ON THE PRIMAL-DUAL STEEPEST DESCENT ALGORITHM FOR EXTENDED LINEAR-QUADRATIC PROGRAMMING

Ciyou Zhu *

Department of Mathematical Sciences
Johns Hopkins University, Baltimore, MD 21218

June 1992; revised May 1993.

Abstract. The aim of this paper is twofold. First, we propose new variants for the primal-dual steepest descent algorithm as one in the family of primal-dual projected gradient algorithms developed by Zhu and Rockafellar [1] for large-scale extended linear-quadratic programming. The variants include a second update scheme for the iterates, where the primal-dual feedback is arranged in a new pattern, as well as alternatives for the "perfect line search" in the original version of [1]. Secondly, we prove new linear convergence results for all these variants of the algorithm, including the original version as a special case, without the additional assumptions used in [1]. For the variants with the second update scheme, a much sharper estimation for the rate of convergence is obtained aws a result of the new primal-dual feedback pattern.

Keywords. Extended linear-quadratic programming, large-scale numerical optimization, projected gradient algorithm, primal-dual feedback.

---

* This work was supported by Eliezer Naddor Postdoctoral Fellowship in Mathematical Sciences at the Department of Mathematical Sciences, the Johns Hopkins University, during the year 1991–92.

Currrent address: Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439.

1. Introduction.

The primal-dual steepest descent algorithm (PDSD for short) is one in the family of primal-dual projected gradient algorithms proposed by Zhu and Rockafellar [1] for large-scale extended linear-quadratic programming, which arises as a flexible modeling scheme in dynamic and stochastic optimization [2–10].

Let $L(u, v)$ be the *Lagrangian function* defined as

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

where $p \in \mathbb{R}^n$, $q \in \mathbb{R}^m$, $R \in \mathbb{R}^{m\times n}$ and the matrices $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric and positive semidefinite. Let U and V be nonempty polyhedral convex sets in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. The primal problem of extended linear-quadratic programming is to

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

Associated with this primal problem is the dual problem

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

The problems $(\mathcal{P})$ and $(\mathcal{Q})$ are called *fully quadratic* if both the matrices P and Q are actually positive definite. The basic properties of the objective functions f and g, and the duality relationship between $(\mathcal{P})$ and $(\mathcal{Q})$, are included in the following two theorems.

Theorem 1.1 [5] (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 [5]**,** [2] (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, both the optimal values* $\inf(\mathcal{P})$ *and* $\sup(\mathcal{Q})$ *are finite and equal, and 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,* $f(\bar{u}) = g(\bar{v})$*.*

Hence the extended linear-quadratic programming can be cast in the form of finding a saddle point $(\bar{u}, \bar{v})$ of the Lagrangian $L(u, v)$ over $U \times V$. With the notations

$$\begin{aligned} \rho_{V,Q}(r) &= \sup_{v \in V} \{ r \cdot v - \tfrac{1}{2} v \cdot Q v \} \ \text{ for } \ r \in \mathbb{R}^m, \\ \rho_{U,P}(s) &= \sup_{u \in U} \{ s \cdot u - \tfrac{1}{2} u \cdot P u \} \ \text{ for } \ s \in \mathbb{R}^n, \end{aligned} \tag{1.2}$$

the objective functions in $(\mathcal{P})$ and $(\mathcal{Q})$ can be written as

$$\begin{aligned} f(u) =& p \cdot u + \tfrac{1}{2} u \cdot P u + \rho_{V,Q}(q - Ru), \\ g(v) =& q \cdot v - \tfrac{1}{2} v \cdot Q v - \rho_{U,P}(R^T v - p). \end{aligned} \tag{1.3}$$

According to Rockafellar [5], the ρ terms here can represent "sharp" constraints as well as penalty terms of piecewise linear-quadratic nature. These terms provide rich possibilities in mathematical modeling.

The extended linear-quadratic programming problems in multistage or stochastic optimization are usually of very high dimension on the one hand, while possessing special structures, such as the *Lagrangian decomposability* [7] on the other (see also Section 2). A foundation for numerical schemes regarding these problems has been laid out by Rockafellar and Wets [2] and Rockafellar [7], and elaborated for problems in multistage format by Rockafellar [8]. The PDSD algorithm [1] is designed specifically to take advantage of these results and to cope with the high dimensionality. The algorithm works with local structure in the primal and dual problems simultaneously. Computations for problems in multistage format could be handled through the system dynamics in such a way that no huge R matrix should be formed explicitly. A novel kind of primal-dual feedback is introduced between the primal part and the dual part of the algorithm to trigger advantageous *interactive restarts* [1]. The algorithm is capable of solving extended linear-quadratic programming problems of both the primal and dual dimensions up to 100,000 effectively on a DECstation 3100 [1].

The convergence of PDSD algorithm was proved in [1] as a special case of the results on the family of the primal-dual projected gradient algorithms. However, the estimation on the rate of convergence there is asymptotic and seems far behind its practical performance. Moreover, the results there were obtained under some additional *critical face conditions* [1]. The primal-dual feedback, which plays an important role in the practical performance of the algorithm, has no effect in the derivation of these theoretical estimations.

In this paper, we propose new variants for the algorithm and prove superior results on the rate of convergence. In Section 2, we propose a second update scheme for the iterates, where the primal-dual feedback is arranged in a new pattern. We also give "fixed" or "adaptive" step length rules as alternatives to the "perfect line search" used in the original version. All these variants, including the original version of the algorithm, are put in a unified framework. Then, in Section 3, we prove new linear convergence results for all these variants without the critical face conditions. The results are of global nature, and the estimates on the rates of convergence are much improved compared with the ones in [1]. For the variants with the new update scheme, sharper estimates for the rates are obtainable because of the new primal-dual feedback pattern. Finally, in Section 4, we discuss our numerical test results and other possible update schemes.

