arXiv:1601.01174v1 [math.OC] 6 Jan 2016

THE SUPPORTING HALFSPACE - QUADRATIC PROGRAMMING STRATEGY FOR THE DUAL OF THE BEST APPROXIMATION PROBLEM

C.H. JEFFREY PANG

ABSTRACT. We consider the best approximation problem (BAP) of projecting a point onto the intersection of a number of convex sets. It is known that Dykstra's algorithm is alternating minimization on the dual problem. We extend Dykstra's algorithm so that it can be enhanced by the SHQP strategy of using quadratic programming to project onto the intersection of supporting halfspaces generated by earlier projection operations. By looking at a structured alternating minimization problem, we show the convergence rate of Dykstra's algorithm when reasonable conditions are imposed to guarantee a dual minimizer. We also establish convergence of using a warmstart iterate for Dykstra's algorithm, show how all the results for the Dykstra's algorithm can be carried over to the simultaneous Dykstra's algorithm, and discuss a different way of incorporating the SHQP strategy. Lastly, we show that the dual of the best approximation problem can have an $O(1/k^2)$ accelerated algorithm that also incorporates the SHQP strategy.

CONTENTS

1. INTRODUCTION

We consider the following problem, known as the best approximation problem (BAP).

Date: January 7, 2016.
2010 *Mathematics Subject Classification.* 41A50, 90C25, 68Q25, 47J25.
Key words and phrases. alternating minimization, Dykstra's algorithm, best approximation problem.

$$
\begin{aligned}
(BAP) \quad & \min \quad f(x) := \frac{1}{2}\|x - d\|^2 \\
& \text{s.t.} \quad\ \ x \in C := C_1 \cap \cdots \cap C_m,
\end{aligned}
\tag{1.1}
$$

where d is a given point and C_i, $i = 1, \ldots, m$, are closed convex sets in a Hilbert space X. The BAP is equivalent to projecting d onto C. We shall assume throughout that $C \neq \emptyset$.

We now recall some background on first order methods, alternating minimization, and algorithms for the best approximation problem.

1.1. **First order methods and alternating minimization.** When presented with a problem with a large number of variables, first order methods (which use gradient descent and avoid computationally expensive operations like solving linear systems) and other methods that decompose the large problems into smaller pieces to be solved may be the only practical alternative.

For these algorithms, the nonasymptotic or absolute rate of convergence of the function values to the optimal objective value hold right from the very first iteration of the algorithm, and are more useful than the asymptotic rates. These rates are typically sublinear, like $O(1/k)$ for example. Classical references on first order methods include [NY83], and newer references include [Nes04, JN11a, JN11b].

As explained in [NY83, Nes04], the nonasymptotic rates of convergence of first order algorithms for smooth convex functions is at best $O(1/k^2)$. Nesterov proposed various $O(1/k^2)$ nonasymptotic methods (which are thus optimal) for such problems (first method [Nes83], second method [Nes88] and third method [Nes05]), and other optimal methods were studied in [AT06, BT09, LLM11]. The paper [BT09] also described an $O(1/k^2)$ algorithm for solving the sum of a smooth convex function and a structured nonsmooth function. These optimal methods are also known as accelerated proximal gradient (APG) methods, and their design and analysis are unified in the paper [Tse08] (who also dealt with convex-concave minimization).

An optimization problem with a large number of variables can have its variables divided into a number of blocks so that each subproblem has fewer variables. These subproblems are solved in some order (often in a cyclic manner) while the variables in other blocks are kept fixed. See the formula (2.3) for an elaboration. This is referred to as alternating minimization (AM), and sometimes referred to Cyclic Coordinate Minimization (CCM). Another alternative is to perform only gradient descent on each block, which would reduce to what is described as Cyclic Coordinate Descent, CCD.

Methods like AM and CCD are quite old. If a function to be minimized were nonsmooth, then it is possible for the AM and CCD to be stuck at a non-optimal solution. A $O(1/k)$ rate of convergence for AM of a two block problem was established in [Bec15] without any assumption of strong convexity. As mentioned in [Bec15], AM is also known in the literature as block-nonlinear Gauss–Seidel method or the block coordinate descent method (see for example [Ber99]). They also state the other contributions of [Aus76, BT13, Ber99, GS99, LT93].

There has been much recent research on stochastic/ randomized CD. Since we are not dealing with stochastic CD in this paper, we shall only mention the papers [Nes12, FR15], and defer to the introduction and tables in [FR15] for a summary of stochastic CD. A recent work [CP15] also identifies some cases where the deterministic CD scheme can have an $O(1/k^2)$ acceleration.

1.2. **The best approximation problem and the method of alternating projections.** The BAP is often associated with the set intersection problem (SIP)

$$(SIP) \quad \text{Find} \quad x \in C := C_1 \cap \cdots \cap C_m.$$

A well studied method for the SIP is the method of alternating projections (MAP). We recall material from [BC11, Deu01a, Deu01b, ER11] on material on the MAP. As its name suggests, the MAP projects the iterates in a cyclic or some other manner so that the iterates converge to a point in the intersection of these sets.

One acceleration of the MAP for convex problems is the supporting halfspace and quadratic programming strategy (SHQP): The projection process generates supporting halfspaces of each C_i, and the set C is a subset of the polyhedron obtained by intersecting these halfspaces. Projecting onto the polyhedron can accelerate the convergence of the MAP, and may lead to superlinear convergence in small problems. The SHQP strategy was discussed in [Pan15c]. See Figure 4.1 for an illustration. This idea was discussed in less generality in [BCK06] and other papers. Other methods of accelerating the MAP include [GPR67, GK89, BDHP03].

As remarked by several authors, the MAP does not converge to the solution of the BAP. Dykstra's algorithm [Dyk83] solves the best approximation problem through a sequence of projections onto each of the sets in a manner similar to the MAP, but correction vectors are added before every projection. The proof of convergence to $P_C(d)$ was established in [BD85] and sometimes referred to as the Boyle-Dykstra theorem. Dykstra's algorithm was rediscovered by [Han88], who showed that Dykstra's algorithm is equivalent to AM on the dual problem. See also [GM89]. When the sets C_i are halfspaces, the convergence is asymptotically linear [DH94]. A nonasymptotic $O(1/k)$ convergence rate of Dykstra's algorithm was obtained in [CP15] using the methods similar to [BT13, Bec15] when a dual minimizer exists. (This does not diminish the significance of the Boyle-Dykstra theorem. In our opinion, a quick glance at the respective proofs shows that the Boyle-Dykstra theorem, which proves the convergence to the primal optimal $P_C(d)$ even when a dual minimizer does not exist, is technically more sophisticated than the proof of the $O(1/k)$ convergence rate of Dykstra's algorithm when a dual minimizer exists.) Dykstra's algorithm is quite old, so we refer the reader to the commentary in [Deu01b, ER11] for more on previous work on Dykstra's algorithm.

In contrast to the situation for the MAP, not much has been done on accelerating Dykstra's algorithm for the BAP. The question of how to accelerate Dykstra's algorithm has been explicitly posed as an open problem in [Deu01b, Deu01a, ER11]. A method was proposed in [LR15]. The only property of Dykstra's algorithm needed for their acceleration is that Dykstra's algorithm generates a sequence converging to $P_C(d)$, so there is still be room for improving Dykstra's algorithm. See also [HS15].

A variant of Dykstra's algorithm that is more suitable for parallel computations is the simultaneous Dykstra's algorithm proposed in [IP91] using the product space formulation of [Pie84].

Some specific best approximation problems can be solved with specialized methods. The projection of a point into the intersection of halfspaces can be solved by classical methods of quadratic programming. Other sets for which the projection onto the intersection is easy include the intersection of an affine space and the semidefinite cone [QS06, Mal04].

The subgradient algorithm can be used to solve a convex constrained optimization problem with a convergence rate of $O(1/\sqrt{k})$. Hence the BAP can be solved at a rate of $O(1/\sqrt{k})$. See [Nes04]. In [Pan15b], we obtained a convergence rate of $O(1/k)$ in the case when the objective function is a strongly convex quadratic function by adapting a Haugazeau's algorithm [Hau68] (see also [BC11]), which is another known method for solving the BAP. We note however that the rate of $O(1/k)$ in Haugazeau's algorithm is typical, even when solving a BAP involving only two halfspaces.

1.3. **Contributions of this paper.** The main contribution of this paper is to extend Dykstra's algorithm so that the SHQP strategy can be incorporated into Dykstra's algorithm. (We try to reserve the use of the word "acceleration" to mean an $O(1/k^2)$ algorithm.) See Algorithm 3.1 for our extension of Dykstra's algorithm. Recall that the Boyle-Dykstra theorem proves the convergence of Dykstra's algorithm to the primal solution of the BAP. We prove that the extended Dykstra's algorithm also converges to the primal solution of the BAP (even when there is no dual minimizer).

Next, we show that a commonly occurring regularity assumption guarantees the existence of a dual minimizer. The existence of such a dual minimizer would, by the results in [CP15], imply that Dykstra's algorithm converges at a $O(1/k)$ rate. This analysis also carries over to our extended Dykstra's algorithm.

We point out that it is useful to use warmstart solutions for Dykstra's algorithm and our extension. While it is recognized that Dykstra's algorithm is the alternating minimization algorithm on the dual, it appears that every description and proof of convergence of Dykstra's algorithm in the literature starts with the default zero vector. See further discussions in Subsection 2.1. We answer the natural question of whether Dykstra's algorithm and our extension converge to the optimal primal solution with a warmstart iterate by adapting the proof of the Boyle-Dykstra Theorem [BD85]. See Appendix A.

We show how all these ideas mentioned earlier can be implemented for the simultaneous Dykstra's Algorithm in Section 5. We also explain another way to incorporate the SHQP strategy on the BAP in Subsection 4.3 that works when a minimizer to the dual problem exists and is more natural to augment to the APG. While this strategy is more natural than our extended Dyktra's algorithm, we were not able to prove its global convergence using the framework of the Boyle-Dykstra theorem.

1.4. **Notation.** Our notation is fairly standard. For a closed convex set D, we let $P_D(\cdot)$ denote the projection onto D. The normal cone of D at a point $x \in D$ in the usual sense of convex analysis is denoted by $N_D(x)$. We will let $\tilde{y} = (y_1, \dots, y_m)$. When we discuss the extended Dykstra's algorithm in Section 3, we will need $\tilde{y} = (y_1, \dots, y_m, y_{m+1})$, but this shouldn't cause too much confusion.

2. Preliminaries: Dykstra's algorithm

In this section, we recall Dykstra's algorithm and some results. We also give a discussion of warmstarting Dykstra's algorithm.

Algorithm 2.1. *(Warmstart Dykstra's algorithm) Let X be a Hilbert space. Consider the problem of projecting a point $d \in X$ onto $C \subset X$, where $C = \cap_{i=1}^m C_i$ and C_i are closed convex sets. Choose starting $y_i^{(0)} \in X$ for all $i \in \{1, \dots, m\}$, and let $x_m^{(0)} = d - (y_1^{(0)} + \cdots + y_m^{(0)})$.*

For $k = 1, 2, \dots$
  $x_0^{(k)} = x_m^{(k-1)}$
  For $i = 1, 2, \dots, m$
    $z_i^{(k)} := x_{i-1}^{(k)} + y_i^{(k-1)}$
    $x_i^{(k)} := P_{C_i}(z_i^{(k)})$
    $y_i^{(k)} := z_i^{(k)} - x_i^{(k)}$
  End for
End for

Let the vector $\tilde{y} \in X^m$ be $(y_1, \dots, y_m)$, where each $y_i \in X$. For each closed convex set $D \subset X$, let $\delta^*(\cdot, D) : X \to \mathbb{R}$ be defined by $\delta^*(y, D) = \max_{x \in D} \langle y, x \rangle$. (The function $\delta^*(\cdot, D)$ is also the conjugate of the indicator function $\delta(\cdot, D)$, thus explaining our notation.) Define the dual problem (D') by

$$(D') \quad \inf_{y_1,\dots,y_m} \quad h(y_1, \dots, y_m) := f(y_1 + \cdots + y_m) + \sum_{i=1}^m \delta^*(y_i, C_i), \quad (2.1)$$

where $y_i \in X$ and the $f : X \to \mathbb{R}$ is as in (1.1).

