $(\hat{u}_0^{\nu+1}, \hat{v}_1^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $U^\nu \times V$, where*

$$U^\nu := \operatorname{aff}\big\{[u_0^r, u_e^r] \times \cdots \times [u_0^\nu, u_e^\nu]\big\} \cap U_0 \tag{5.1}$$

and $\dim(\operatorname{aff}\big\{[u_0^r, u_e^r] \times \cdots \times [u_0^\nu, u_e^\nu]\big\}) = \nu - r + 1$. *In particular, $\hat{v}_1^{\nu+1}$ will be the unique maximizing point relative to V for the envelope function*

$$g^\nu(v) := \min_{u \in U^\nu} L(u,v) \geq \min_{u \in U} L(u,v) = g(u),$$

and one will have $g^{\nu+1} \leq g^\nu$ in that primal cycle. Moreover, for $\nu = r + d_1 - 1$ with $d_1 := \dim U_0$, it will be true that $g^\nu = g$ in an ultimate quadratic region for the problem, and also that $\hat{v}_1^{\nu+1} = \bar{v}$, as long as the algorithm does not terminate earlier.

Similarly, if $\nu = s \geq \hat{\nu}$ is the first iteration of some dual cycle with $v_0^s \in V_0$, then for all $\nu \geq s$ in that cycle, on exiting from Step 5 of Algorithm 2 with $\hat{v}_0^{\nu+1}$ the element $\hat{u}_1^{\nu+1} \in G(\hat{v}_0^{\nu+1})$ that will be calculated on return to Step 1 will be such that $(\hat{u}_1^{\nu+1}, \hat{v}_0^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $U \times V^\nu$, where

$$V^\nu := \operatorname{aff}\big\{[v_0^s, v_e^s] \times \cdots \times [v_0^\nu, v_e^\nu]\big\} \cap V_0, \tag{5.2}$$

with $\dim(\operatorname{aff}\big\{[v_0^s, v_e^s] \times \cdots \times [v_0^\nu, v_e^\nu]\big\}) = \nu - s + 1$. *In particular, $\hat{u}_1^{\nu+1}$ will be the unique minimizing point relative to U for the envelope function*

$$f^\nu(u) := \max_{v \in V^\nu} L(u,v) \leq \max_{v \in V} L(u,v) = f(u),$$

and one will have $f^{\nu+1} \geq f^\nu$ in that dual cycle. Moreover, for $\nu = s + d_2 - 1$ with $d_2 := \dim V_0$, it will be true that $f^\nu = f$ in an ultimate quadratic region for the problem, and also that $\hat{u}_1^{\nu+1} = \bar{u}$, as long as the algorithm does not terminate earlier.

Proof. We concentrate on the primal part; the proof of the dual part is parallel. The argument is similar to the one given for Proposition 5.1, but with the segment $[u_0^\nu, u_2^\nu]$ replaced by U^ν. Recall from the proof of Theorem 4.5 that for $\nu \geq r$, the primal procedure is equivalent to the conjugate gradient algorithm on the restriction of f to the affine hull $\operatorname{aff} U_0$ of the critical face U_0. Therefore the vectors $u_e^r - u_0^r, \cdots, u_e^\nu - u_0^\nu$ are linearly independent, and $\hat{u}_0^{\nu+1}$ minimizes $f(u)$ over $u \in U^\nu$. The inequality $g^{\nu+1} \leq g^\nu$ follows from the inclusion $U^{\nu+1} \supset U^\nu$. When $\nu = r + d_1 - 1$ we have $\dim\big([u_0^r, u_e^r] \times \cdots \times [u_0^\nu, u_e^\nu]\big) = d_1$, and then $U^\nu = U_0$. From the fact that (3.3) holds

in $\tilde{V}^*$ (cf. the derivation of this relation in the proof of Theorem 3.6) we get $g^\nu = g$ in an ultimate quadratic region. □

This result tells us that on entering an ultimate quadratic region, the primal iterations in Algorithm 2 produce an improving envelope for the dual objective function which approaches that function, whereas the dual iterations produce an improving envelope for the primal objectives which approaches that function. To some extent this explains the phenomenon we have observed in our numerical experiments that restarts often incur fast termination, or at least bring significant progress in the next few iterations.

6. Numerical Tests. Numerical tests of Algorithm 1, the Primal-Dual Steepest Descent Algorithm (PDSD), and Algorithm 2, the Primal-Dual Conjugate Gradient Algorithm (PDCG), have been conducted on a DECstation 3100 with double precision on some medium to large-sized problems. For comparisons we have used the Basic Finite Envelope Method (BFEM) of [6] and the Stanford LSSOL code of Gill, Hammarling, Murray, Saunders and Wright [24] for quadratic programming. To enhance the performance of LSSOL in this situation, we tailored its Cholesky factorization subroutine to take advantage of the special structure of the P and Q matrices in our examples.