2. The Primal-Dual Steepest Descent Algorithm.

The family of primal-dual projected gradient algorithms in [1], as well as the finite-envelope algorithm developed earlier by Rockafellar and Wets [2–7] are all designed for solving large-scale extended linear-quadratic programming problems arising in multistage or stochastic optimization, where the problems exhibit the Lagrangian decomposability (or double decomposability) [7]. The latter term 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 $u \in U$. This is the case, for example, when the matrices P and Q are block diagonal, and the sets U and V are corresponding Cartesian products of polyhedra of low dimensions. These subproblems of maximization and minimization calculate not only the objective values $f(u)$ and $g(v)$ but also, in the fully quadratic case when L is strongly convex-concave, the uniquely determined vectors

$$F(u) = \operatorname*{argmax}_{v \in V} L(u,v) \quad \text{and} \quad G(v) = \operatorname*{argmin}_{u \in U} L(u,v). \tag{2.1}$$

The mappings F and G play a central role in the PDSD algorithm.

We cite from [7] and [1] several fundamental properties which are useful later in this paper. We write

$$\|w\|_M = [w \cdot Mw]^{\frac{1}{2}}$$

for the norm corresponding to a symmetric positive definite matrix M. It reduces to the ordinary Euclidean norm when M is the identity matrix. In this latter case, the subscript will be dropped. We use the related operator norm for matrices and use $[w_1, w_2]$ to denote the line segment between two points w_1 and w_2. We impose *the blanket assumption that the problem is fully quadratic* for the rest of the paper and refer consistently to

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

When the problem under consideration is not fully quadratic, an outer loop of *proximal point iteration* can be used to create fully quadratic inner loop problems. See [2], [7], and [11] for related discussions.

Let $P^{\frac{1}{2}}$ and $Q^{\frac{1}{2}}$ be the "square roots" of P and Q respectively defined by orthogonal factorization. Define

$$\gamma := \gamma(P,Q,R) := \|Q^{-\frac{1}{2}} R P^{-\frac{1}{2}}\|. \tag{2.2}$$

Proposition 2.1 [7] (optimality estimates). *Suppose u and v are elements of U and V satisfying $f(u) - g(v) \le \varepsilon$ for a certain $\varepsilon \ge 0$. Then u and v are ε-optimal in the sense that $|f(u) - f(\bar{u})| \le \varepsilon$ and $|g(v) - g(\bar{v})| \le \varepsilon$. Moreover,*

$$\|u - \bar{u}\|_P^2 + \|v - \bar{v}\|_Q^2 \le 2\varepsilon.$$

Proposition 2.2 [7] (regularity properties). *The functions f and g are continuously differentiable everywhere with*

$$\nabla f(u) = \nabla_u L(u, F(u)) \quad \text{and} \quad \nabla g(v) = \nabla_v L(G(v), v),$$

while the mappings F and G defined by (2.1) are Lipschitz continuous with

$$\|F(u') - F(u)\|_Q \le \gamma \|u' - u\|_P \text{ for all } u \text{ and } u',$$
$$\|G(v') - G(v)\|_P \le \gamma \|v' - v\|_Q \text{ for all } v \text{ and } v'.$$

Proposition 2.3 [7, 1] (modified gradient projection). *For arbitrary $u \in U$ and $v \in V$,*

$$G(F(u)) - u = P\text{-projection of } -\nabla_P f(u) \text{ on } U - u,$$
$$F(G(v)) - v = Q\text{-projection of } \nabla_Q g(v) \text{ on } V - v,$$

where $\nabla_P f(u) = P^{-1}\nabla f(u)$ symbolizes the gradient of f relative to the P-norm, while $\nabla_Q g(v) = Q^{-1}\nabla g(v)$ symbolizes the gradient of g relative to the Q-norm. Moreover, the vector $G(F(u)) - u$ is a feasible descent direction of f at u unless $u = \bar{u}$. Similarly, the vector $F(G(v)) - v$ is a feasible ascent direction of g at v unless $v = \bar{v}$.

The PDSD algorithm first searches on line segments $[u, G(F(u))]$ and $[v, F(G(v))]$ in primal and dual variables respectively to get some *intermediate points* as candidates for the next iterates. (Proposition 2.3 above suggests the name "projected gradient.") Then a novel kind of primal-dual feedback is incorporated in the updating. In the case of "forward feedback," the next iterates will be chosen between the intermediate points and their images under the mappings F and G, while in the case of "backward feedback," the next iterates will be chosen between the intermediate points and the images of the current iterates under the mappings F and G. This kind of interactive effect ties the primal and dual part of the operation closely, and has proven to be important to the performance of the algorithm.

In the following, we introduce new variants of PDSD algorithm. The second update scheme for the iterates corresponds to the backward feedback, for which a sharper bound for the rate of convergence is obtained. We also give alternatives for the "perfect line search" used in the original version. We put all these variants, including two different update schemes and three step length rules, in a unified framework. We refer to the algorithm with, say, update scheme 2 and step length rule (iii), as PDSD–2(iii). Under this convention, the PDSD algorithm in [1] is referred to as PDSD–1(i).

Primal-Dual Steepest Descent Algorithm.

Step 0 (initialization). *Set* $\nu := 0$ *(iteration counter). Specify starting points* $u^0 \in U$ *and* $v^0 \in V$*. Choose one of the step length rules in Step 2. (If rule (iii) is chosen, then also choose some constant* $\delta \in (0,1)$*, and let* $\alpha_{-1} = \beta_{-1} = 1$*.) Choose one of the update schemes in Step 3. Construct primal and dual sequences* $\{u^\nu\} \subset U$ *and* $\{v^\nu\} \subset V$ *as follows.*

Step 1 (optimality test). *If*

$$\min\{\, f(u^\nu), f\big(G(v^\nu)\big)\,\} - \max\{\, g(v^\nu), g\big(F(u^\nu)\big)\,\} = 0,$$

then terminate with

$$\bar{u} = \operatorname{argmin}\{\, f(u) \,|\, u = u^\nu, \ \text{ or } \ u = G(v^\nu)\,\}$$
$$\bar{v} = \operatorname{argmax}\{\, g(v) \,|\, v = v^\nu, \ \text{ or } \ v = F(u^\nu)\,\}$$

being optimal solutions to $(\mathcal{P})$ *and* $(\mathcal{Q})$*.*

Step 2 (line search). *Use one of the following step length rules chosen at initialization to determine* α_ν *and* β_ν *for generating intermediate points*

$$\hat{u}^{\nu+1} :=(1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu)),$$
$$\hat{v}^{\nu+1} :=(1-\beta_\nu)v^\nu + \beta_\nu F(G(v^\nu)),$$

in primal and dual variables respectively.