We review some easy results on (D').

Proposition 2.2. *Let X be a Hilbert space. Let C_i be closed convex sets in X for $i \in \{1, \dots, m\}$, and let $C = \cap_{i=1}^m C_i$. Let $d \in X$ and $\bar{x} = P_C(d)$. Let $\tilde{y} = (y_1, \dots, y_m)$. We have the following:*

1. $\inf_{y_1,\dots,y_m} h(y_1, \dots, y_m) = \frac{1}{2}\|d\|^2 - \frac{1}{2}\|d - \bar{x}\|^2$.
2. *Let $v : X^m \to \mathbb{R}$ be defined by*
$$v(y_1, \dots, y_m) = \tfrac{1}{2}\|d - (y_1 + \cdots + y_m) - \bar{x}\|^2 + \sum_{i=1}^m \delta^*(y_i, C_i - \bar{x}). \quad (2.2)$$
Then $v(\tilde{y}) = h(\tilde{y}) - \langle d, \bar{x} \rangle + \frac{1}{2}\|\bar{x}\|^2$, and $\inf_{\tilde{y}} v(\tilde{y}) = 0$.
3. *We have $v(y_1, \dots, y_m) \geq \frac{1}{2}\|d - (y_1 + \cdots + y_m) - \bar{x}\|^2$.*
4. *If $(y_1, \dots, y_m)$ is a minimizer of $v(\cdot)$ (or equivalently, $h(\cdot)$), then $\bar{x} = d - (y_1 + \cdots + y_m)$.*
5. *If $m = 1$, then $y_1 = d - \bar{x}$ is a minimizer of $v(\cdot)$ (or equivalently, $h(\cdot)$).*

Proof. Statement (1) can be obtained from [GM89, pages 32–33]. For Statement (2), note that

$$\begin{aligned} v(\tilde{y}) &= \tfrac{1}{2}\|d - (y_1 + \cdots + y_m) - \bar{x}\|^2 + \sum_{i=1}^m \delta^*(y_i, C_i - \bar{x}) \\ &= \tfrac{1}{2}\|d - (y_1 + \cdots + y_m)\|^2 - \langle d - (y_1 + \cdots + y_m), \bar{x} \rangle + \tfrac{1}{2}\|\bar{x}\|^2 \\ &\quad + \left[\sum_{i=1}^m \delta^*(y_i, C_i)\right] - \langle y_1 + \cdots + y_m, \bar{x} \rangle \\ &= \tfrac{1}{2}\|d - (y_1 + \cdots + y_m)\|^2 - \langle d, \bar{x} \rangle + \tfrac{1}{2}\|\bar{x}\|^2 + \left[\sum_{i=1}^m \delta^*(y_i, C_i)\right] \\ &= h(\tilde{y}) - \langle d, \bar{x} \rangle + \tfrac{1}{2}\|\bar{x}\|^2. \end{aligned}$$

The rest of Statement (2) is elementary. Statements (3) and (4) follow easily from the fact $0 \in C_i - \bar{x}$, which gives $\delta^*(y_i, C_i - \bar{x}) \geq \langle y_i, 0\rangle = 0$. Statement (5) is easy. $\square$

As explained in [Han88, GM89] and perhaps other sources, alternating minimization in the order

$$\begin{aligned} y_1^{(k)} &= \arg\min_y h(y, y_2^{(k-1)}, y_3^{(k-1)}, \dots, y_m^{(k-1)}) \\ y_2^{(k)} &= \arg\min_y h(y_1^{(k)}, y, y_3^{(k-1)}, \dots, y_m^{(k-1)}) \\ &\vdots \\ y_m^{(k)} &= \arg\min_y h(y_1^{(k)}, y_2^{(k)}, \dots, y_{m-1}^{(k)}, y), \end{aligned} \tag{2.3}$$

leads to the Dykstra's algorithm as presented in Algorithm 2.1 through Proposition 2.2(5). We also have the following easily verifiable facts:

$$x_i^{(k)} = d - y_1^{(k)} - \cdots - y_{i-1}^{(k)} - y_i^{(k)} - y_{i+1}^{(k-1)} - \cdots - y_m^{(k-1)} \tag{2.4}$$

$$\text{and } z_i^{(k)} = d - y_1^{(k)} - \cdots - y_{i-1}^{(k)} - y_{i+1}^{(k-1)} - \cdots - y_m^{(k-1)} \tag{2.5}$$

2.1. **Warmstart Dykstra's algorithm.** It appears that all descriptions and proofs of convergence of Dykstra's algorithm use the default starting point $y_i^{(0)} = 0$ for all $i \in \{1, \dots, m\}$. We saw earlier that Dykstra's algorithm is alternating minimization on the dual problem with starting point $\tilde{y}^{(0)}$. In particular, the iterates $\tilde{y}^{(k)}$ are such that $\{h(\tilde{y}^{(k)})\}_k$ is a non-increasing sequence of real numbers to the dual objective value. One may then choose a starting point $\tilde{y}^{(0)}$ such that $h(\tilde{y}^{(0)})$ is closer to the dual objective value than the default starting point of all zeros. There are several ways to obtain a different starting point.

(1) One can use greedy algorithms (that may not guarantee global convergence to the optimal solution) to decrease the dual objective values. A plausible strategy is to use the greedy algorithms till they do not appear to achieve good decrease in the value $h(\cdot)$, then switch to the warmstart Dykstra's algorithm, or our extended algorithm in Algorithm 3.1, to guarantee convergence to the optimal primal solution.
(2) A warmstart solution may be available after solving a nearby problem. For example, one might want to resolve a problem after a set has been added or removed, or after a perturbation of parameters. Alternatively, there may be a nearby structured problem that can be solved approximately with less effort than the original problem.

The proof of convergence of Dykstra's algorithm with a different starting point is not too different from the Boyle-Dykstra theorem. We defer the proof to Appendix A, where we also prove the convergence of our extended Dykstra's algorithm to be introduced in Section 3.

3. Extended Dykstra's algorithm

As mentioned in Subsection 1.2, the SHQP strategy (of collecting halfspaces containing C generated by earlier projections and then projecting onto the intersection of the halfspaces by QP) can enhance the convergence of the method of alternating projections for the set intersection problem. In this section, we present our extension of Dykstra's algorithm in Algorithm 3.1 and how it can incorporate the SHQP strategy. In order to extend the proof of the Boyle-Dykstra theorem to establish the primal convergence of our extended Dykstra's algorithm, we need Theorem 3.4(2). The proof of Theorem 3.4(2) illustrates why lines 8 and 12 of Algorithm 3.1 were designed as such. The other parts of the Boyle-Dykstra theorem follow with little modifications, so we defer the rest of the convergence proof to Appendix A. We now present our extended Dykstra's algorithm.

Algorithm 3.1. *(Extended Dykstra's algorithm) Consider the BAP* (1.1). *Let* $y_i^{(0)} \in X$ *be the starting dual variables for each component* $i \in \{1, \dots, m\}$. *We also introduce a variable* $y_{m+1}^{(k)} \in X$, *with starting value* $y_{m+1}^{(0)}$ *being* 0, *in our calculations. Let* $H_{m+1}^0 = X$. *Set* $x_{m+1}^{(0)} = d - \sum_{i=1}^{m+1} y_i^{(0)}$.

01 *For* $k = 1, 2, \dots$
02 $\quad x_0^{(k)} = x_{m+1}^{(k-1)}$
03 $\quad$ *For* $i = 1, 2, \dots, m$
04 $\quad\quad z_i^{(k)} := x_{i-1}^{(k)} + y_i^{(k-1)}$
05 $\quad\quad x_i^{(k)} := P_{C_i}(z_i^{(k)})$
06 $\quad\quad y_i^{(k)} := z_i^{(k)} - x_i^{(k)}$
07 $\quad$ *End for*
08 $\quad$ *Let* $C_{m+1}^k \subset X$ *be such that* $C \subset C_{m+1}^k \subset H_{m+1}^{k-1}$.
09 $\quad z_{m+1}^{(k)} := x_m^{(k)} + y_{m+1}^{(k-1)}$
10 $\quad x_{m+1}^{(k)} = P_{C_{m+1}^k}(z_{m+1}^{(k)})$
11 $\quad y_{m+1}^{(k)} = z_{m+1}^{(k)} - x_{m+1}^{(k)}$
12 $\quad$ *Let* H_{m+1}^k *be the halfspace with normal* $y_{m+1}^{(k)}$ *passing through* $x_{m+1}^{(k)}$, *i.e.,*

$$H_{m+1}^k = \{x : \langle y_{m+1}^{(k)}, x - x_{m+1}^{(k)} \rangle \leq 0\}.$$

13 *End for*

Remark 3.2. (Designing C_{m+1}^k) In line 8 of Algorithm 3.1, the set C_{m+1}^k can be chosen to be the intersection of H_{m+1}^{k-1} and the halfspaces generated through earlier projections. The projection $P_{C_{m+1}^k}(\cdot)$ can then be calculated easily using methods of quadratic programming if the number of halfspaces defining C_{m+1}^k is small. It is clear to see that Algorithm 3.1 reduces to the original Dykstra's algorithm if we had kept $H_{m+1}^k = C_{m+1}^k = X$ for all $k \in \{1, 2, \dots\}$. The choice of storing halfspaces for H_{m+1}^k in line 12 simplifies computations involved.

Remark 3.3. (Positioning sets of type C_{m+1}^k) If the number m is large, then one can introduce more than just one additional set of the type C_{m+1}^k at the end of all the original sets in an implementation of Algorithm 3.1. For example, one can introduce the additional set after every fixed number of original sets so that the quadratic programs formed will have a manageable number of halfspaces.

Theorem 3.4 below will be crucial in proving that the iterates $\{x_i^{(k)}\}$ of Algorithm 3.1 converges to the optimal primal solution. The proof of the Theorem 3.4 explains how the sets C_{m+1}^k and H_{m+1}^k were designed in order to maintain the conclusion in Theorem 3.4(2).