Comparisons with LSSOL are based on the fact that any extended-linear-quadratic programming problem can be converted into a standard quadratic programming problem by introducing auxiliary variables and additional constraints [1, Theorem 1]. It must be kept in mind, however, that such a transformation not only increases the dimension substantially but disrupts much of the large-scale structure that might be put to use. A fundamental difficulty with any comparisons with available QP methods, therefore, is that such methods are not really designed to handle the kinds of problems we wish to tackle, which stem from [1], [2] and [6]. They typically require setting up and working with the huge R matrix, and trying to exploit any sparsity patterns that might be present in it, whereas we never need this matrix but work with decomposition in the calculation of the F and G, as explained in Section 1, after Proposition 1.6.

The integer recorded as the "size" of each problem is the number of primal variables and also the number of dual variables. (The two would not have to be the same.) Thus, size = 100 means that problem $(\mathcal{P})$ is an extended linear-quadratic programming problem on $\mathbb{R}^{100}$ for which the dual $(\mathcal{D})$ is likewise such a problem on $\mathbb{R}^{100}$, while the associated Lagrangian saddle point problem concerns a quadratic convex-concave function on a product of polyhedral sets in $\mathbb{R}^{100} \times \mathbb{R}^{100}$. In order to solve such a problem using LSSOL, it must be reformulated as a primal problem in 400 variables with 100 general equality constraints and 200 lower bounds on the auxiliary variables, in addition to having the original polyhedral constraints on the 100 primal variables.

In all the tests of PDCG and PDSD we have taken $\delta = 10^{-2}$ as the progress threshold and $\varepsilon = 10^{-8}$ as the optimality threshold. For PDCG we have taken $k = 5$ as the cycle size parameter (whereas PDSD always has $k = 1$ by definition). We have run BFEM with "mode=1," which means that in each iteration a quadratic saddle point subproblem is solved over a product of two triangles. For the sake of expediency in solving this small subproblem we have set it up as a standard QP problem in the manner of [1, Theorem 1] and have applied LSSOL. No doubt the CPU time could be improved by using a customized procedure within BFEM instead of this heavy-handed approach.

The generation of test problems of large size raises serious questions about the

representative nature of such problems. It does not make sense to think of a large problem simply in terms of a large matrix, the elements of which are all random. Rather, a certain amount of structure must be respected. As an attempt to address this issue, we have taken all our problems to have the (deterministic) dynamical structure described in Rockafellar and Wets [5]. Only the parameters natural to this structure have been randomized. The dynamical structure enables us to use special routines in calculating $f(u)$ and $F(u)$, and on the other hand $g(v)$ and $G(v)$ [7]. For this purpose, and in implementing BFEM, we rely on code written by S. E. Wright [25] at the University of Washington.

The problems have been obtained as discretized versions of certain continuous-time problems of extended linear-quadratic optimal control of the kind developed in Rockafellar [4]. The first digit of the problem number corresponds to different continuous-time problems and the second digit corresponds to different discretization levels, i.e., the number of subintervals into which the fixed time interval has been partitioned, which determines the size of the discretized problem. Hence, e.g., the problems 0.1, 1.1, ..., 9.1 are the discretization of 10 different continuous-time problems with the same discretization level (*a transverse family of test problems*), and the problems 1.0, 1.1, ..., 1.7 are the discretization of one continuous-time problem with 8 different discretization levels (*a vertical family of test problems*). Only the data values in the continuous-time model have been generated randomly, and in each vertical family these are the same for all the problems. By increasing the number of subintervals, one can get larger and larger problems which remain stable with respect to the numerical scaling.

Table 1. Test results of problems 0.1–9.1.

		CPU time (sec.)				Iterations			
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	PDCG	PDSD	BFEM	LSSOL
0.1	100	4.6	4.8	6.6	283.1	23	34	31	500
1.1	100	5.0	5.8	7.5	295.0	28	50	37	497
2.1	100	5.0	4.0	8.1	299.7	28	24	41	495
3.1	100	3.0	2.6	3.4	339.8	5	5	8	562
4.1	100	3.8	3.5	3.8	353.2	13	17	11	619
5.1	100	3.2	2.7	3.5	314.5	8	6	9	544
6.1	100	3.5	3.0	3.8	339.2	11	11	11	552
7.1	100	3.6	3.7	4.3	256.0	13	20	14	445
8.1	100	4.5	5.2	*17.5	290.6	22	42	**	481
9.1	100	3.5	3.3	4.0	347.2	12	15	12	591

Table 2. Test results of problems 0.2–9.2.