(i) *Perfect line search:*

$$\alpha_\nu := \operatorname*{argmin}_{\alpha \in [0,1]} f\big((1-\alpha)u^\nu + \alpha G(F(u^\nu))\big),$$
$$\beta_\nu := \operatorname*{argmax}_{\beta \in [0,1]} g\big((1-\beta)v^\nu + \beta F(G(v^\nu))\big).$$

(ii) *Fixed step lengths:*

$$\alpha_\nu := \min\{1, \frac{1}{2\gamma^2}\} \quad \text{and} \quad \beta_\nu := \min\{1, \frac{1}{2\gamma^2}\}.$$

(We adopt the convention $0^{-1} = +\infty$ in this paper.)

(iii) *Adaptive step lengths:*

$$\begin{aligned}\alpha_\nu := \max\big\{\, \alpha_{\nu-1}\delta^j \,\big|\, & f\big((1-\alpha_{\nu-1}\delta^j)u^\nu + \alpha_{\nu-1}\delta^j G(F(u^\nu))\big) - f(u^\nu) \\ & \le \big(f(u^\nu) - g(F(u^\nu))\big)(-\tfrac{1}{2}\alpha_{\nu-1}\delta^j),\ j \in \{0,1,2,\dots\}\,\big\},\\ \beta_\nu := \max\big\{\, \beta_{\nu-1}\delta^j \,\big|\, & g(v^\nu) - g\big((1-\beta_{\nu-1}\delta^j)v^\nu + \beta_{\nu-1}\delta^j F(G(v^\nu))\big) \\ & \le \big(f(G(v^\nu)) - g(v^\nu)\big)(-\tfrac{1}{2}\beta_{\nu-1}\delta^j),\ j \in \{0,1,2,\dots\}\,\big\}.\end{aligned}$$

Step 3 (update the iterates). *Use one of the following rules chosen at initialization to determine the next iterates.*

1. *Update with forward feedback:*

$$u^{\nu+1} := \operatorname{argmin}\{\, f(u) \,|\, u = \hat{u}^{\nu+1} \ \text{or}\ u = G(\hat{v}^{\nu+1})\,\},$$
$$v^{\nu+1} := \operatorname{argmax}\{\, g(v) \,|\, v = \hat{v}^{\nu+1} \ \text{or}\ v = F(\hat{u}^{\nu+1})\,\}.$$

(If both the arguments give the same objective value, use the first one in updating for decisiveness. The same rule applies also to the next set of formulas.)

2. *Update with backward feedback:*

$$u^{\nu+1} := \operatorname{argmin}\{\, f(u) \,|\, u = \hat{u}^{\nu+1} \ \text{or}\ u = G(v^{\nu})\,\},$$
$$v^{\nu+1} := \operatorname{argmax}\{\, g(v) \,|\, v = \hat{v}^{\nu+1} \ \text{or}\ v = F(u^{\nu})\,\}.$$

Then return to Step 1 with the counter ν increased by 1.

Observe that the primal-dual feedback also takes place in the optimality test. It follows from Proposition 2.2 that $F(u^\nu) \to \bar{v}$ and $G(v^\nu) \to \bar{u}$ as $u^\nu \to \bar{u}$ and

$v^\nu \to \bar{v}$. With the optimality test in Step 1, the algorithm will terminate if either $u^\nu = \bar{u}$ or $v^\nu = \bar{v}$ by Theorem 1.2.

In Step 2, there are three step length rules to choose from. By Theorem 1.1 and Proposition 2.2, the objective functions in the line searches are piecewise quadratic and continuously differentiable. In the typical decomposable case when P and Q are diagonal, and U and V are "boxes" representing upper and lower bounds, one can further get the explicit expressions for the derivatives of these functions. By taking advantage of all these properties, even the perfect line search will not be prohibitively difficult. In our numerical experimentations, the perfect line search takes approximately two-thirds of the time in each iteration.

An interesting result of Theorem 3.1 in next section is that the same estimated rate of convergence as for the perfect line search (i) can be reached by certain fixed step lengths in rule (ii). However the parameter γ of the problem, which determines the length of steps in (ii), is usually unavailable. Therefore we provide a third rule with adaptive step lengths, which resembles the Armijo stepsize rule for unconstrained minimization. However, we here use certain duality gap, instead of the slope of the line search function, in determining the step lengths. Theorem 3.2 in next section shows that the adaptive step length is well defined, that the step lengths will be fixed after a finite number of adaptations, and that an estimated rate of convergence very close to the one with perfect line search is obtainable.

Update scheme 1 in Step 3 can also be written as

$$u^{\nu+1} := \begin{cases} \hat{u}^{\nu+1}, & \text{if } f(\hat{u}^{\nu+1}) \le f(G(\hat{v}^{\nu+1})), \\ G(\hat{v}^{\nu+1}) & \text{otherwise,} \end{cases} \tag{2.3}$$

$$v^{\nu+1} := \begin{cases} \hat{v}^{\nu+1}, & \text{if } g(\hat{v}^{\nu+1}) \ge g(F(\hat{u}^{\nu+1})), \\ F(\hat{u}^{\nu+1}) & \text{otherwise.} \end{cases} \tag{2.4}$$

We say that there is an *interactive restart in the primal variable* if $u^{\nu+1} = G(\hat{v}^{\nu+1})$, in which case, the primal iterate is updated by using the dual information. Similarly, we say that there is an *interactive restart in the dual variable* if $v^{\nu+1} = F(\hat{u}^{\nu+1})$, in which case, the dual iterate is updated by using the primal information. Update scheme 2 can be written in the same manner as

$$u^{\nu+1} := \begin{cases} \hat{u}^{\nu+1}, & \text{if } f(\hat{u}^{\nu+1}) \le f(G(v^{\nu})), \\ G(v^{\nu}) & \text{otherwise.} \end{cases} \tag{2.5}$$

$$v^{\nu+1} := \begin{cases} \hat{v}^{\nu+1}, & \text{if } g(\hat{v}^{\nu+1}) \ge g(F(u^{\nu})), \\ F(u^{\nu}) & \text{otherwise,} \end{cases} \tag{2.6}$$

with the interactive restarts defined accordingly. Although the practical performance of the algorithm with these two different update schemes are very close in our tests, a sharper bound for the rate of convergence of the algorithm with scheme 2 will be obtained in the next section.