Theorem 3.4. *(Properties of Algorithm 3.1) In Algorithm 3.1, define the dual function $h^k : X^{m+1} \to \mathbb{R}$ at the kth iteration and $\bar{h} : X^{m+1} \to \mathbb{R}$ by*

$$\begin{aligned} h^k(\tilde{y}) &= \tfrac{1}{2}\|d-(y_1+\cdots+y_{m+1})\|^2 + \left[\sum_{i=1}^m \delta^*(y_i, C_i)\right] + \delta^*(y_{m+1}, H^k_{m+1}) \\ \bar{h}(\tilde{y}) &= \tfrac{1}{2}\|d-(y_1+\cdots+y_{m+1})\|^2 + \left[\sum_{i=1}^m \delta^*(y_i, C_i)\right] + \delta^*(y_{m+1}, C). \end{aligned} \tag{3.1}$$

Let $\tilde{y}^{(k)} = (y_1^{(k)}, \ldots, y_m^{(k)}, y_{m+1}^{(k)})$. The following hold:

1. $h^{k-1}(\tilde{y}^{(k-1)}) \geq h^k(\tilde{y}^{(k)}) + \frac{1}{2}\sum_{i=1}^{m+1} \|y_i^{(k)} - y_i^{(k-1)}\|^2$.
2. *The sum* $\sum_{j=1}^\infty \sum_{i=1}^{m+1} \|y_i^{(j)} - y_i^{(j-1)}\|^2$ *is finite.*

Proof. We have the following chain of inequalities:

$$\begin{aligned} & \tfrac{1}{2}\|d-(y_1^{(k)}+\cdots+y_m^{(k)})-y_{m+1}^{(k-1)}\|^2 + \delta^*(y_{m+1}^{(k-1)}, H^{k-1}_{m+1}) \\ \geq\; & \tfrac{1}{2}\|d-(y_1^{(k)}+\cdots+y_m^{(k)})-y_{m+1}^{(k-1)}\|^2 + \delta^*(y_{m+1}^{(k-1)}, C^{k}_{m+1}) \\ \geq\; & \tfrac{1}{2}\|d-(y_1^{(k)}+\cdots+y_m^{(k)})-y_{m+1}^{(k)}\|^2 + \delta^*(y_{m+1}^{(k)}, C^{k}_{m+1}) + \tfrac{1}{2}\|y_{m+1}^{(k)} - y_{m+1}^{(k-1)}\|^2 \\ =\; & \tfrac{1}{2}\|d-(y_1^{(k)}+\cdots+y_m^{(k)})-y_{m+1}^{(k)}\|^2 + \delta^*(y_{m+1}^{(k)}, H^{k}_{m+1}) + \tfrac{1}{2}\|y_{m+1}^{(k)} - y_{m+1}^{(k-1)}\|^2. \end{aligned} \tag{3.2}$$

The first inequality comes from the fact that $C^k_{m+1} \subset H^{k-1}_{m+1}$, which implies that $\delta^*(\cdot, H^{k-1}_{m+1}) \geq \delta^*(\cdot, C^k_{m+1})$. The second inequality comes from the fact that $y^{(k)}_{m+1}$ is the minimizer of the strongly convex function with modulus 1 defined by

$$y \mapsto \tfrac{1}{2}\|d-(y_1^{(k)}+\cdots+y_m^{(k)})-y\|^2 + \delta^*(y, C^k_{m+1}).$$

The final equation follows readily from the definition of H^k_{m+1}. We can apply the same principle in (3.2) to show that for all $i \in \{1, \ldots, m\}$, we have

$$\begin{aligned} & \tfrac{1}{2}\|d - y_1^{(k)} - \cdots y_{i-1}^{(k)} - y_i^{(k-1)} - y_{i+1}^{(k-1)} - \cdots - y_{m+1}^{(k-1)}\|^2 + \delta^*(y_i^{(k-1)}, C_i) \\ \geq\; & \tfrac{1}{2}\|d - y_1^{(k)} - \cdots y_{i-1}^{(k)} - y_i^{(k)} - y_{i+1}^{(k-1)} - \cdots - y_{m+1}^{(k-1)}\|^2 \\ & + \delta^*(y_i^{(k)}, C_i) + \tfrac{1}{2}\|y_i^{(k)} - y_i^{(k-1)}\|^2. \end{aligned} \tag{3.3}$$

Combining (3.2) and (3.3) gives (1).

From the fact that $H^k_{m+1} \supset C$, we have $\delta^*(\cdot, H^k_{m+1}) \geq \delta^*(\cdot, C)$, which in turn implies that $h^k(y) \geq \bar{h}(y)$. Moreover, for each k, we make use of the observation in Proposition 2.2(1) to get

$$\inf_{y \in X^{m+1}} h^k(y) = \min_{y \in X^{m+1}} \bar{h}(y).$$

Hence

$$\begin{aligned} \tfrac{1}{2}\sum_{j=1}^k \sum_{i=1}^{m+1} \|y_i^{(j)} - y_i^{(j-1)}\|^2 &\leq h^0(\tilde{y}^{(0)}) - h^k(\tilde{y}^{(k)}) \\ &\leq h^0(\tilde{y}^{(0)}) - \bar{h}(\tilde{y}^{(k)}) \\ &\leq h^0(\tilde{y}^{(0)}) - \min_y \bar{h}(y). \end{aligned}$$

Thus (2) follows. ☐

The rest of the proof of the primal convergence of Algorithm 3.1 is not too different from the Boyle-Dykstra theorem, so we will prove the convergence result in Appendix A.

4. Convergence rate of alternating minimization and Dykstra's algorithm

In this section, we first recall the proof of the $O(1/k)$ convergence rate of alternating minimization under the assumption of strong convexity of subproblems and bounded level sets. This will then give us the convergence rate of the function $h(\cdot)$ in the dual of Dykstra's algorithm. We also discuss how this analysis can be carried over to our extended Dykstra's algorithm. In Subsection 4.3, we introduce another more natural way to incorporate the SHQP heuristic into Dykstra's algorithm and attains the nonasymptotic $O(1/k)$ convergence rate when there is a dual minimizer. But we note that we are unable to prove the global convergence to the primal optimal solution for this new strategy.

4.1. **General convergence rate result on alternating minimization.** In this subsection, we recall that under certain conditions, alternating minimization has a nonasymptotic convergence rate of $O(1/k)$. We need the following result proved in [BT13] and [Bec15].

Lemma 4.1. *(Sequence convergence rate) Let $\alpha > 0$. Suppose the sequence of nonnegative numbers $\{a_k\}_{k=0}^{\infty}$ is such that*

$$a_k \geq a_{k+1} + \alpha a_{k+1}^2 \text{ for all } k \in \{1, 2, \dots\}.$$

(1) [BT13, Lemma 6.2] If furthermore, $a_1 \leq \frac{1.5}{\alpha}$ and $a_2 \leq \frac{1.5}{2\alpha}$, then

$$a_k \leq \tfrac{1.5}{\alpha k} \text{ for all } k \in \{1, 2, \dots\}.$$

(2) [Bec15, Lemma 3.8] For any $k \geq 2$,

$$a_k \leq \max\left\{ \left(\tfrac{1}{2}\right)^{(k-1)/2} a_0, \tfrac{4}{\alpha(k-1)} \right\}.$$

In addition, for any $\epsilon > 0$, if

$$k \geq \max\left\{ \tfrac{2}{\ln(2)}[\ln(a_0) + \ln(1/\epsilon)], \tfrac{4}{\alpha\epsilon} \right\} + 1,$$

then $a_n \leq \epsilon$.

The second formula refines the first by reducing the dependence of a_k on the first few terms of $\{a_i\}_i$.

We now prove our general convergence rate result for alternating minimization. The following result was discussed in [CP15] and its ideas appeared in [BT13, Bec15].

Theorem 4.2. *($O(1/k)$ Convergence rate of alternating minimization) Let $f : X^m \to \mathbb{R}$ be a smooth convex function, and $g_i : X \to \mathbb{R}$ be (not necessarily smooth) convex functions for $i \in \{1, \dots, m\}$, Define $h : X^m \to \mathbb{R}$ by*

$$h(y_1, y_2, \dots, y_m) = f(y_1, y_2, \dots, y_m) + \sum_{i=1}^{m} g_i(y_i).$$

such that

(1) *The gradient $f' : X^m \to X^m$ is Lipschitz continuous with modulus L, and*
(2) *There is a number $\mu > 0$ such that for all $i \in \{1, \dots, m\}$ and fixed variables $y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_m$, the map*

$$y \mapsto f(y_1, y_2, \dots, y_{i-1}, y, y_{i+1}, \dots, y_m)$$

is strongly convex with modulus $\mu > 0$.

(3) *A minimizer* $\tilde{y}^* = (y_1^*, y_2^*, \dots, y_m^*)$ *of* $h(\cdot)$ *exists. Moreover,* M_i *defined by* $M_i = \sup\{\|y_i^{(k)} - y_i^*\| : k \geq 0\}$ *is finite for all* $i \in \{1, \dots, m-1\}$.

Suppose two successive iterates $\tilde{y}^{(k-1)} = (y_1^{(k-1)}, y_2^{(k-1)}, \dots, y_m^{(k-1)})$ *and* $\tilde{y}^{(k)}$ *defined similarly are produced by alternating minimization described in* (2.3). *Let* $M = \max_{i\in\{1,\dots,m-1\}} M_i$. *Then*

$$h(\tilde{y}^{(k-1)}) - h(\tilde{y}^*) \geq h(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \tfrac{\mu}{2(m-1)^3M^2L^2}[h(\tilde{y}^{(k)}) - h(\tilde{y}^*)]^2. \tag{4.1}$$

Applying Lemma 4.1 to $a_k := h(\tilde{y}^{(k)}) - h(\tilde{y}^*)$ *gives*

$$h(\tilde{y}^{(k)}) - h(\tilde{y}^*) \leq \tfrac{1}{k}\max\{\tfrac{3(m-1)^3M^2L^2}{\mu}, h(\tilde{y}^{(1)}) - h(\tilde{y}^*), 2[h(\tilde{y}^{(2)}) - h(\tilde{y}^*)]\},$$

and

$$h(\tilde{y}^{(k)}) - h(\tilde{y}^*) \leq \max\left\{\left(\tfrac{1}{2}\right)^{(k-1)/2}[h(\tilde{y}^{(0)}) - h(\tilde{y}^*)], \tfrac{8(m-1)^3M^2L^2}{\mu(k-1)}\right\}.$$

Proof. The proof of this result follows similar ideas as those in [CP15], which in turn appeared in [BT13, Bec15]. Since we will use elements of this proof for the proof of Theorem 4.5, we now give a self contained proof. For each i, let $h_i : X \to \mathbb{R}$ be defined by

$$h_i(y) = h(y_1^{(k)}, y_2^{(k)}, \dots, y_{i-1}^{(k)}, y, y_{i+1}^{(k-1)}, \dots, y_m^{(k-1)}).$$

In other words, $h_i(\cdot)$ is the ith block of $h : X^m \to \mathbb{R}$. The mapping $h_i(\cdot)$ has minimizer $y_i^{(k)}$, and is strongly convex with modulus μ from assumption (2). Hence

$$h_i(y_i^{(k-1)}) \geq h_i(y_i^{(k)}) + \tfrac{\mu}{2}\|y_i^{(k)} - y_i^{(k-1)}\|^2.$$

Hence

$$h(\tilde{y}^{(k-1)}) - h(\tilde{y}^*) \geq h(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \sum_{i=1}^m \tfrac{\mu}{2}\|y_i^{(k)} - y_i^{(k-1)}\|^2. \tag{4.2}$$

Next, we try to find a subgradient in $\partial h(\tilde{y}^{(k)})$ by looking at the components $\partial h_i(\tilde{y})$. It is clear that $0 \in \partial h_m(y_m^{(k)})$. We then look at the ith component of $f'(\cdot)$, which we denote by $f_i'(\cdot)$. For each $i \in \{1, \dots, m\}$, the optimality conditions of each iteration of alternating minimization implies that

$$0 \in f_i'(y_1^{(k)}, y_2^{(k)}, \dots, y_{i-1}^{(k)}, y_i^{(k)}, y_{i+1}^{(k-1)}, \dots, y_m^{(k-1)}) + \partial g_i(y_i^{(k)}).$$

Thus

$$f_i'(\tilde{y}^{(k)}) - f_i'(y_1^{(k)}, y_2^{(k)}, \dots, y_{i-1}^{(k)}, y_i^{(k)}, y_{i+1}^{(k-1)}, \dots, y_m^{(k-1)}) \in f_i'(\tilde{y}^{(k)}) + \partial g_i(y_i^{(k)}).$$

Choose a subgradient $s \in \partial h(\tilde{y}^{(k)})$, with $s \in X^m$ such that

$$s_i = f_i'(\tilde{y}^{(k)}) - f_i'(y_1^{(k)}, y_2^{(k)}, \dots, y_{i-1}^{(k)}, y_i^{(k)}, y_{i+1}^{(k-1)}, \dots, y_m^{(k-1)}).$$

We have

$$\begin{aligned}\|s_i\| &\leq \|f_i'(\tilde{y}^{(k)}) - f_i'(y_1^{(k)}, y_2^{(k)}, \dots, y_{i-1}^{(k)}, y_i^{(k)}, y_{i+1}^{(k-1)}, \dots, y_m^{(k-1)})\| \\ &\leq L\sum_{j=i+1}^m \|y_j^{(k)} - y_j^{(k-1)}\| \\ &\leq L\sum_{j=2}^m \|y_j^{(k)} - y_j^{(k-1)}\|.\end{aligned} \tag{4.3}$$