		CPU time (sec.)				Iterations			
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	PDCG	PDSD	BFEM	LSSOL
0.2	340	9.2	8.9	15.3		24	28	31	
1.2	340	12.5	14.4	19.3		35	50	39	
2.2	340	10.1	11.9	20.5		25	38	42	
3.2	340	5.2	4.3	6.8		9	8	11	
4.2	340	7.8	6.6	8.7		18	17	15	
5.2	340	6.5	5.5	8.0		14	12	12	
6.2	340	5.7	5.1	7.3		11	11	12	
7.2	340	5.4	5.9	7.7		10	15	13	
8.2	340	9.8	11.2	20.3		25	38	42	
9.2	340	6.0	6.4	9.5		12	17	17	

Table 3. Test results of problems 0.4–9.4.

		CPU time (sec.)				Iterations			
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	PDCG	PDSD	BFEM	LSSOL
0.4	5140	122.6	138.6	270.8		23	32	38	
1.4	5140	177.9	230.9	315.7		32	52	44	
2.4	5140	218.6	191.7	399.3		40	44	56	
3.4	5140	46.7	45.0	110.1		8	9	16	
4.4	5140	111.8	94.8	126.7		20	20	18	
5.4	5140	71.4	64.5	133.2		12	13	19	
6.4	5140	80.5	78.2	141.2		14	16	20	
7.4	5140	54.9	85.1	104.7		10	19	15	
8.4	5140	161.1	235.9	362.9		29	55	50	
9.4	5140	76.5	77.8	115.5		14	17	16	

The test results in Tables 1, 2 and 3 concern transverse families of size 100, 340 and 5140 respectively. The problems in the first family are small enough for the LSSOL approach to be viable as a comparison. But for the second and third families, our DECstation 3100 falls short of having enough memory for the LSSOL approach. Here we see that PDCG and PDSD are in the leading positions with BFEM not very far behind in terms of CPU times.

The notation $**$ for the iterations of BFEM on problem 8.1 signifies that the method failed to terminate with optimality in 100 iterations. The corresponding figure for CPU time is preceded by * since it only indicates how long the first 100 iterations took. (The same conventions are adopted in all other tables.)

Table 4. Test results of discretized problems 0.0–0.7.

		CPU time (sec.)/Iterations				
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	Value
0.0	40	2.9/11	3.0/15	3.3/13	35.3/327	23.8626
0.1	100	4.3/23	4.8/34	6.6/31	244.4/500	15.7824
0.2	340	9.0/24	9.1/28	15.2/31		15.2107
0.3	1300	27.1/22	32.1/32	58.7/34		15.2145
0.4	5140	122.5/23	137.2/32	269.2/38		15.2179
0.5	20500	568.6/27	593.7/32	1396.2/46		15.2188
0.6	81940	2873.8/27	2722.6/32	*10637.6/$**$		15.2190
0.7	100020	4209.3/28	3976.5/32	7086.3/38		15.2191

Table 5. Test results of discretized problems 1.0–1.7.

		CPU time (sec.)/Iterations				
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	Value
1.0	40	2.9/15	3.0/21	3.9/22	40.9/360	242.05983
1.1	100	4.9/28	5.9/50	7.5/37	294.8/497	249.07378
1.2	340	12.4/35	14.4/50	19.1/39		249.77975
1.3	1300	45.3/37	52.2/52	76.0/44		249.79866
1.4	5140	178.4/32	230.7/52	317.9/44		249.79956
1.5	20500	812.4/36	1007.5/52	1421.5/45		249.79972
1.6	81940	4015.8/36	4699.9/52	6119.9/45		249.79976
1.7	100020	5749.6/36	6538.5/52	8264.0/44		249.79976

Table 6. Test results of discretized problems 2.0–2.7.

		CPU time (sec.)/Iterations				
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	Value
2.0	40	3.6/28	4.4/63	*9.7/**	44.7/446	-261.5042
2.1	100	4.7/28	4.1/24	8.2/41	254.8/495	-362.2297
2.2	340	9.5/25	11.1/38	20.0/42		-369.7334
2.3	1300	41.4/33	39.8/39	97.6/56		-369.8073
2.4	5140	220.3/40	191.9/44	402.9/56		-369.8046
2.5	20500	936.9/43	769.4/40	1945.4/63		-369.8036
2.6	81940	4370.3/40	3396.5/39	8893.3/71		-369.8034
2.7	100020	6247.2/40	5123.7/40	10032.1/53		-369.8033

The test results in Tables 4, 5 and 6 refer to vertical families based on the first three continuous-time problems. They cover sizes that are generally too large for the LSSOL approach to be workable. The aim in this case is to examine the effects of increasing size in a context where these effects can be isolated from other aspects of the testing.