To conclude Section 2, we give a lemma that will be used later in deriving convergence results. The proof of the lemma follows closely the idea in the proofs of Rockafellar and Wets [2, Proposition 3 and Theorem 5].

Lemma 2.4. *For any $u \in U$,*

$$f\big((1-\alpha)u + \alpha G(F(u))\big) - f(u) \le \big(f(u) - g(F(u))\big)(-\alpha + \gamma^2\alpha^2\big) \tag{2.7}$$

for all $\alpha \in [0,1]$. Similarly, for any $v \in V$,

$$g(v) - g\big((1-\beta)v + \beta F(G(v))\big) \le \big(f(G(v)) - g(v)\big)(-\beta + \gamma^2\beta^2\big) \tag{2.8}$$

for all $\beta \in [0,1]$.

Proof. For any $u_0 \in U$, denote $v_1 := F(u_0)$ and $u_2 := G(v_1)$. Then the Lagrangian $L(u,v)$ can be written in the expanded form at (u, v_1) as

$$L(u,v) = L(u,v_1) + \nabla_v L(u,v_1)\cdot(v - v_1) - \tfrac{1}{2}(v - v_1)\cdot Q(v - v_1),$$

where the term $\nabla_v L(u,v_1)\cdot(v - v_1)$ can be further written as

$$\nabla_v L(u,v_1)\cdot(v - v_1) = \nabla_v L(u_0,v_1)\cdot(v - v_1) - (v - v_1)\cdot R(u - u_0).$$

Note that $v_1 = F(u_0)$ means v_1 is the argmax of $L(u_0, v)$ on V, which in turn implies $\nabla_v L(u_0,v_1)\cdot(v - v_1) \le 0$ for all $v \in V$. Hence

$$L(u,v) \le L(u,v_1) - (v - v_1)\cdot R(u - u_0) - \tfrac{1}{2}(v - v_1)\cdot Q(v - v_1). \tag{2.9}$$

Now for any $u \in [u_0, u_2]$ and $v = F(u)$, it follows from (2.9) that

$$\begin{aligned}
&L\big(u, F(u)\big) - L(u, v_1) \\
&\le -\big(F(u) - v_1\big)\cdot R(u - u_0) - \tfrac{1}{2}\big(F(u) - v_1\big)\cdot Q\big(F(u) - v_1\big) \\
&\le \max_{w \in \mathbb{R}^m} \{w\cdot R(u - u_0) - \tfrac{1}{2} w\cdot Qw\} \\
&= \tfrac{1}{2}(u - u_0)\cdot\big(R^T Q^{-1} R\big)(u - u_0) \\
&= \tfrac{1}{2}\|(Q^{-\frac{1}{2}} R P^{-\frac{1}{2}}) P^{\frac{1}{2}}(u - u_0)\|^2 \\
&\le \tfrac{1}{2}\gamma^2 \|u - u_0\|_P^2
\end{aligned} \tag{2.10}$$

However, $L\big(u, F(u)\big) = f(u)$ and

$$\begin{aligned} L\big((1-\alpha)u_0 + \alpha u_2, v_1\big) &\le (1-\alpha)L(u_0, v_1) + \alpha L(u_2, v_1) \\ &= (1-\alpha)f(u_0) + \alpha g(v_1) \end{aligned}$$

for $0 \le \alpha \le 1$. Thus, by taking $u = (1-\alpha)u_0 + \alpha u_2$ in (2.10), we get

$$f\big((1-\alpha)u_0 + \alpha u_2\big) - f(u_0) + \alpha\big(f(u_0) - g(v_1)\big) \le \tfrac{1}{2}\alpha^2\gamma^2\|u_2 - u_0\|_P^2. \qquad (2.11)$$

On the other hand,

$$\begin{aligned} f(u_0) - g(v_1) &= L(u_0, v_1) - L(u_2, v_1) \\ &= \nabla_u L(u_2, v_1)\cdot(u_0 - u_2) + \tfrac{1}{2}(u_0 - u_2)\cdot P(u_0 - u_2) \end{aligned}$$

by the definition of v_1 and u_2. Observe that $\nabla_u L(u_2, v_1)\cdot(u_0 - u_2) \ge 0$ since u_2 is the argmin of $L(u, v_1)$ on U. Therefore

$$f(u_0) - g(v_1) \ge \tfrac{1}{2}(u_0 - u_2)\cdot P(u_0 - u_2) = \tfrac{1}{2}\|u_2 - u_0\|_P^2. \qquad (2.12)$$

Combining (2.11) and (2.12), we get

$$f\big(u_0 + \alpha(u_2 - u_0)\big) - f(u_0) \le \big(f(u_0) - g(v_1)\big)(-\alpha + \gamma^2\alpha^2)$$

for $0 \le \alpha \le 1$, which yields (2.7). One can prove (2.8) similarly. □

3. Global Linear Convergence of the PDSD Algorithm.

In this section, we prove linear convergence results for all the six variants of the PDSD algorithm formulated in Section 2. We first give results for the algorithms with (i) perfect line search and (ii) fixed step lengths. Define the function $\theta : [0, +\infty) \to (0, 1)$ as

$$\theta(s) = \begin{cases} 1 - s & \text{if } s < \frac{1}{2}, \\ \frac{1}{4s} & \text{if } s \ge \frac{1}{2}. \end{cases} \qquad (3.1)$$

Theorem 3.1 (convergence of PDSD with step length rules (i) and (ii)).