The above derivation also reminds us that $\|s_m\| = 0$. Thus, making use of condition (3), we have

$$\begin{array}{rcl} h(\tilde{y}^*) & \geq & h(\tilde{y}^{(k)}) + \langle s, \tilde{y}^* - \tilde{y}^{(k)}\rangle \\ \Rightarrow h(\tilde{y}^{(k)}) - h(\tilde{y}^*) & \leq & -\langle s, \tilde{y}^* - \tilde{y}^{(k)}\rangle \\ & \leq & \sum_{i=1}^{m-1} \|s_i\| \|y_i^* - y_i^{(k)}\| \\ & \leq & L\left[\sum_{j=2}^{m} \|y_j^{(k)} - y_j^{(k-1)}\|\right]\left[\sum_{i=1}^{m-1} \|y_i^* - y_i^{(k)}\|\right] \\ & \leq & (m-1)ML\left[\sum_{j=2}^{m} \|y_j^{(k)} - y_j^{(k-1)}\|\right]. \end{array} \tag{4.4}$$

Applying (4.4) on (4.2) gives

$$\begin{array}{rcl} h(\tilde{y}^{(k-1)}) - h(\tilde{y}^*) & \geq & h(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \sum_{i=1}^{m} \frac{\mu}{2} \|y_i^{(k)} - y_i^{(k-1)}\|^2 \\ & \geq & h(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \sum_{i=2}^{m} \frac{\mu}{2} \|y_i^{(k)} - y_i^{(k-1)}\|^2 \\ & \geq & h(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \frac{\mu}{2(m-1)}\left[\sum_{i=2}^{m} \|y_i^{(k)} - y_i^{(k-1)}\|\right]^2 \\ & \geq & h(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \frac{\mu}{2(m-1)^3 M^2 L^2}[h(\tilde{y}^{(k)}) - h(\tilde{y}^*)]^2. \end{array} \tag{4.5}$$

Let $a_k = h(\tilde{y}^{(k)}) - h(\tilde{y}^*)$. Applying Lemma 4.1 gives us our conclusion. (For the first formula, $\alpha = \min\{\frac{\mu}{2(m-1)^3 M^2 L^2}, \frac{1.5}{a_1}, \frac{0.75}{a_2}\}$.) □

It is clear to see that condition (3) in Theorem 4.2 is satisfied when the level sets of $h(\cdot)$ are bounded. Condition (3) can be easily amended to having all but one of the M_i for $i \in \{1, \ldots, m\}$ being finite.

4.2. **Convergence rate of extended Dykstra's algorithm.** In Dykstra's algorithm, the function $f(\cdot)$ in (1.1) is quadratic, and therefore its gradient is linear. Furthermore, each block $f_i(\cdot)$ is strongly convex with modulus 1. Thus conditions (1) and (2) of Theorem 4.2 are satisfied. We make some remarks condition (3) of Theorem 4.2.

Remark 4.3. (Condition (3) of Theorem 4.2 for Dykstra's algorithm) As pointed out in [Han88], there may not exist a minimizer $\tilde{y}^*$ of the dual problem (D'). Consider for example the problem of projecting onto the intersection of two circles in $\mathbb{R}^2$ intersecting at only one point. Furthermore, Gaffke and Mathar [GM89, Lemma 2] showed that for Dykstra's algorithm, if there is a $\lambda > 2$ such that $\|x_m^{(k)} - \bar{x}\|^2 \in O(1/k^\lambda)$, then $y_i^* = \lim_{k\to\infty} y_i^k$ exists with $\delta^*(y_i^*, C_i)$ finite, and $\tilde{y}^* = (y_1^*, \ldots, y_m^*)$ minimizing the function $h(\cdot)$ of (2.1). This result can somewhat be seen as a converse of Theorem 4.2.

Remark 4.4. (Finiteness of the M_i's) In our analysis of Dykstra's algorithm, suppose all but one of the M_i's in Theorem 4.2(3) are finite for $i \in \{1, \ldots, m\}$. The Boyle-Dykstra theorem implies that the limit

$$\lim_{k\to\infty} [d - y_1^{(k)} - \cdots - y_m^{(k)}] = \lim_{k\to\infty} x_m^{(k)}$$

exists. This would imply that all the M_i's are finite.

We now provide the additional details to show that Algorithm 3.1 (the extended Dykstra's algorithm) also converges at an $O(1/k)$ rate.

Theorem 4.5. *(Convergence rate of extended Dykstra's algorithm) Consider Algorithm 3.1. Recall the definition of* $h(\cdot)$ *in* (2.1)*. Suppose the following holds:*

(3′) *A minimizer* $\tilde{y}^* = (y_1^*, y_2^*, \dots, y_m^*)$ *of* $h(\cdot)$ *exists. Moreover,* M_i *defined by* $M_i = \sup\{\|y_i^{(k)} - y_i^*\| : k \geq 0\}$ *is finite for all* $i \in \{1, \dots, m+1\}$.

(Compare this to condition (3) of Theorem 4.2.) Recall the definition of $h^k(\cdot)$ *in* (3.1)*. Then the sequence* $\{h^k(y_1^{(k)}, \dots, y_{m+1}^{(k)})\}_k$*, converges to* $h(\tilde{y}^*)$ *at a rate of* $O(1/k)$.

Proof. We highlight the differences this proof has with that of Theorem 4.2. Theorem 3.4(1) shows that

$$h^{k-1}(\tilde{y}^{(k-1)}) \geq h^k(\tilde{y}^{(k)}) + \tfrac{1}{2}\sum_{i=1}^{m+1} \|y_i^{(k)} - y_i^{(k-1)}\|^2,$$

which plays the role of (4.2). Next, if $\tilde{y}^* = (y_1^*, \dots, y_m^*)$ is a minimizer of $h(\cdot)$, then $(y_1^*, \dots, y_m^*, 0)$ is a minimizer of $h^k(\cdot)$ for all k. Moreover,

$$h^k(y_1^*, \dots, y_m^*, 0) = h(\tilde{y}^*).$$

Next, we can prove an analogous result to (4.3) with $L = 1$. The analogous result to (4.4) is

$$h^k(\tilde{y}^{(k)}) - h(\tilde{y}^*) \leq mML\left[\sum_{j=2}^{m+1} \|y_j^{(k)} - y_j^{(k-1)}\|\right]. \tag{4.6}$$

The analogous result to (4.5) is

$$h^{k-1}(\tilde{y}^{(k-1)}) - h(\tilde{y}^*) \geq h^k(\tilde{y}^{(k)}) - h(\tilde{y}^*) + \tfrac{\mu}{2m^3M^2L^2}[h^k(\tilde{y}^{(k)}) - h(\tilde{y}^*)]^2. \tag{4.7}$$

The conclusion follows with steps similar to the proof of Theorem 4.2. □

An indicator of whether an $O(1/k)$ convergence rate is achieved would be whether condition (3) in Theorem 4.2 is satisfied. The next result gives sufficient conditions.

Theorem 4.6. *(Condition for bounded dual iterates) Suppose* $X = \mathbb{R}^n$*, and consider the BAP* (1.1)*.*

(1) *Suppose at the primal optimal solution* $x^* = P_C(d)$*, we have*

$$\begin{aligned}&\sum_{i=1}^m v_i = 0 \text{ and } v_i \in N_{C_i}(x^*) \text{ for all } i \in \{1, \dots, m\}\\ &\quad\text{implies } v_i = 0 \text{ for all } i \in \{1, \dots, m\}.\end{aligned} \tag{4.8}$$

Then the iterates $\{\tilde{y}^{(k)}\}$ *of Dykstra's algorithm are bounded. Moreover, an accumulation point exists, and is an optimal solution for* (D')*, so condition (3) of Theorem 4.2 holds.*

(2) *Suppose at the primal optimal solution* $x^* = P_C(d)$*, we have*

$$\begin{aligned}&\sum_{i=1}^{m+1} v_i = 0, v_{m+1} \in N_C(x^*) \text{ and } v_i \in N_{C_i}(x^*) \text{ for all } i \in \{1, \dots, m\}\\ &\quad\text{implies } v_i = 0 \text{ for all } i \in \{1, \dots, m+1\}.\end{aligned} \tag{4.9}$$

Then the iterates $\{\tilde{y}^{(k)}\}$ *of the extended Dykstra's algorithm are bounded. Moreover, an accumulation point exists, and is a minimizer of* $\bar{h} : (\mathbb{R}^n)^{m+1} \to \mathbb{R}$ *defined in* (3.1)*, so condition (3′) of Theorem 4.5 holds.*

(3) *Suppose* $N_{C_i}(x^*)$ *does not contain a line for all* $i \in \{1, \dots, m\}$*. In other words, the cones* $N_{C_i}(x^*)$ *are pointed for all* i*. Then* (4.8) *and* (4.9) *are equivalent.*

Proof. For (1), we prove the boundedness of the iterates for Dykstra's algorithm. The other parts of the result are straightforward. Seeking a contradiction, suppose the iterates $\{\tilde{y}^{(k)}\}$ are not bounded. Then

$$\begin{aligned} \sum_{i=1}^{m} y_i^{(k)} &= d - x_m^{(k)} \\ \frac{1}{\max_i \|y_i^{(k)}\|} \sum_{i=1}^{m} y_i^{(k)} &= \frac{1}{\max_i \|y_i^{(k)}\|}[d - x_m^{(k)}]. \end{aligned} \tag{4.10}$$

By the convergence of Dykstra's algorithm, $\lim_{k\to\infty}[d - x_m^{(k)}]$ exists. Moreover, $\limsup_{k\to\infty} \max_i \|y_i^{(k)}\| = \infty$, so by taking a subsequence if necessary (we do not relabel), the limit of the RHS of (4.10) is zero. Let $\hat{y}_i^{(k)} = \frac{y_i^{(k)}}{\max_j \|y_j^{(k)}\|}$. We thus have

$$\sum_{i=1}^{m} \hat{y}_i^{(k)} = 0.$$

The sequence $\{(\hat{y}_1^{(k)}, \dots, \hat{y}_m^{(k)})\}_k$ has a convergent subsequence. Let an accumulation point be $(\hat{y}_1^*, \dots, \hat{y}_m^*)$. Note that $\hat{y}_i^{(k)} \in N_{C_i}(x_i^{(k)})$, so $\hat{y}_i^* \in N_{C_i}(x^*)$. But not all the $\hat{y}_i^*$ are zero. This gives us the contradiction to (4.8).

We now show how to amend the proof of (1) to prove (2). For the extended Dykstra's algorithm, we can obtain the formula

$$\frac{1}{\max_i \|y_i^{(k)}\|} \sum_{i=1}^{m+1} y_i^{(k)} = \frac{1}{\max_i \|y_i^{(k)}\|}[d - x_{m+1}^{(k)}],$$

which is similar to (4.10). The sequence $\{(\hat{y}_1^{(k)}, \dots, \hat{y}_{m+1}^{(k)})\}_k$ is defined similarly by $\hat{y}_i^{(k)} = \frac{y_i^{(k)}}{\max_j \|y_j^{(k)}\|}$, and has a convergent subsequence with accumulation point $(\hat{y}_1^*, \dots, \hat{y}_m^*)$. For any $c \in C$, we have

$$\langle \hat{y}_{m+1}^{(k)}, c - x_{m+1}^{(k)} \rangle \leq 0.$$

As we take limits, we have

$$\langle \hat{y}_{m+1}^*, c - x^* \rangle \leq 0,$$

so $\hat{y}_{m+1}^* \in N_C(x^*)$. The same steps would imply that (4.9) is violated, hence a contradiction.