In these results the stability of the scaling is reflected by the way the optimal value settles down and converges. Much to be noted is the fact that, although the CPU time goes up as the problem size becomes larger, the number of iterations remains almost constant once the approximation is close. This suggests that the methods are able fairly quickly to identify the general location of the primal and dual optimal solutions, and that they accomplish this in a manner that is relatively insensitive to the number of variables and constraints. Quite the opposite behavior would be expected, of course, from an active-set QP method. The increase in CPU time seems mainly due to the increase in overhead in setting up the line searches as well as in the evaluations of $f(u)$, $F(u)$, $g(v)$ and $G(v)$ when the dimension is high.

Tables 7, 8 and 9 test the importance of the interactive restarts in PDCG and PDSD. The problems in this case are the same as in Tables 4, 5 and 6 correspondingly. For each problem, the methods were applied in the proposed form, allowing interactive restarts (the *yes* case), but then also in the modified form in which all such restarts are suppressed (the *no* case). The difference that this makes is evident. Interactive restarts have a big effect on the performance, and in the case of PDSD even dictate whether the method is successful or not, in the sense of terminating within 100 iterations. The tables also furnish information on the number of interactive restarts that occurred. For instance, for problem 0.5 under the interactive version of the PDSD method one reads that termination came in 32 iterations, and that in the course of these there were 7 interactive primal restarts and 6 interactive dual restarts. The noninteractive version took 89 iterations.

Another fact to be observed in these large problems is that the simplicity of PDSD sometimes overtakes the carefully designed properties of PDCG in CPU time. An interpretation is that when the dimension is very high, but PDCG is not yet near to the solutions and is anyway just using cycle size $k = 5$, the conjugate-gradient-like features do not always provide a gain that offsets the extra overhead. While the number of iterations in PDCG remains less, the time it takes, in comparison to PDSD, can be more. Perhaps the greatest advantage of these methods comes therefore from the information feedback involved in the interactive restarts, rather than from the attention paid to the choice of the descent (or ascent) direction.

Table 7. Test results of restart role in problems 0.0–0.7.

	CPU time (sec.)				Iterations			
	PDCG		PDSD		PDCG		PDSD	
Prb.	Yes	No	Yes	No	Yes	No	Yes	No
0.0	2.9	3.0	3.0	3.4	11(3/3)	16	15(3/8)	38
0.1	4.3	4.4	4.8	8.7	23(4/6)	25	34(4/9)	94
0.2	9.0	9.8	9.1	*24.8	24(3/7)	28	28(4/8)	**
0.3	27.1	35.6	32.1	*92.6	22(3/6)	31	32(7/6)	**
0.4	122.5	137.1	137.2	*416.0	23(4/6)	27	32(7/6)	**
0.5	568.6	561.6	593.7	1671.3	27(4/7)	27	32(7/6)	89
0.6	2873.8	2949.1	2722.6	*8326.5	27(4/7)	27	32(7/6)	**
0.7	4209.3	4012.9	3976.5	*12423.5	28(4/7)	27	32(7/6)	**

Table 8. Test results of restart role in problems 1.0–1.7.

	CPU time (sec.)				Iterations			
	PDCG		PDSD		PDCG		PDSD	
Prb.	Yes	No	Yes	No	Yes	No	Yes	No
1.0	2.9	3.3	3.0	4.3	15(3/4)	25	21(5/6)	64
1.1	4.9	5.7	5.9	*9.5	28(3/7)	38	50(5/7)	**
1.2	12.4	13.2	14.4	*26.2	35(2/7)	39	50(8/6)	**
1.3	45.3	54.1	52.2	*97.3	37(2/8)	47	52(4/8)	**
1.4	178.4	233.7	230.7	*444.9	32(3/6)	45	52(4/6)	**
1.5	812.4	1011.5	1007.5	*1978.8	36(2/7)	48	52(4/6)	**
1.6	4015.8	5043.6	4699.9	*8795.4	36(2/7)	48	52(4/6)	**
1.7	5749.6	7325.6	6538.5	*12726.2	36(2/7)	48	52(4/6)	**

Table 9. Test results of restart role in problems 2.0–2.7.