(a) *The sequences $\{f(u^\nu)\}$ and $\{g(v^\nu)\}$ generated by PDSD–1(i) or PDSD–1(ii) converge linearly to the common optimal value $f(\bar{u}) = g(\bar{v})$ in the sense that*

$$f(u^{\nu+1}) - f(\bar{u}) \le \big(1 - \theta(\gamma^2)\big)\big(f(u^\nu) - f(\bar{u})\big), \qquad (3.2)$$

$$g(\bar{v}) - g(v^{\nu+1}) \le \big(1 - \theta(\gamma^2)\big)\big(g(\bar{v}) - g(v^\nu)\big). \qquad (3.3)$$

Moreover,

$$\|u^{\nu+1}-\bar{u}\|_P^2+\|v^{\nu+1}-\bar{v}\|_Q^2 \le 2\big(1-\theta(\gamma^2)\big)^{\nu+1}\big(f(u^0)-g(u^0)\big). \tag{3.4}$$

(b) *The sequences* $\{f(u^\nu)\}$ *and* $\{g(v^\nu)\}$ *generated by PDSD–2(i) or PDSD–2(ii) converge linearly to the common optimal value* $f(\bar{u})=g(\bar{v})$ *in the sense that*

$$f(u^{\nu+1})-g(v^{\nu+1}) \le \frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)}\big(f(u^\nu)-g(v^\nu)\big). \tag{3.5}$$

Moreover,

$$\|u^{\nu+1}-\bar{u}\|_P^2+\|v^{\nu+1}-\bar{v}\|_Q^2 \le 2\Big(\frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)}\Big)^{\nu+1}\big(f(u^0)-g(u^0)\big). \tag{3.6}$$

Proof. It follows from (2.7) that

$$f\big((1-\alpha)u^\nu+\alpha G(F(u^\nu))\big)-f(u^\nu) \le \big(f(u^\nu)-g(F(u^\nu))\big)(-\alpha+\gamma^2\alpha^2) \tag{3.7}$$

for all $\alpha\in[0,1]$. But $\min\{\,-\alpha+\gamma^2\alpha^2\mid 0\le\alpha\le 1\,\}=-\theta(\gamma^2)$ with

$$\operatorname{argmin}\{\,-\alpha+\gamma^2\alpha^2\mid 0\le\alpha\le 1\,\}=\min\{1,\frac{1}{2\gamma^2}\}.$$

Hence for the fixed step length $\alpha_\nu=\min\{1,\frac{1}{2\gamma^2}\}$ in rule (ii),

$$f\big((1-\alpha_\nu)u^\nu+\alpha_\nu G(F(u^\nu))\big)-f(u^\nu) \le \big(f(u^\nu)-g(F(u^\nu))\big)\big(-\theta(\gamma^2)\big). \tag{3.8}$$

Obviously (3.8) is also true for the step length $\alpha_\nu=\operatorname*{argmin}_{\alpha\in[0,1]} f\big((1-\alpha)u^\nu+\alpha G(F(u^\nu))\big)$ in rule (i), since the perfect line search should not make the first term of (3.8) any larger. According to the update scheme in Step 3, we have $f(u^{\nu+1})\le f((1-\alpha_\nu)u^\nu+\alpha_\nu G(F(u^\nu)))$. Therefore

$$f(u^{\nu+1})-f(u^\nu) \le \big(f(u^\nu)-g(F(u^\nu))\big)\big(-\theta(\gamma^2)\big). \tag{3.9}$$

Similarly it follows from (2.8) that

$$g(v^\nu)-g\big((1-\beta)v^\nu+\beta F(G(v^\nu))\big) \le \big(f(G(v^\nu))-g(v^\nu)\big)(-\beta+\gamma^2\beta^2) \tag{3.10}$$

for all $\beta \in [0,1]$, which yields

$$g(v^\nu) - g(v^{\nu+1}) \le \big(f(G(v^\nu)) - g(v^\nu)\big)\big(-\theta(\gamma^2)\big). \tag{3.11}$$

Combining (3.9) and (3.11), we get

$$f(u^\nu) - g(v^\nu) - f(u^{\nu+1}) + g(v^{\nu+1}) \ge \theta(\gamma^2)\big(f(u^\nu) - g(v^\nu) - g(F(u^\nu)) + f(G(v^\nu))\big). \tag{3.12}$$

With the νth *duality gap* ε_ν and the νth *auxiliary duality gap* $\tilde{\varepsilon}_\nu$ defined as

$$\varepsilon_\nu := f(u^\nu) - g(v^\nu) \text{ and } \tilde{\varepsilon}_\nu := f(G(v^\nu)) - g(F(u^\nu)) \tag{3.13}$$

respectively, (3.12) can be written in the form

$$\varepsilon_\nu - \varepsilon_{\nu+1} \ge \theta(\gamma^2)(\varepsilon_\nu + \tilde{\varepsilon}_\nu),$$

or equivalently,

$$\varepsilon_{\nu+1} \le \big(1 - \theta(\gamma^2)\big)\varepsilon_\nu - \theta(\gamma^2)\tilde{\varepsilon}_\nu. \tag{3.14}$$

If update scheme 2 is used in Step 3 of the algorithm, then $f(u^{\nu+1}) \le f(G(v^\nu))$ and $g(v^{\nu+1}) \ge g(F(u^\nu))$. Hence $\varepsilon_{\nu+1} \le \tilde{\varepsilon}_\nu$. Therefore (3.14) implies

$$\varepsilon_{\nu+1} \le \big(1 - \theta(\gamma^2)\big)\varepsilon_\nu - \theta(\gamma^2)\varepsilon_{\nu+1},$$

from which (3.5) follows. Using (3.5) for $\nu = 0, 1, \dots,$ we get

$$f(u^{\nu+1}) - g(v^{\nu+1}) \le \Big(\frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)}\Big)^{\nu+1}\big(f(u^0) - g(v^0)\big),$$

which yields (3.6) by Proposition 2.1.