Lastly, we prove (3). It is obvious that (4.9) implies (4.8) (just take the particular case when $v_{m+1} = 0$). We now prove that (4.8) implies (4.9). If (4.8) holds, then the formula for intersection of normal cones of convex sets (see [RW98, Theorem 6.42]) implies that

$$N_C(x^*) = \sum_{i=1}^{m} N_{C_i}(x^*).$$

Suppose $\sum_{i=1}^{m+1} v_i = 0$, where $v_{m+1} \in N_C(x^*)$ and $v_i \in N_{C_i}(x^*)$ for all $i \in \{1, \dots, m\}$. We can write $v_{m+1} = \sum_{i=1}^{m+1} \tilde{v}_i$, where $\tilde{v}_i \in N_{C_i}(x^*)$ for all $i \in \{1, \dots, m\}$. Then $\sum_{i=1}^{m}(v_i + \tilde{v}_i) = 0$, and $(v_i + \tilde{v}_i) \in N_{C_i}(x^*)$. Condition (4.8) would imply that $v_i + \tilde{v}_i = 0$ for all $i \in \{1, \dots, m\}$. Since $N_{C_i}(x^*)$ contains no lines for all $i \in \{1, \dots, m\}$, we have $v_i = \tilde{v}_i = 0$ for all $i \in \{1, \dots, m\}$. This implies that (4.9) holds. □

Remark 4.7. We make a few remarks on Theorems 4.6 and 4.2.

(1) A simple example of a line and a halfspace shows that (4.8) and (4.9) cannot be equivalent if the conditions in (3) were omitted. Even so, we can check that in this simple example, the extended Dykstra's algorithm should perform better than the Dykstra's algorithm in general, even when (4.9) fails. See Figure 4.1.
(2) Even if condition (1) in Theorem 4.6 is not satisfied, condition (3) of Theorem 4.2 can hold. For example, consider the case of two (one dimensional) lines intersecting only at the origin in $\mathbb{R}^3$.
(3) The condition (4.8) is well known to be equivalent to the stability of the sets $\{C_i\}_{i=1}^m$ under perturbations. See [Kru06] for example. Condition (4.8) is also important for establishing linear convergence of the method of alternating projections for convex sets. See [BB96].

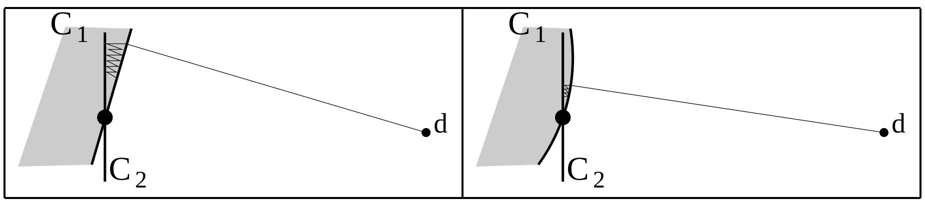

FIGURE 4.1. In the diagram on the left, the line shows the path Dykstra's algorithm takes. But for both the extended Dykstra's algorithm in Algorithm 3.1 (even if (4.9) is not satisfied) and Algorithm 4.8, we have convergence to $P_C(d)$ in a small number of steps. The diagram on the right shows that Algorithms 3.1 and 4.8 are also advantageous for nonpolyhedral problems.

4.3. **SHQP strategy for Dykstra's algorithm.** We now show that in the case where a minimizer exists for $h(\cdot)$ as defined in (2.1), the SHQP strategy can be incorporated into Dykstra's algorithm. We present the following additional step.

Algorithm 4.8. *(SHQP strategy for Dykstra's algorithm) Consider the original warmstart Dykstra's algorithm (Algorithm 2.1). Between lines 7 and 8, we can add as many copies of the following code segment as needed.*

01 Choose $J \subset \{1, \dots, m\}$

02 Update $y_1^{(k)}, \dots, y_m^{(k)}$ by solving the following optimization problem

$$\begin{aligned} (y_1^{(k)}, \dots, y_m^{(k)}) \leftarrow \quad & \underset{y_1,\dots,y_m}{\arg\min} \quad f(y_1 + \cdots + y_m) + \sum_{i=1}^m \delta^*(y_i, H_i) \\ & \quad s.t. \qquad y_i = y_i^{(k)} \text{ if } i \notin J. \end{aligned} \tag{4.11}$$

To illustrate the effectiveness of the step in Algorithm 4.8, let us for now assume that $J = \{1, \dots, m\}$. Let $y_1^{(k),\circ}, \dots, y_m^{(k),\circ} \in X$ be the values of $y_i^{(k)}$ before line 2 was performed in Algorithm 4.8, and let $y_1^{(k),+}, \dots, y_m^{(k),+}$ be the respective values after line 2 was performed. Note that in Dykstra's algorithm, line 5 ($x_i^{(k)} = P_{C_i}(z_i^{(k)})$) is obtained by projecting onto the set C_i, and this projection produces a supporting halfspace H_i at $x_i^{(k)}$ so that $H_i \supset C_i$. Moreover, the halfspace H_i also satisfies

$$\delta^*(y_i^{(k),\circ}, C_i) = \delta^*(y_i^{(k),\circ}, H_i). \tag{4.12}$$

The intersection $\cap_{i=1}^m H_i$ would be a polyhedral outer approximate of $C = \cap_{i=1}^m C_i$. Since $H_i \supset C_i$, we have $\delta^*(\cdot, H_i) \geq \delta^*(\cdot, C_i)$. We therefore have

$$\begin{aligned}
& f(y_1^{(k),\circ} + \cdots + y_m^{(k),\circ}) + \sum_{i=1}^m \delta^*(y_i^{(k),\circ}, C_i) \\
\overset{(4.12)}{=} \quad & f(y_1^{(k),\circ} + \cdots + y_m^{(k),\circ}) + \sum_{i=1}^m \delta^*(y_i^{(k),\circ}, H_i) \\
\overset{(4.11)}{\geq} \quad & f(y_1^{(k),+} + \cdots + y_m^{(k),+}) + \sum_{i=1}^m \delta^*(y_i^{(k),+}, H_i) \\
\geq \quad & f(y_1^{(k),+} + \cdots + y_m^{(k),+}) + \sum_{i=1}^m \delta^*(y_i^{(k),+}, C_i).
\end{aligned}$$

Thus performing the step in line 2 of Algorithm 4.8 improves the dual objective $h(\cdot)$. Note that the minimization problem (4.11) is the dual of the problem of projecting a point onto the polyhedron $\cap_{i \in J} H_i$, which can be solved effectively by quadratic programming if the number of halfspaces is small. If the number of halfspaces is large, then line 1 of Algorithm 4.8 gives the flexibility of solving a quadratic program of manageable size instead. In general, H_i can be chosen to be the intersection of halfspaces such that (4.12) is valid.

If the boundary of C_i is smooth, then H_i approximates C_i at $x_i^{(k)}$, and the algorithm reduces to sequential quadratic programming. This gives a reason why the additional step in Algorithm 4.8 can be effective in practice.

The step explained here gives a similar kind of enhancement to what we saw earlier for the extended Dykstra's algorithm. It is clear to see that the recurrence (4.1) is not affected by the additional step in Algorithm 4.8. Thus the convergence analysis given in Subsection 4.1 remains valid. But when $h(\cdot)$ does not have a minimizer, we were not able to extend the Boyle-Dykstra Theorem (specifically, Lemma A.4 below) for the proof of global convergence of the extension of Dykstra's algorithm using Algorithm 4.8.

5. Simultaneous Dykstra's algorithm

Recall that Dykstra's algorithm reduces the best approximation problem to a series of projections. A variant of Dykstra's algorithm which is more suitable for parallel computations is the simultaneous Dykstra's algorithm proposed and studied in [IP91]. In this section, we give some details on deriving the simultaneous Dykstra's algorithm, and then show how the principles described in extending Dykstra's algorithm can be applied for the simultaneous Dykstra's algorithm.

Consider the BAP (1.1), where we want to find the projection of d onto $C = \cap_{i=1}^m C_i$. We now recall the product space formulation of [Pie84]. Define $\mathcal{C} \subset X^m$ and $\mathcal{D} \subset X^m$ by

$$\begin{aligned}
\mathcal{C} \quad &:= \quad C_1 \times \cdots \times C_m \\
\text{and } \mathcal{D} \quad &:= \quad \{(x, \ldots, x) \in X^m : x \in X\}.
\end{aligned} \tag{5.1}$$

Let $\lambda_1, \ldots, \lambda_m$ be m positive numbers that sum to one, and let the inner product $\langle \cdot, \cdot \rangle_{\bar{Q}}$ in X^m be defined by

$$\langle (u_1, \ldots, u_m), (v_1, \ldots v_m) \rangle_{\bar{Q}} := \sum_{i=1}^m \lambda_i \langle u_i, v_i \rangle.$$

The projection of the point $(d,\ldots,d) \in X^m$ onto $\mathcal{C} \cap \mathcal{D}$ can easily be seen to be $(P_C(d),\ldots,P_C(d))$. Dykstra's algorithm can be applied onto the product space formulation. This gives the simultaneous Dykstra's algorithm proposed and studied in [IP91], which we present below.

Algorithm 5.1. [IP91]*(Simultaneous Dykstra's algorithm) Consider the BAP* (1.1). *Let* $y_i^{(0)} \in X$ *be the starting dual variables for each component* $i \in \{1,\ldots,m\}$. *Set* $x^{(0)} = d - \sum_{i=1}^m \lambda_i y_i^{(0)}$.

01 *For* $k = 1, 2, \ldots$
02   *For* $i = 1, 2, \ldots, m$ *(Parallel projection)*
03         $z_i^{(k)} := x^{(k-1)} + y_i^{(k-1)}$
04         $x_i^{(k)} = P_{C_i}(z_i^{(k)})$
05         $y_i^{(k)} = z_i^{(k)} - x_i^{(k)}$
06   *end for*
07     $x^{(k)} = \sum_{i=1}^m \lambda_i x_i^{(k)}$
08 *end for*

We give a brief explanation of the simultaneous Dykstra's algorithm. Lines 3 to 5 correspond to the projection onto $\mathcal{C}$. Line 7 corresponds to projection of $(x_1^{(k)},\ldots,x_m^{(k)})$ onto $\mathcal{D}$, i.e., $(x^{(k)},\ldots,x^{(k)}) = P_{\mathcal{D}}(x_1^{(k)},\ldots,x_m^{(k)})$. The advantage of the simultaneous Dykstra's algorithm is that lines 3 to 5 can be performed in parallel.

We now discuss the convergence rate of the simultaneous Dykstra's algorithm. We saw in Section 4 that the regularity condition (4.8) is a sufficient condition for $O(1/k)$ convergence. We now show that this regularity condition holds for the original problem if and only if it holds for the product space formulation.

Proposition 5.2. *(Equivalence of constraint qualification) Let* C_i *be closed convex sets for* $i \in \{1,\ldots,m\}$, *and let* $C = \cap_{i=1}^m C_i$. *Let* $\mathcal{C}$ *and* $\mathcal{D}$ *be as defined in* (5.1). *At a point* $x^* \in C$, *the conditions*

$$\begin{aligned}&\sum_{i=1}^m v_i = 0 \text{ and } v_i \in N_{C_i}(x^*) \text{ for all } i \in \{1,\ldots,m\}\\ &\qquad \text{implies } v_i = 0 \text{ for all } i \in \{1,\ldots,m\}\end{aligned} \tag{5.2}$$

and

$$\begin{aligned}&(v_1,\ldots,v_m) + (w_1,\ldots,w_m) = 0,\, (v_1,\ldots,v_m) \in N_{\mathcal{C}}(x^*,\ldots,x^*)\\ &\text{ and } (w_1,\ldots,w_m) \in N_{\mathcal{D}}(x^*,\ldots,x^*)\\ &\text{implies } (v_1,\ldots,v_m) = (w_1,\ldots,w_m) = 0\end{aligned} \tag{5.3}$$

are equivalent.