	CPU time (sec.)				Iterations			
	PDCG		PDSD		PDCG		PDSD	
Prb.	Yes	No	Yes	No	Yes	No	Yes	No
2.0	3.6	4.4	4.4	*5.4	28(5/6)	52	63(7/8)	**
2.1	4.7	6.6	4.1	7.0	28(7/4)	54	24(8/5)	68
2.2	9.5	16.1	11.1	*24.9	25(7/4)	51	38(7/3)	**
2.3	41.4	59.6	39.8	*92.9	33(9/5)	52	39(10/5)	**
2.4	220.3	295.2	191.9	*423.2	40(9/5)	58	44(9/5)	**
2.5	936.9	1362.4	769.4	*1899.4	43(9/6)	65	40(11/5)	**
2.6	4370.3	6385.8	3396.5	*8497.5	40(9/5)	61	39(11/5)	**
2.7	6247.2	9387.8	5123.7	*12359.7	40(9/5)	61	40(11/5)	**

REFERENCES

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

[2] R.T. Rockafellar and R.J.-B. Wets, *Linear-quadratic problems with stochastic penalties: the finite generation algorithm*, in Numerical Techniques for Stochastic Optimization Problems, Y. Ermoliev and R. J-B Wets (eds.), Springer-Verlag Lecture Notes in Control and Information Sciences No. 81, 1987, pp. 545-560.

[3] R.T. Rockafellar, *A generalized approach to linear-quadratic programming*, in Proc. International Conf. on Numerical Optimization and Appl. (Xi'an, China, 1986), pp. 58–66.

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

[5] R.T. Rockafellar and R.J.-B.Wets, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Opt. 28 (1990), pp. 810-822.
[6] R.T. Rockafellar, *Computational schemes for solving large-scale problems in extended linear-quadratic programming*, Math. Programming 48 (1990), pp. 447-474.
[7] R.T. Rockafellar, *Large-scale extended linear-quadratic programming and multistage optimization*, in Proc. Fifth Mexico-U.S. Workshop on Numerical Analysis (S. Gomez, J.-P. Hennart, R. Tapia, eds.), SIAM Publications, 1990.
[8] A. King, *An implementation of the Lagrangian finite generation method*, in Numerical Techniques for Stochastic Programming Problems, Y. Ermoliev and R. J-B Wets (eds.), Springer-Verlag, 1988.
[9] J.M. Wagner, *Stochastic Programming with Recourse Applied to Groundwater Quality Management*, doctoral dissertation, M.I.T., 1988.
[10] J.-S. Pang, *Methods for quadratic programming: a survey*, Computers and Chem. Engineering 7 (1983), pp. 583–594.
[11] Y.-Y. Lin and J.-S. Pang, *Iterative methods for large convex quadratic programs: a survey*, SIAM J. Control Opt. 25 (1987), pp. 383–411.
[12] Y. Ye and E. Tse, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming 44 (1989), pp. 157-179.
[13] R.D.C. Monteiro and I. Adler, *Interior path following primal-dual algorithms. Part II: Convex quadratic programming*, Math. Programming 44 (1989), pp. 43-66.
[14] D. Goldfarb and S. Liu, it An $0(n^3L)$ primal interior point algorithm for convex quadratic programming, preprint.
[15] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Opt. 29 (1991), pp. 119–138.
[16] P. Tseng, F̃urther applications of a splitting algorithm to decomposition in variational inequalities and convex programming, Math. Programming Studies 48 (1990), pp. 249–263.
[17] R.T. Rockafellar, *Convex Analysis*, Princeton Univ. Press, Princeton, NJ, 1970.
[18] R.T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Opt. 14 (1976), pp. 877–898.
[19] R.T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. of Op. Research 1 (1976), pp. 97–116.
[20] C. Zhu, *Modified proximal point algorithm for extended linear-quadratic programming*, accepted by Comput. Optim. and Appl., 1992.
[21] M.S. Bazaraa and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, Wiley, 1979.
[22] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice Hall, 1976
[23] J.V. Burke and J.J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal. 25 (1988), pp. 1197–1211.
[24] P.E. Gill, S.J. Hammarling, W. Murray, M.A. Saunders, and M.H. Wright, *User's guide for LSSOL (Version 1.0): a FORTRAN package for constrained linear least-squares and convex quadratic programming*, Tech. Rept. SOL 86–1, Dept. of Operations Research, Stanford University, 1986.
[25] S.E. Wright (with introduction by R.T. Rockafellar), *DYNFGM: Dynamic Finite Generation Method*, report, Department of Mathematics, University of Washington, 1989.