If update scheme 1 is used in Step 3 of the algorithm, then the relation $\varepsilon_{\nu+1} \le \tilde{\varepsilon}_\nu$ is not necessarily ture. However, by Theorem 1.2,

$$f(u) \ge f(\bar{u}) = g(\bar{v}) \ge g(v) \quad \text{for all } u \in U,\ v \in V.$$

Hence it follows from (3.9) and (3.11) that

$$f(u^{\nu+1}) - f(u^\nu) \le \big(f(u^\nu) - f(\bar{u})\big)\big(-\theta(\gamma^2)\big),$$
$$g(v^\nu) - g(v^{\nu+1}) \le \big(g(\bar{v}) - g(v^\nu)\big)\big(-\theta(\gamma^2)\big).$$

These two inequalities yield (3.2) and (3.3) respectively. Moreover, observe that $\tilde{\varepsilon}_\nu \geq 0$. Hence by (3.14), we have

$$f(u^{\nu+1}) - g(v^{\nu+1}) \leq \big(1-\theta(\gamma^2)\big)\big(f(u^\nu) - g(v^\nu)\big). \tag{3.15}$$

Using (3.15) for $\nu = 0, 1, \ldots,$ we get

$$f(u^{\nu+1}) - g(v^{\nu+1}) \leq \big(1-\theta(\gamma^2)\big)^{\nu+1}\big(f(u^0) - g(v^0)\big),$$

which yields (3.4) by Proposition 2.1. □

Next we give convergence results for the algorithm with adaptive step lengths (iii). We have to show, in the first place, that these step lengths are well defined. Let the function $\tilde{\theta} : [0, +\infty) \to (0, 1)$ be defined as

$$\tilde{\theta}(s) = \min\{\frac{1}{2}, \frac{1}{4s}\}. \tag{3.16}$$

Obviously $\theta(s) \geq \tilde{\theta}(s)$ for all $s \in [0, +\infty)$, and the equality holds when $s \geq \frac{1}{2}$.

Theorem 3.2 (convergence of PDSD with step length rule (iii)).

(a) *For any choice of $\delta \in (0, 1)$, the step lengths α_ν and β_ν in the PDSD algorithm with rule (iii) are well defined. Both α_ν and β_ν are nonincreasing as ν increases, and*

$$\alpha_\nu > \delta \min\{1, \frac{1}{2\gamma^2}\} \quad \text{and} \quad \beta_\nu > \delta \min\{1, \frac{1}{2\gamma^2}\} \tag{3.17}$$

for all ν. Moreover, both α_ν and β_ν will be fixed after a finite number of iterations.

(b) *The sequences $\{f(u^\nu)\}$ and $\{g(v^\nu)\}$ generated by PDSD–1(iii) converge linearly to the common optimal value $f(\bar{u}) = g(\bar{v})$ in the sense that*

$$f(u^{\nu+1}) - f(\bar{u}) \leq \big(1-\delta\tilde{\theta}(\gamma^2)\big)\big(f(u^\nu) - f(\bar{u})\big), \tag{3.18}$$

$$g(\bar{v}) - g(v^{\nu+1}) \leq \big(1-\delta\tilde{\theta}(\gamma^2)\big)\big(g(\bar{v}) - g(v^\nu)\big). \tag{3.19}$$

Moreover,

$$\|u^{\nu+1} - \bar{u}\|_P^2 + \|v^{\nu+1} - \bar{v}\|_Q^2 \leq 2\big(1-\delta\tilde{\theta}(\gamma^2)\big)^{\nu+1}\big(f(u^0) - g(u^0)\big). \tag{3.20}$$

(c) *The sequences* $\{f(u^\nu)\}$ *and* $\{g(v^\nu)\}$ *generated by PDSD–2(iii) converge linearly to the common optimal value* $f(\bar u) = g(\bar v)$ *in the sense that*

$$f(u^{\nu+1}) - g(v^{\nu+1}) \le \frac{1-\delta\tilde\theta(\gamma^2)}{1+\delta\tilde\theta(\gamma^2)}\big(f(u^\nu) - g(v^\nu)\big), \tag{3.21}$$

Moreover

$$\|u^{\nu+1} - \bar u\|_P^2 + \|v^{\nu+1} - \bar v\|_Q^2 \le 2\Big(\frac{1-\delta\tilde\theta(\gamma^2)}{1+\delta\tilde\theta(\gamma^2)}\Big)^{\nu+1}\big(f(u^0) - g(u^0)\big). \tag{3.22}$$

Proof. First, we claim that for all nonnegative $\alpha \le \min\{1, \frac{1}{2\gamma^2}\}$,

$$f\big((1-\alpha)u^\nu + \alpha G(F(u^\nu))\big) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)\big(\frac{-\alpha}{2}\big). \tag{3.23}$$

This follows directly from (2.7) and the fact that

$$-\alpha + \gamma^2\alpha^2 \le \frac{-\alpha}{2} \quad \text{for all } 0 \le \alpha \le \min\{1, \frac{1}{2\gamma^2}\}.$$

Hence the step length $\alpha_\nu = \alpha_{\nu-1}\delta^j$ in rule (iii), where j is the *first* element in the ordered nonnegative integer set $\{0, 1, 2, \ldots\}$ satisfying

$$f\big((1-\alpha_{\nu-1}\delta^j)u^\nu + \alpha_{\nu-1}\delta^j G(F(u^\nu))\big) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)(-\tfrac{1}{2}\alpha_{\nu-1}\delta^j), \tag{3.24}$$

is well defined. Obviously $\{\alpha_\nu\}$ is nonincreasing.

According to the claim and the step rule, we have either $\alpha_\nu = \alpha_{\nu-1}$ or $\alpha_{\nu-1}\delta^{j-1} > \min\{1, \frac{1}{2\gamma^2}\}$ with $j \ge 1$, because otherwise $\alpha_{\nu-1}\delta^{j-1}$ instead of $\alpha_{\nu-1}\delta^j$ will be taken as the step length α_ν. Suppose $\alpha_{\nu-1} > \delta\min\{1, \frac{1}{2\gamma^2}\}$. Then in either case,

$$\alpha_\nu = \alpha_{\nu-1}\delta^j > \delta\min\{1, \frac{1}{2\gamma^2}\}. \tag{3.25}$$

Note that $\alpha_{-1} = 1 > \delta$. This proves the first inequality in (3.17) by induction. The second inequality in (3.17) regarding β_ν can be proved similarly, and the last conclusion in part (a) is now obvious.