Proof. Note that $(v_1,\ldots,v_m) \in N_{\mathcal{C}}(x^*,\ldots,x^*)$ if and only if $v_i \in N_{C_i}(x^*)$ for all i. Next, since $\mathcal{D}$ is a linear subspace, we have $(w_1,\ldots,w_m) \in N_{\mathcal{D}}(x^*,\ldots,x^*)$ if and only if $(w_1,\ldots,w_m) \in \mathcal{D}^\perp$. Proposition 5.3 gives the equivalent condition $\sum \lambda_i w_i = 0$. So in other words,

$$\begin{aligned}&(v_1,\ldots,v_m) + (w_1,\ldots,w_m) = 0,\, (v_1,\ldots,v_m) \in N_{\mathcal{C}}(x^*,\ldots,x^*)\\ &\text{ and } (w_1,\ldots,w_m) \in N_{\mathcal{D}}(x^*,\ldots,x^*)\end{aligned}$$

is equivalent to

$$v_i \in N_{C_i}(x^*) \text{ and } w_i = -v_i \text{ for all } i \in \{1, \dots, m\}, \text{ and } \sum_{i=1}^{m} \lambda_i v_i = 0.$$

Conditions (5.2) and (5.3) are now easily seen to be equivalent. □

As is well known in the study of Dykstra's algorithm, no correction vectors for $\mathcal{D}$ are necessary since $\mathcal{D}$ is an affine space. But we need to elaborate on the correction vector to $\mathcal{D}$ before we show the derivation of $x^{(0)}$. Let this correction vector be $\tilde{w}^{(k)} = (w_1^{(k)}, \dots, w_m^{(k)})$. We have $\tilde{w}^{(k)} \in N_{\mathcal{D}}(x^{(k)}, \dots, x^{(k)})$. But since $\mathcal{D}$ is a linear subspace, we have $\tilde{w}^{(k)} \in \mathcal{D}^\perp$. We have the following easy result.

Proposition 5.3. *Let $\tilde{w} = (w_1, \dots, w_m)$ be a vector in X^m. Then $\tilde{w} \in \mathcal{D}^\perp$ if and only if $\sum \lambda_i w_i = 0$.*

Proof. This follows easily from the following chain:

$$\begin{array}{rl} & \tilde{w} \in \mathcal{D}^\perp \\ \iff & \langle \tilde{w}, v \rangle = 0 \text{ for all } v \in \mathcal{D} \\ \iff & \langle \sum \lambda_i w_i, v \rangle = 0 \text{ for all } v \in X \\ \iff & \sum \lambda_i w_i = 0. \end{array}$$

□

Let $\tilde{y}^{(k)} = (y_1^{(k)}, \dots, y_m^{(k)})$. The default starting vector for the simultaneous Dykstra's algorithm in [IP91] is $\tilde{y}^{(0)} = 0 \in X^m$, but we can warmstart Dykstra's algorithm as explained in Subsection 2.1. We now show that $x^{(0)} = d - \sum \lambda_i y_i^{(0)}$ is indeed the formula to warmstart the simultaneous Dykstra's algorithm.

Proposition 5.4. *(Formula for $x^{(0)}$) In Algorithm 5.1, for the starting dual vector $\tilde{y}^{(k)} = (y_1^{(k)}, \dots, y_m^{(k)}) \in X^m$, the starting iterate for $x^{(0)}$ is $x^{(0)} = d - \sum \lambda_i y_i^{(0)}$.*

Proof. Let $\tilde{w}^{(k)} = (w_1^{(k)}, \dots, w_m^{(k)})$ be the correction vector corresponding to $\mathcal{D}$. The iterates $(x^{(k)}, \dots, x^{(k)}) \in X^m$ lie in $\mathcal{D}$ for all k, and $(d, \dots, d) \in \mathcal{D}$. From our study of Dykstra's algorithm earlier, we have

$$(w_1^{(k)}, \dots, w_m^{(k)}) = (d, \dots, d) - (x^{(k)}, \dots, x^{(k)}) - (y_1^{(k)}, \dots, y_m^{(k)}).$$

Moreover, we have $\sum \lambda_i w_i^{(k)} = 0$ from Proposition 5.3, so $\sum \lambda_i (d - x^{(k)} - y_i^{(k)}) = 0$. Together with the fact that $\sum \lambda_i = 1$, we get the needed formula for $x^{(0)}$. □

We now look at how to improve Algorithm 5.1. Line 7 can be improved by projecting $(x_1^{(k)}, \dots, x_m^{(k)})$ onto a set better than $\mathcal{D}$. Recall that line 4 produces supporting halfspaces of the set C_i. Consider the set C_{m+1}^k defined as the intersection of the supporting halfspaces produced in line 4, and let $\mathcal{C}^k \subset X^m$ be defined by $\mathcal{C}^k = C_{m+1}^k \times \cdots \times C_{m+1}^k$ (m copies). We can add the set $\mathcal{C}^k$ to play the role of C_{m+1}^k in the extended Dykstra's algorithm (Algorithm 3.1) to enhance the algorithm.

5.1. **A two-level Dykstra's algorithm.** If we want to apply the SHQP strategy to enhance the simultaneous Dykstra's algorithm, then we might want to cut up the problem into smaller blocks so that the quadratic programs formed are defined by a manageable number of halfspaces. It is reasonable to assume that information about the sets C_i communicate upwards from the leaves to the root of a tree (in the sense of graph theory). We illustrate with an example with $m = 4$ where we break down the size of the quadratic programs to be at most 2. Let the sets $\mathcal{D}_1$ and $\mathcal{D}_2$ be defined by

$$\begin{aligned} \mathcal{D}_1 &= \{(x_1,x_2,x_3,x_4)\in X^4 : x_1 = x_2\} \\ \text{and } \mathcal{D}_2 &= \{(x_1,x_2,x_3,x_4)\in X^4 : x_3 = x_4\}. \end{aligned}$$

We present a two level Dykstra's algorithm.

Algorithm 5.5. *(Two level Dykstra's algorithm) Consider the BAP* (1.1) *where* $m = 4$. *Let* $y_i^{(0)} \in X$ *be the starting dual variables for each component* $i \in \{1,\dots,4\}$. *Set* $x^{(0)} = d - \sum_{i=1}^4 \lambda_i y_i^{(0)}$.

01 *For* $k = 1,2,\dots$
02 *For* $i \in \{1,2,3,4\}$
03 $z_i^{(k)} := x^{(k-1)} + y_i^{(k-1)}$
04 $x_i^{(k)} = P_{C_i}(z_i^{(k)})$
05 $y_i^{(k)} = z_i^{(k)} - x_i^{(k)}$
06 *end for*
07 $x_{(1,2)}^{(k)} = \frac{\lambda_1}{\lambda_1+\lambda_2}x_1^{(k)} + \frac{\lambda_2}{\lambda_1+\lambda_2}x_2^{(k)}$
08 $x_{(3,4)}^{(k)} = \frac{\lambda_3}{\lambda_3+\lambda_4}x_3^{(k)} + \frac{\lambda_4}{\lambda_3+\lambda_4}x_4^{(k)}$
09 $x^{(k)} = (\lambda_1+\lambda_2)x_{(1,2)}^{(k)} + (\lambda_3+\lambda_4)x_{(3,4)}^{(k)}$
10 *end for*

Lines 2 to 6 describe the operation involved in projecting onto $\mathcal{C}$, which is not different from the simultaneous Dykstra's algorithm (Algorithm 5.1). Line 7 describes the operation in projecting onto $\mathcal{D}_1$, line 8 describes the operation in projecting onto $\mathcal{D}_2$, and line 9 describes the operation in projecting onto $\mathcal{D}$.

The agent that collects information on $x_1^{(k)}$ and $x_2^{(k)}$ to obtain $x_{(1,2)}^{(k)}$ can also collect the halfspaces generated by the projection operation used to obtain $x_1^{(k)}$ and $x_2^{(k)}$. We can make use of these halfspaces to form a superset of C that plays the role of C_{m+1}^k in the extended Dykstra's algorithm (Algorithm 3.1). In other words, the operations in lines 8 to 12 of Algorithm 3.1 can be inserted between lines 7 and 8 of Algorithm 5.5. We can also insert these same lines between lines 8 and 9 and between lines 9 and 10 to enhance Algorithm 5.5. It is now easy to extend the principles highlighted here for problems involving $m > 4$ sets and with more than 2 levels.

6. Using the APG for (D')

In this section, we depart from the dual alternating minimization strategy treated in the rest of the paper, and discuss using the accelerated proximal gradient (APG) algorithm to solve (D') in (2.1) in order to get a $O(1/k^2)$ convergence rate. We remark that the APG can be augmented by the strategy described in Subsection 4.3.

We recall the APG as presented in [Tse08, Section 3], which traces its roots to Nesterov's second optimal method [Nes88]. We decide that it is best to adopt the notation of [Tse08] even though it conflicts with some of the notation we have used in the rest of the paper.

Algorithm 6.1. [Tse08, Algorithm 1] *Consider the problem of minimizing*

$$h(x) = f(x) + P(x),$$

where $f : X \to \mathbb{R}$ is a smooth convex function whose gradient $\nabla f : X \to X$ is Lipschitz with constant L, and $P : X \to \mathbb{R}$ is a (not necessarily smooth) convex function. For each $y \in X$, define $l_f(\cdot; y) : X \to \mathbb{R}$ (a linearization of $h(\cdot)$ at y) by

$$l_f(x; y) = f(y) + \langle \nabla f(y), x - y \rangle + P(x).$$

Choose $\theta_0 \in (0, 1]$, x_0, $z_0 \in \mathrm{dom}(P)$. $k \leftarrow 0$. Go to 1.

(1) *Choose a nonempty closed convex set $X_k \subset X$ with $X_k \cap \mathrm{dom}(P) \neq \emptyset$. Let*

$$y_k = (1 - \theta_k)x_k + \theta_k z_k \ , \tag{6.1}$$

$$z_{k+1} = \arg\min_{x \in X_k} \{l_f(x; y_k) + \tfrac{\theta_k L}{2} \|x - z_k\|^2\}, \tag{6.2}$$

$$\hat{x}_{k+1} = (1 - \theta_k)x_k + \theta_k z_{k+1}. \tag{6.3}$$

Choose x_{k+1} such that

$$h(x_{k+1}) \leq l_f(\hat{x}_{k+1}; y_k) + \tfrac{L}{2} \|\hat{x}_{k+1} - y_k\|^2. \tag{6.4}$$

Choose $\theta_{k+1} \in (0, 1]$ satisfying

$$\tfrac{1-\theta_{k+1}}{\theta_{k+1}^2} \leq \tfrac{1}{\theta_k^2}. \tag{6.5}$$

$k \leftarrow k + 1$, and go to 1.

The following is the convergence result of Algorithm 6.1. We simplify their result by taking $X_k = X$ for all k.

Theorem 6.2. [Tse08, Corollary 1(a)] *Let $\{(x_k, y_k, z_k, \theta_k, X_k)\}$ be generated by Algorithm 6.1 with $\theta_0 = 1$. Fix any $\epsilon > 0$. Suppose $\theta_k \leq \frac{2}{k+2}$ (which is the case when $\theta_0 = 1$ and θ_{k+1} is determined from θ_k by setting (6.5) to an equation), and $X_k = X$ for all k. Then for any $x \in \mathrm{dom}(P)$ with $h(x) \leq \inf(h) + \epsilon$, we have*

$$\min_{i=0,1,\ldots,k+1} \{h(x_i)\} \leq h(x) + \epsilon \text{ whenever } \quad k \geq \sqrt{\tfrac{4L}{\epsilon}} \|x - z_0\| - 2.$$

Even though the line (6.4) is different from that in [Tse08, (14)], it is easy to check that the inequality [Tse08, (23)] remains valid with this change.

Theorem 6.2 shows that the infimum of $\{h(x_k)\}_k$ produced by Algorithm 6.1 would converge to the infimum of $h(\cdot)$. Furthermore, if a minimizer of $h(\cdot)$ exists, then the convergence rate of $\{h(x_k) - \inf(h)\}_k$ is of $O(1/k^2)$.