Combining (3.24) and (3.25), we have

$$f\big((1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu))\big) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)\big(\frac{-\delta}{2}\min\{1, \frac{1}{2\gamma^2}\}\big).$$

Therefore, by observing $f(u^{\nu+1}) \le f\big((1-\alpha_\nu)u^\nu + \alpha_\nu G(F(u^\nu))\big)$ in the updating, we get

$$f(u^{\nu+1}) - f(u^\nu) \le \big(f(u^\nu) - g(F(u^\nu))\big)\big(-\delta\tilde{\theta}(\gamma^2)\big). \tag{3.26}$$

Similarly, we have

$$g(v^\nu) - g(v^{\nu+1}) \le \big(f(G(v^\nu)) - g(v^\nu)\big)\big(-\delta\tilde{\theta}(\gamma^2)\big). \tag{3.27}$$

Now (3.26) and (3.27) lead to the conclusions in (b) and (c) in the same manner as (3.9) and (3.11) lead to the conclusions in Theorem 3.1. □

Theorems 3.1 and 3.2 provide global linear convergence results for all the variants of the PDSD algorithm formulated in Section 2 without any additional assumptions. The parameter $\gamma = \|Q^{-\frac{1}{2}}RP^{-\frac{1}{2}}\|$ of the problem plays an important role in the estimations regarding the rates of convergence of the algorithm. It also characterizes the Lipschitzian constant for the mappings F and G in Proposition 2.2. In fact, γ can be viewed as a normalized measure of the "coupling" between the primal and dual variables of the problem. In the extremal case when $\gamma = 0$ (which implies $R = 0$), we have $F(u) = \bar{v}$ for all u and $G(v) = \bar{u}$ for all v. Hence the algorithm will terminate in one iteration. On the other hand, *a large γ implies a difficult problem* for the algorithm.

It follows from Theorem 3.1 that for problems with large γ, the duality gap $\varepsilon_\nu = f(u^\nu) - g(v^\nu)$ of the iterates generated by PDSD–1(i) or PDSD–1(ii) decreases at least with the ratio

$$1 - \theta(\gamma^2) = 1 - \frac{1}{4\gamma^2}, \tag{3.28}$$

while the one generated by PDSD–2(i) or PDSD–2(ii) decreases at least with the ratio

$$1 - \frac{1-\theta(\gamma^2)}{1+\theta(\gamma^2)} \sim 1 - \frac{1}{2\gamma^2}. \tag{3.29}$$

These are much improved estimates compared with the earlier results in [1, Theorem 4.2] with an asymptotic ratio

$$1 - \frac{1}{4(\gamma^2+1)^4 + 5(\gamma^2+1)^2 + 2(\gamma^2+1)} \sim 1 - \frac{1}{4\gamma^8}$$

under the critical face conditions. However, if the iterates eventually reach the corresponding critical faces, the technique in [1, Theorem 4.2] still gives a better asymptotic ratio

$$\Big(1 - \frac{1}{0.5(\gamma^2+1)+0.5}\Big)^2 \sim 1 - \frac{1}{0.25\gamma^2} \text{ (for large } \gamma)$$

under the perfect line search. This is consistent with the observation that the algorithm with perfect line search often gives better per-step progress towards the end of iteration than other line search rules in our numerical tests.

The fixed step length in rule (ii) is related to the parameter γ of the problem, which is usually unavailable. According to Theorem 3.2, the convergence ratios in Theorem 3.1 for problems with $\gamma^2 \geq \frac{1}{2}$ could be approached with the adaptive step lengths in rule (iii). Moreover, these step lengths will eventually be fixed after a finite number of iterations. Comparing the estimations in Theorem 3.2 with the ones in Theorem 3.1, one may get the impression that a choice of δ close to 1 would eventually give better per-step ratios. But such a choice will, at the same time, increase the number of trials in identifying the proper step length. Hence, in the practical implementation of rule (iii), one has to compromise between these two ends. One can also start the trial of j there with some negative integer instead of 0. Then the step length will be allowed to increase if a larger progress in the line search is possible.

4. Numerical Test Results and Other Update Schemes.

Although the estimated rates for PDSD–2 are better than the ones for PDSD–1, we find in our numerical tests that their practical performance are actually very close. For comparison, we have run PDSD–1(i) and PDSD–2(i) on the *transverse family* of the test problems 0.4 – 9.4 used in [1], where both the primal and the dual dimensions are 5140. The stopping criterion in the optimality test for the practical implementation of the algorithm is

$$\min\{\, f(u^\nu), f\big(G(v^\nu)\big)\,\} - \max\{\, g(v^\nu), g\big(F(u^\nu)\big)\,\} \leq \varepsilon, \tag{4.1}$$

where $\varepsilon > 0$ is a prespecified threshold for the duality gap. The results in terms of CPU times, as well as numbers of iterations, are given in Table 1. For instance, 45(8/6) in the iterations column of PDSD–2(i) for Problem 0.4 means that the algorithm terminates successfully in 45 iterations, with 8 interactive primal restarts and 6 interactive dual restarts during the process. (The tests are run on a DECstation 3100 with double precision, where the software has been updated since the test in [1].)

We also tried the algorithm without the primal-dual feedback in the update, namely, using

$$u^{\nu+1} := \hat{u}^{\nu+1} \quad \text{and} \quad v^{\nu+1} := \hat{v}^{\nu+1}$$

directly in the updating of Step 3. Then the algorithm generates two unrelated sequences in primal and dual variables respectively until the stopping criterion (4.1) on the duality gap is satisfied. We refer to this *extra* version for test purpose as PDSD–0. (In the case of perfect line search, one can prove by using [1, Proposition 5.1] that the dual part of this extra version reduces to a special case of the finite generation algorithm [2].) The corresponding results are put in the columns headed PDSD–0(i). The notation ** in these columns signifies that the algorithm failed to terminate in 100 iterations, in which case the figure for CPU time is preceded by * since it indicates only how long the first 100 iterations took. The test results show clearly the importance of the primal-dual feedback. Both PDSD–1(i) and PDSD–2(i) perform much better than PDSD–0(i).