For the BAP (1.1) of projecting a point onto the intersection of m sets, the function $h(\cdot)$ was described in (2.1). For $\tilde{y} \in X^m$, the mapping

$$\tilde{y} \mapsto \|d - \sum y_i\|^2 = f(y_1 + \cdots + y_m) \tag{6.6}$$

has Hessian

$$\begin{pmatrix} I & I & \cdots & I \\ I & I & & I \\ \vdots & & \ddots & \vdots \\ I & I & \cdots & I \end{pmatrix}$$

(i.e., there are m^2 blocks in an $m \times m$ block square matrix), and the gradient of the map in (6.6) is Lipschitz with constant $L = m$. The step (6.2) can now be easily carried out using Proposition 2.2(5) to obtain all m components of the minimizer

z_{i+1}. We can use the strategy described in Subsection 4.3 to get a better iterate x_{k+1} satisfying (6.4) than $\hat{x}_{k+1}$.

7. Conclusion

In this paper, we showed ways to incorporate the SHQP heuristic to improve Dykstra's algorithm. For the case when C_i are hyperplanes, the numerical experiments in [Pan15a] shows the effectiveness of the strategies explained in this paper. We defer further numerical experiments to future work.

Appendix A. Proof of convergence of Algorithm 3.1

In this appendix, we present the proof of convergence of Algorithm 3.1, the extended Dykstra's algorithm. We already saw that if $H_{m+1}^k = C_{m+1}^k = X$ for all $k \geq 0$, then Algorithm 3.1 reduces to the original Dykstra's algorithm. Apart from Theorem 3.4, our proof is mostly the same as the Boyle-Dykstra theorem [BD85] as presented in [Deu01b]. Note that the proof here also includes the warmstart case.

Throughout this section, we follow the notation of Algorithm 3.1. We need to follow the notation in [Deu01b] and define the sequences $\{e_i\}_{i=-m}^{\infty}$ and $\{\tilde{x}_i\}_{i=0}^{\infty}$ by

$$e_{(m+1)(k-1)+i} = y_i^{(k)} \tag{A.1}$$

$$\tilde{x}_{(m+1)(k-1)+i} = x_i^{(k)}. \tag{A.2}$$

The statement of Lemma A.6 makes the new notation more natural. We denote $[i]$ to be the integer in $\{1, \dots, m+1\}$ such that $m+1$ divides $i - [i]$.

Lemma A.1. *In Algorithm 3.1, for each $i \geq 1$, such that $[i] \in \{1, \dots, m\}$.*

$$\delta^*(e_i, C_{[i]} - y) = \langle \tilde{x}_i - y, e_i \rangle \geq 0 \text{ for all } y \in C_{[i]}. \tag{A.3}$$

Furthermore, if $[i] = m+1$, then

$$\delta^*(e_i, C_{m+1}^{i/(m+1)} - y) = \langle \tilde{x}_i - y, e_i \rangle \geq 0 \text{ for all } y \in C. \tag{A.4}$$

Proof. The proof of inequality (A.3) is exactly the same as [Deu01b, Lemma 9.17], but our statement is now only valid for all $n \geq 1$. We have

$$\begin{aligned} & \langle \tilde{x}_i - y, e_i \rangle \\ = \; & \left\langle P_{C_{[i]}}(\tilde{x}_{i-1} + e_{i-(m+1)}) - y, \tilde{x}_{i-1} + e_{i-(m+1)} - P_{K_{[i]}}(\tilde{x}_{i-1} + e_{i-(m+1)}) \right\rangle \geq 0, \end{aligned}$$

where the inequality is an immediate consequence from the properties of projections. The second inequality in (A.4) is also clear. The equations in both (A.3) and (A.4) are straightforward from the definition of $\delta^*(\cdot, \cdot)$. $\square$

Lemma A.2. *In Algorithm 3.1, for each $i \geq 0$,*

$$d - \tilde{x}_i = e_{i-m} + e_{i-(m-1)} + \cdots + e_{i-1} + e_i. \tag{A.5}$$

Proof. This is easily seen from lines 4 and 9 of Algorithm 3.1 and the formula for $z_i^{(k)}$ in (2.5). $\square$

Lemma A.3. *In Algorithm 3.1, $\{\tilde{x}_i\}$ is a bounded sequence, and*

$$\sum_{i=1}^{\infty} \|\tilde{x}_{i-1} - \tilde{x}_i\|^2 < \infty. \tag{A.6}$$

In particular,

$$\|\tilde{x}_{i-1} - \tilde{x}_i\| \to 0 \text{ as } i \to \infty. \tag{A.7}$$

Proof. Formula (A.6) is just a rephrasing of Theorem 3.4(2). Formula (A.7) follows easily.

We now show the boundedness of $\{\tilde{x}_i\}$. For i, let $k = \lfloor \frac{i}{m+1} \rfloor$. Define v_i as

$$v_i := \frac{1}{2}\|\tilde{x}_i - P_C(d)\|^2 + \sum_{l=i-m}^{i} \langle e_l, \tilde{x}_l - P_C(d)\rangle. \tag{A.8}$$

Recall the definition of $h^k(\cdot)$ in (3.1). We have

$$\begin{aligned}
v_i &= \frac{1}{2}\|\tilde{x}_i - P_C(d)\|^2 + \sum_{l=i-m}^{i} \langle e_l, \tilde{x}_l - P_C(d)\rangle \\
&= \frac{1}{2}\|x^{(k)}_{i-k(m+1)} - P_C(d)\|^2 + \sum_{l=1}^{i-k(m+1)} \delta^*(y_l^{(k)}, C_{[l]} - P_C(d)) \\
&\quad +\delta^*(y^{(k-1)}_{m+1}, C^{k-1}_{m+1} - P_C(d)) + \sum_{l=i-k(m+1)+1}^{m} \delta^*(y_l^{(k-1)}, C_{[l]} - P_C(d)) \\
&= h^{k-1}(y_1^{(k)}, y_2^{(k)}, \dots, y^{(k)}_{i-k(m+1)}, y^{(k-1)}_{i-k(m+1)+1}, \dots, y^{(k-1)}_{m+1}) \\
&\quad -\langle d, P_C(d)\rangle + \frac{1}{2}\|P_C(d)\|^2.
\end{aligned}$$

The proof of Theorem 3.4 shows that v_i is non-increasing.Since $0 \in C^{k-1}_{m+1} - P_C(d)$ and $0 \in C_{[l]} - P_C(d)$, we have $v_i \geq \frac{1}{2}\|\tilde{x}_i - P_C(d)\|^2$ (just like in Proposition 2.2(3)), which shows that $\{\tilde{x}_i\}$ is a bounded sequence. ◻

Lemma A.4. *In Algorithm 3.1, for any $i \in \mathbb{N}$,*

$$\|e_i\| \leq \sum_{k=1}^{i} \|\tilde{x}_{k-1} - \tilde{x}_k\| + \max_{1\leq l\leq m+1} \|e_{l-(m+1)}\|. \tag{A.9}$$

Proof. The proof is adjusted from [Deu01b, Lemma 9.21]. We induct on i. It is clear to see that (A.9) holds for all $i \in \{-m, \dots, 0\}$. Suppose (A.9) holds for all $r \leq i$. Let $M_1 = \max_{1\leq l\leq m+1} \|e_{l-(m+1)}\|$. Then

$$\begin{aligned}
\|e_{i+1}\| &= \|\tilde{x}_i - \tilde{x}_{i+1} + e_{i+1-(m+1)}\| \leq \|\tilde{x}_i - \tilde{x}_{i+1}\| + \|e_{i+1-(m+1)}\| \\
&\leq \|\tilde{x}_i - \tilde{x}_{i+1}\| + \sum_{k=1}^{i+1-(m+1)} \|\tilde{x}_{k-1} - \tilde{x}_k\| + M_1 \leq \sum_{k=1}^{i+1} \|\tilde{x}_{k-1} - \tilde{x}_k\| + M_1,
\end{aligned}$$

which implies that (A.9) holds for $r = i+1$. ◻

Lemma A.5. *In Algorithm 3.1,*

$$\liminf_i \sum_{k=i-m}^{i} |\langle \tilde{x}_k - \tilde{x}_i, e_k\rangle| = 0. \tag{A.10}$$

Proof. The proof needs to be adjusted from [Deu01b, Lemma 9.22]. Let $M_1 = \max_{1\leq l\leq m+1} \|e_{l-(m+1)}\|$. Using Schwarz's inequality and Lemma A.4, we get

$$\begin{aligned}
& \sum_{k=i-m}^{i} |\langle \tilde{x}_k - \tilde{x}_i, e_k\rangle| \\
\leq\ & \sum_{k=i-m}^{i} \|e_k\|\|\tilde{x}_k - \tilde{x}_i\| \\
\leq\ & \sum_{k=i-m}^{i} \left[\left(M_1 + \sum_{j=1}^{k} \|\tilde{x}_{j-1} - \tilde{x}_j\|\right) \|\tilde{x}_k - \tilde{x}_i\|\right] \\
\leq\ & \sum_{k=i-m}^{i} \left[\left(\sum_{j=1}^{k} \|\tilde{x}_{j-1} - \tilde{x}_j\|\right)\left(\sum_{l=k+1}^{i} \|\tilde{x}_{l-1} - \tilde{x}_l\|\right)\right] + M_1 \sum_{k=i-m}^{i} \|\tilde{x}_k - \tilde{x}_i\| \\
\leq\ & \underbrace{(m+1)\left(\sum_{j=1}^{i} \|\tilde{x}_{j-1} - \tilde{x}_j\|\right)\left(\sum_{l=i-(m-1)}^{i} \|\tilde{x}_{l-1} - \tilde{x}_l\|\right)}_{(1)} + \underbrace{M_1 \sum_{k=i-m}^{i} \|\tilde{x}_k - \tilde{x}_i\|}_{(2)}.
\end{aligned}$$

Term (2) converges to zero by Lemma A.3. Let $a_i = \|x_{i-1} - x_i\|$. To show our result, it suffices to show that

$$\liminf_i \left[\left(\sum_{j=1}^{i} a_j\right)\left(\sum_{l=i-(m-1)}^{i} a_l\right)\right] = 0$$

given that $\sum_{j=1}^{\infty} a_j^2$ is finite. We refer the reader to the proof in [Deu01b, Lemma 9.22] for the proof of this fact. □

Lemma A.6. *In Algorithm 3.1, there exists a subsequence $\{\tilde{x}_{i_j}\}$ of $\{\tilde{x}_i\}$ such that*

$$\limsup_j \langle y - \tilde{x}_{i_j}, d - \tilde{x}_{i_j}\rangle \leq 0 \text{ for each } y \in C, \text{ and} \tag{A.11}$$

$$\lim_j \sum_{k=i_j-m}^{i_j} |\langle \tilde{x}_k - \tilde{x}_{i_j}, e_k\rangle| = 0. \tag{A.12}$$

Proof. The proof is almost exactly the same as [Deu01b, Lemma 9.23]. Using Lemma A.2, we have for all $y \in C$, $i \geq m$ that

$$\begin{aligned}
\langle y - \tilde{x}_i, d - \tilde{x}_i\rangle &= \langle y - \tilde{x}_i, e_{i-m} + e_{i-m+1} + \cdots + e_i\rangle \\
&= \sum_{k=i-m}^{i} \langle y - \tilde{x}_i, e_k\rangle \\
&= \sum_{k=i-m}^{i} \langle y - \tilde{x}_k, e_k\rangle + \sum_{k=i-m}^{i} \langle \tilde{x}_k - \tilde{x}_i, e_k\rangle.
\end{aligned}$$

By Lemma A.1, the first sum is no more than 0. Hence

$$\langle y - \tilde{x}_i, d - \tilde{x}_i\rangle \leq \sum_{k=i-m}^{i} \langle \tilde{x}_k - \tilde{x}_i, e_k\rangle. \tag{A.13}$$

By Lemma A.5, we deduce that there is a subsequence $\{i_j\}_j$ such that (A.12) holds. Note that the right hand side of (A.13) does not depend on y. In view of (A.13), it follows that (A.11) also holds. □