Table 1. Test results on problems 0.4-9.4 [1]

		CPU time (sec.)			Iterations		
Problem	Size	PDSD–1(i)	PDSD–2(i)	PDSD–0(i)	PDSD–1(i)	PDSD–2(i)	PDSD–0(i)
0.4	5140	110	141	*337	32(7/6)	45(8/6)	**
1.4	5140	183	172	*356	52(4/6)	50(3/6)	**
2.4	5140	147	224	*341	42(8/4)	67(10/3)	**
3.4	5140	35	42	212	9(4/4)	13(3/3)	68
4.4	5140	72	72	*346	19(7/4)	22(6/4)	**
5.4	5140	51	66	178	13(6/4)	20(7/2)	52
6.4	5140	62	74	82	16(5/7)	23(7/7)	24
7.4	5140	64	72	92	18(8/3)	22(6/3)	28
8.4	5140	189	180	*341	55(5/5)	54(3/4)	**
9.4	5140	62	65	110	17(6/7)	20(4/1)	35

There are other possible variants for the algorithm. Notice that the iteration of PDSD–2(i) can be written as

$$u^{\nu+1} := \operatorname{argmin}\{\, f(u) \mid u \in [u^\nu, G(F(u^\nu))] \ \text{ or } \ u \in G(v^\nu)\,\},$$
$$v^{\nu+1} := \operatorname{argmax}\{\, g(v) \mid v \in [v^\nu, F(G(v^\nu))] \ \text{ or } \ v \in F(u^\nu)\,\}.$$

This suggests a third update scheme with four perfect line searches in each iteration

$$u^{\nu+1} := \operatorname{argmin}\{\, f(u) \mid u \in [u^\nu, G(F(u^\nu))] \ \text{ or } \ u \in [G(v^\nu), G(F(G(v^\nu)))]\,\}, \tag{4.2}$$
$$v^{\nu+1} := \operatorname{argmax}\{\, g(v) \mid v \in [v^\nu, F(G(v^\nu))] \ \text{ or } \ v \in [F(u^\nu), F(G(F(u^\nu)))]\,\}. \tag{4.3}$$

Obviously, it should converge at least as fast as PDSD–2(i).

Recall that the intermediate points resulted from line searches on $[u^\nu, G(F(u^\nu))]$ and $[v^\nu, F(G(v^\nu))]$ are denoted by $\hat{u}^{\nu+1}$ and $\hat{u}^{\nu+1}$ respectively. Let $\tilde{u}^{\nu+1}$ and $\tilde{u}^{\nu+1}$ be the corresponding line search results in primal and dual on $[G(v^\nu), G(F(G(v^\nu)))]$ and $[F(u^\nu), F(G(F(u^\nu)))]$ respectively. With a reasoning similar to the one that leads to (3.8), we are able to get

$$f(u^\nu) - f(\hat{u}^{\nu+1}) \ge \big(f(u^\nu) - g(F(u^\nu))\big)\theta(\gamma^2), \tag{4.4}$$
$$g(\hat{u}^{\nu+1}) - g(u^\nu) \ge \big(f(G(v^\nu)) - g(v^\nu)\big)\theta(\gamma^2), \tag{4.5}$$
$$f(G(v^\nu)) - f(\tilde{u}^{\nu+1}) \ge \big(f(G(v^\nu)) - g(F(G(v^\nu)))\big)\theta(\gamma^2), \tag{4.6}$$
$$g(\tilde{u}^{\nu+1}) - g(F(v^\nu)) \ge \big(f(G(F(u^\nu))) - g(F(u^\nu))\big)\theta(\gamma^2). \tag{4.7}$$

Now (4.4) and (4.5) yield

$$\begin{aligned} f(\hat{u}^{\nu+1}) - g(\hat{v}^{\nu+1}) \le& \big(1-\theta(\gamma^2)\big)\big(f(u^\nu) - g(v^\nu)\big) \\ &- \theta(\gamma^2)\big(f(G(v^\nu)) - g(F(u^\nu))\big), \end{aligned} \tag{4.8}$$

while (4.5) and (4.6) yield

$$\begin{aligned} f(\tilde{u}^{\nu+1}) - g(\tilde{v}^{\nu+1}) \le& \big(1-\theta(\gamma^2)\big)\big(f(G(v^\nu)) - g(F(u^\nu))\big) \\ &- \theta(\gamma^2)\big(f(G(F(u^\nu))) - g(F(G(v^\nu)))\big). \end{aligned} \tag{4.9}$$

Eliminating the term $f(G(v^\nu)) - g(F(u^\nu))$ in (4.8) and (4.9), we get

$$\begin{aligned} &\big(1-\theta(\gamma^2)\big)\big(f(\hat{u}^{\nu+1}) - g(\hat{v}^{\nu+1})\big) + \theta(\gamma^2)\big(f(\tilde{u}^{\nu+1}) - g(\tilde{v}^{\nu+1})\big) \\ &\le \big(1-\theta(\gamma^2)\big)^2\big(f(u^\nu) - g(v^\nu)\big) - \theta(\gamma^2)\big(f(G(F(u^\nu))) - g(F(G(v^\nu)))\big). \end{aligned} \tag{4.10}$$

According to the update scheme, the duality gap $\varepsilon_{\nu+1}$ should be no larger than either $f(\hat{u}^{\nu+1}) - g(\hat{v}^{\nu+1})$ or $f(\tilde{u}^{\nu+1}) - g(\tilde{v}^{\nu+1})$ or $f(G(F(u^\nu))) - g(F(G(v^\nu)))$. Hence we obtain an estimate

$$\frac{\varepsilon_{\nu+1}}{\varepsilon_\nu} \leq \frac{\big(1-\theta(\gamma^2)\big)^2}{1+\big(\theta(\gamma^2)\big)^2}$$

from (4.10) for the third update scheme in (4.2) and (4.3). For problems with large γ, this is a slightly better result compared with (3.5) for PDSD–2(i) at the cost of two additional line searches.

Acknowledgments. The author thanks two anonymous referees for their very helpful comments and suggestions. The third update scheme in (4.2) and (4.3) was due to one of them.

REFERENCES.

1. C. Zhu and R. T. Rockafellar, "Primal-dual projected gradient algorithms for extended linear-quadratic programming," SIAM J. Opt. 3 (1993), pp. 751–783.

2. 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.

3. 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.

4. 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.

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

6. 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.

7. R. T. Rockafellar, "Computational schemes for solving large-scale problems in extended linear-quadratic programming," Math. Programming 48 (1990), pp. 447-474.

8. 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, and R. Tapia, eds.), SIAM, 1990.

9. 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.

10. J. M. Wagner, *Stochastic Programming with Recourse Applied to Groundwater Quality Management*, doctoral dissertation, MIT, 1988.

11. C. Zhu, "Modified proximal point algorithm for extended linear-quadratic programming," Computational Opt. and Applications 1 (1992), pp. 185–205.