Theorem A.7. *(Warmstart Boyle-Dykstra Theorem) Consider Algorithm 3.1. Define the sequence* $\{\tilde{x}_n\}$ *as in Step 2 of Algorithm 3.1 and* (A.2)*. Then*

$$\lim_i \|\tilde{x}_i - P_C(d)\| = 0.$$

Proof. The proof of this result is mostly the same as [Deu01b, Lemma 9.23]. By Lemma A.6, there exists a subsequence $\{\tilde{x}_{i_j}\}$ such that

$$\limsup_j \langle y - \tilde{x}_{i_j}, d - \tilde{x}_{i_j}\rangle \leq 0 \text{ for each } y \in C. \tag{A.14}$$

Since $\{\tilde{x}_i\}$ is bounded by Lemma A.3, it follows by [Deu01b, Theorem 9.12] (by passing to a further subsequence if necessary), that there is a $y_0 \in X$ such that

$$\tilde{x}_{i_j} \xrightarrow{w} y_0, \tag{A.15}$$

and

$$\lim_j \|\tilde{x}_{i_j}\| \text{ exists.} \tag{A.16}$$

By another property of Hilbert spaces ([Deu01b, Theorem 9.13]),

$$\|y_0\| \leq \liminf_j \|\tilde{x}_{i_j}\| = \lim_j \|\tilde{x}_{i_j}\|. \tag{A.17}$$

Since $[i]$ takes on only $m+1$ possibilities, an infinite number of the i_j's must be of the same value. If this value is in $\{1,\ldots,m\}$, say i_0, then an infinite number of the $\tilde{x}_{i_j}$'s lie in C_{i_0}. Since C_{i_0} is closed and convex, it is weakly closed by [Deu01b, Theorem 9.16], and hence $y_0 \in C_{i_0}$. By (A.7), $\tilde{x}_i - \tilde{x}_{i-1} \to 0$. By a repeated application of this fact, we see that all the sequences $\{\tilde{x}_{i_j+1}\}$, $\{\tilde{x}_{i_j+2}\}$, $\ldots$ converge weakly to y_0, and hence $y_0 \in C_j$ for every j. That is,

$$y_0 \in C.$$

For any $y \in C$, (A.17) and (A.14) imply that

$$\begin{aligned} \langle y - y_0, d - y_0\rangle &= \langle y, d\rangle - \langle y, y_0\rangle - \langle y_0, d\rangle + \|y_0\|^2 \\ &\leq \lim_j [\langle y, d\rangle - \langle y, \tilde{x}_{i_j}\rangle - \langle x_{i_j}, d\rangle + \|\tilde{x}_{i_j}\|^2] \\ &= \lim_j \langle y - \tilde{x}_{i_j}, d - \tilde{x}_{i_j}\rangle \leq 0. \end{aligned} \tag{A.18}$$

Hence $y_0 = P_C(d)$. Moreover, putting $y = y_0$ in (A.18), we get equality in the chain of inequalities, and hence

$$\lim_j \|\tilde{x}_{i_j}\|^2 = \|y_0\|^2 \tag{A.19}$$

and

$$\lim_j \langle y_0 - \tilde{x}_{i_j}, d - \tilde{x}_{i_j}\rangle = 0.$$

By (A.15) and (A.19), it follows from [Deu01b, Theorem 9.10(2)] that $\|\tilde{x}_{i_j} - y_0\| \to 0$. Hence

$$\|\tilde{x}_{i_j} - P_C(d)\| = \|\tilde{x}_{i_j} - y_0\| \to 0. \tag{A.20}$$

We now show an alternative strategy different from what was presented in [Deu01b, Theorem 9.24]. Recall the definition of v_i in (A.8). We have

$$\begin{aligned} v_i &= \frac{1}{2}\|\tilde{x}_i - P_C(d)\|^2 + \sum_{l=i-m}^{i} \langle e_l, \tilde{x}_l - P_C(d)\rangle \\ &= \frac{1}{2}\|\tilde{x}_i - P_C(d)\|^2 + \langle d - \tilde{x}_i, \tilde{x}_i - P_C(d)\rangle + \sum_{l=i-m}^{i} \langle e_l, \tilde{x}_l - \tilde{x}_i\rangle \\ &\leq \frac{1}{2}\|\tilde{x}_i - P_C(d)\|^2 + \|d - \tilde{x}_i\|\|\tilde{x}_i - P_C(d)\| + \sum_{l=i-m}^{i} \langle e_l, \tilde{x}_l - \tilde{x}_i\rangle. \end{aligned}$$

From (A.20) and (A.12), and the fact that $v_i \geq 0$, we have $\liminf_{i\to\infty} v_i = \lim_{j\to\infty} v_{i_j} = 0$. Since $\{v_i\}$ is nonincreasing, we have $\lim_{i\to\infty} v_i = 0$. Moreover, recall back in the proof of Lemma A.3 that $v_i \geq \frac{1}{2}\|\tilde{x}_i - y_0\|^2$. These facts combine to show us that $\tilde{x}_i \to y_0$, which is what we seek. □

References

[AT06] A. Auslender and M. Teboulle, *Interior gradient and proximal methods for convex and conic optimization*, SIAM J. Optim. **16** (2006), 697–725.

[Aus76] A. Auslender, *Optimisation*, Méthodes numériques, Maîtrise de Mathématiques et Applications Fondamentales, Masson, Paris, 1976.

[BB96] H.H. Bauschke and J.M. Borwein, *On projection algorithms for solving convex feasibility problems*, SIAM Rev. **38** (1996), 367–426.

[BC11] H.H. Bauschke and P.L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, 2011.

[BCK06] H.H. Bauschke, P.L. Combettes, and S.G. Kruk, *Extrapolation algorithm for affine-convex feasibility problems*, Numer. Algorithms **41** (2006), 239–274.

[BD85] J.P. Boyle and R.L. Dykstra, *A method for finding projections onto the intersection of convex sets in Hilbert spaces*, Advances in Order Restricted Statistical Inference, Lecture notes in Statistics, Springer, New York, 1985, pp. 28–47.

[BDHP03] H.H. Bauschke, F. Deutsch, H.S. Hundal, and S.-H. Park, *Accelerating the convergence of the method of alternating projections*, Trans. Amer. Math. Soc. **355** (2003), no. 9, 3433–3461.

[Bec15] A. Beck, *On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes*, SIAM J. Optim. **25** (2015), no. 1, 185–209.

[Ber99] D.P. Bertsekas, *Nonlinear programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[BT09] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sciences **2** (2009), no. 1, 183–202.

[BT13] A. Beck and L. Tetruashvili, *On the convergence of block coordinate descent type methods*, SIAM J. Optim. **23** (2013), no. 4, 2037–2060.

[CP15] A. Chambolle and T. Pock, *A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions*, manuscript.

[Deu01a] F. Deutsch, *Accelerating the convergence of the method of alternating projections via a line search: A brief survey*, Inherently Parallel Algorithms in Feasibility and Optimization and their Applications (D. Butnariu, Y. Censor, and S. Reich, eds.), Elsevier, 2001, pp. 203–217.

[Deu01b] ———, *Best approximation in inner product spaces*, Springer, 2001, CMS Books in Mathematics.

[DH94] F. Deutsch and H. Hundal, *The rate of convergence of Dykstra's cyclic projections algorithm: the polyhedral case*, Numer. Funct. Anal. Optimiz. **15** (1994), no. 5-6, 536–565.

[Dyk83] R.L. Dykstra, *An algorithm for restricted least-squares regression*, J. Amer. Statist. Assoc. **78** (1983), 837–842.

[ER11] R. Escalante and M. Raydan, *Alternating projection methods*, SIAM, 2011.

[FR15] O. Fercoq and P. Richtárik, *Accelerated, parallel, and proximal coordinate descent*, SIAM J. Optim. **25** (2015), no. 4, 1997–2023.

[GK89] W.B. Gearhart and M. Koshy, *Acceleration schemes for the method of alternating projections*, J. Comput. Appl. Math. **26** (1989), 235–249.

[GM89] N. Gaffke and R. Mathar, *A cyclic projection algorithm via duality*, Metrika **36** (1989), 29–54.

[GPR67] L.G. Gubin, B.T. Polyak, and E.V. Raik, *The method of projections for finding the common point of convex sets*, USSR Comput. Math. Math. Phys. **7** (1967), no. 6, 1–24.

[GS99] L. Grippo and M. Sciandrone, *Globally convergent block-coordinate techniques for unconstrained optimization*, Optim. Methods Softw. **10** (1999), 587–637.

[Han88] S.P. Han, *A successive projection method*, Math. Programming **40** (1988), 1–14.

[Hau68] Y. Haugazeau, *Sur les inéquations variationnelles et la minimisation de fonctionenelles convexes*, Ph.D. thesis, Université de Paris, 1968.

[HS15] N.J. Higham and N. Strabić, *Anderson acceleration of the alternating projections method for computing the nearest correlation matrix.*, preprint.

[IP91] A.N. Iusem and A.R. De Pierro, *On the convergence of Han's method of convex programming with quadratic objective*, Math. Programming **52** (1991), 265–284.

[JN11a] A. Juditsky and A. Nemirovski, *First-order methods for nonsmooth convex large-scale optimization, I: General purpose methods*, Optimization for Machine Learning (S. Sra, S. Nowozin, and S.J. Wright, eds.), MIT Press, 2011, pp. 1–28.

[JN11b] ———, *First order methods for nonsmooth convex large-scale optimization, II: Utilizing problem's structure*, Optimization for Machine Learning (S. Sra, S. Nowozin, and S.J. Wright, eds.), MIT Press, 2011, pp. 29–63.

[Kru06] A.Y. Kruger, *About regularity of collections of sets*, Set-Valued Anal. **14** (2006), 187–206.

[LLM11] G. Lan, Z. Lu, and R.D.C. Monteiro, *Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming*, Math. Program. **126** (2011), no. 1, 1–29.

[LR15] W. López and M. Raydan, *An acceleration scheme for Dykstra's algorithm*, Comput. Optim. Appl. (2015), Online first.

[LT93] Z.-Q. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res. **46** (1993), 157–178.

[Mal04] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl. **26** (2004), 272–284.

[Nes83] Y. Nesterov, *A method for solving a convex programming problem with rate of convergence $O(\frac{1}{k^2})$*, Soviet Math. Doklady **269** (1983), no. 3, 543–547, (in Russian).

[Nes88] ———, *On an approach to the construction of optimal methods of minimization of smooth convex functions*, Èkonom. i. Mat. Metody **24** (1988), 509–517.

[Nes04] ———, *Introductory lectures on convex optimization*, Kluwer, 2004.

[Nes05] ———, *Smooth minimization of nonsmooth functions*, Math. Program. **103** (2005), 127–152.

[Nes12] ———, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim. **22** (2012), 341–362.

[NY83] A. S. Nemirovski and D. B. Yudin, *Problem complexity and method efficiency in optimization*, Wiley Intersciences, 1983.

[Pan15a] C.H.J. Pang, *Accelerating the alternating projection algorithm for the case of affine subspaces using supporting hyperplanes*, Linear Algebra Appl. **469** (2015), 419–439.

[Pan15b] ———, *First order constrained optimization algorithms with feasibility updates*, (preprint).

[Pan15c] ———, *Set intersection problems: Supporting hyperplanes and quadratic programming*, Math. Program. Ser. A **149** (2015), 329–359.

[Pie84] G. Pierra, *Decomposition through formalization in a product space*, Math. Programming **28** (1984), 96–115.

[QS06] H. Qi and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl. **28** (2006), 360–385.

[RW98] R.T. Rockafellar and R.J.-B. Wets, *Variational analysis*, Grundlehren der mathematischen Wissenschaften, vol. 317, Springer, Berlin, 1998.

[Tse08] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, manuscript.

Current address: Department of Mathematics, National University of Singapore, Block S17 08-11, 10 Lower Kent Ridge Road, Singapore 119076
E-mail address: matpchj@nus.edu.sg