CONVERGENCE OF STRING-AVERAGING PROJECTION SCHEMES FOR INCONSISTENT CONVEX FEASIBILITY PROBLEMS

Yair Censor and Eli Tom
Department of Mathematics, University of Haifa,
Mt. Carmel, Haifa 31905, Israel.
(yair@math.haifa.ac.il, eli.tom@intel.com)

November 12, 2002. Revised: May 25, 2003. Final Revision: August 4, 2003

Abstract

We study iterative projection algorithms for the convex feasibility problem of finding a point in the intersection of finitely many nonempty, closed and convex subsets in the Euclidean space. We propose (without proof) an algorithmic scheme which generalizes both the string-averaging algorithm and the block-iterative projections (BIP) method with fixed blocks and prove convergence of the string-averaging method in the inconsistent case by translating it into a fully sequential algorithm in the product space.

Keywords: Projection methods; convex feasibility; string-averaging; product space; inconsistent feasibility problem

1 INTRODUCTION

This paper discusses theoretically new algorithmic structures of iterative projection algorithms for solving the convex feasibility problem. Let $V = R^n$ be the n-dimensional Euclidean space and let $C_1, C_2, \ldots, C_m$, be nonempty closed convex subsets of V. The **convex feasibility problem** is to find a point $x^* \in C := \cap_{i=1}^m C_i$. If $C \neq \emptyset$ the problem is **consistent**, otherwise it is **inconsistent**. See Bauschke and Borwein [3] for a general overview of projection algorithms for the consistent case. In practical applications one often does not know in advance whether a given problem is consistent or not and any change of the sets C_i, $i = 1, 2, \ldots, m$, can turn a consistent problem into an inconsistent one and vice versa. Therefore, it is desirable to know the behavior of an algorithm for both the consistent and the inconsistent cases.

The algorithmic structures studied here use an arbitrary point $x^0 \in V$ as an initial approximation and generate a sequence $\left\{x^k\right\}_{k\geq 0} \subseteq V$ by repeated application of an algorithmic operator T, i.e., $x^{k+1} = T\left(x^k\right)$. For the consistent case any generated sequence should, ideally, converge to a limit point $x^* = \lim_{k\to\infty} x^k$ in C. Inconsistent problems pose two questions: Does the generated sequence $\left\{x^k\right\}_{k\geq 0}$ converge at all? If so, can its limit point be characterized? The algorithms we focus on here employ **orthogonal projections** (projections, for short) onto convex sets. Projections belong to the broader class of **nonexpansive operators**, whose properties we use to prove convergence and to characterize the limit point x^* whenever it exists. The algorithms under study differ in the amount of computation parallelism that they allow, which is a desirable feature when implementing such algorithms on parallel computers. The algorithms are **inherently parallel** which means, according to [11, Preface, p. vii], that they "are logically (i.e., in their mathematical formulations) parallel, not just parallelizable under some conditions, such as when the underlying problem is decomposable in a certain manner". We study, in particular, the **string-averaging** algorithm, recently proposed and studied by Censor, Elfving and Herman [12] and the **block-iterative projections (BIP)** algorithm of Aharoni and Censor [1]. The string-averaging algorithm projects a point sequentially along several independent strings of constraints. Projecting along each string is sequential, but the strings are independent and projecting along them can be performed in parallel. The BIP algorithm projects a point onto blocks of constraints sets, moving sequentially from one block to the next. Projecting a point onto a block (of sets) is parallel since it is done by projecting the point on each set independently and then combining the projections. The behavior of these algorithms for the consistent case is known, i.e., both algorithms converge to a point common to all sets of the convex feasibility problem.

The contribution of this paper is two-fold. First we develop a general algorithmic scheme which generalizes both the string-averaging and BIP algorithms. While a proof of convergence of this general algorithmic scheme, for the general case, is still missing, the scheme is of interest due to its great practical potential and due to the fact that its special cases of the string-averaging and BIP algorithms have been proven to converge in the consistent case. Secondly, we provide, for the string-averaging algorithm a convergence result for the inconsistent case, which is a new finding because this algorithm was until now studied only for the consistent case [12]. Our analysis is done by translating the string-averaging algorithm into a fully sequential algorithm in a product space and applying a convergence result for the sequential algorithm. We extend Gubin, Polyak and Raik's theorem of cyclic convergence of sequential projection algorithms for the inconsistent case [18, Theorem 2] so that it can cover also affine (thus unbounded) sets, and use this extension in the product space. Our work complements in a new way earlier results for convergence of fully simultaneous projection algorithms both in the consistent and the inconsistent case, see, e.g., Combettes [15], and for sequential projection algorithms, see, Bauschke, Borwein and Lewis [4, Section 5]. See also Reich [22] and Goebel and Reich [17]. The paper is laid out as follows. The string-averaging algorithm and the BIP method are reviewed in Section 3 and the general algorithmic scheme is presented. Our extension of the theorem of Gubin, Polyak and Raik is presented in Section 4 along with the product space formulation. Finally, the convergence of the string-averaging algorithm in the inconsistent case is given in Section 5. Since the appearance of the first paper on the string-averaging method [12], more work was done on this algorithm by Crombez [16] who generalized it to the problem of finding fixed points of strict paracontractions in the consistent case, by Hyangjoo Rhee [23] who applied it to a problem in approximation theory, and by Bauschke, Matoušková and Reich [5] who discuss convergence in Hilbert spaces.

2 PRELIMINARIES

In the n-dimensional Euclidean space $V = R^n$ with Euclidean scalar product $\langle \cdot, \cdot \rangle$ norm $\|\cdot\|$ and distance function d, the distance between a point and a set is defined by $d(x, C) := \inf \{d(x, y) \mid y \in C\}$. The projection of a point $x \in R^n$ onto a set $\Omega \subseteq R^n$ is denoted by $P_\Omega(x)$ and defined as a point of Ω for which $P_\Omega(x) = \operatorname{argmin}\{\| x - z \| \mid z \in \Omega\}$. Projections belong to a broader class of nonexpansive operators some of whose properties we shall use in our work. Let $T : V \to V$, then T is nonexpansive on V if

$$\|T(x) - T(y)\| \leq \|x - y\|, \text{ for all } x, y \in V. \tag{1}$$

It is well-known that projection operators are nonexpansive as shown, e.g., by Cheney and Goldstein [14, Theorem 3], in the next proposition.

Proposition 2.1 If C a is nonempty closed convex subset of V then for all $x, y \in V$ we have $\|P_C(x) - P_C(y)\| \leq \|x - y\|$, and equality holds only if $\|x - P_C(x)\| = \|y - P_C(y)\|$.

Combining nonexpansive operators is done by composition or by convex combination.

Definition 2.2 $\omega \in R^m$ is a `weight vector` if $\omega_i \geq 0,\ i = 1, 2, \ldots, m$, and $\sum_{i=1}^m \omega_i = 1$.

Proposition 2.3 If $T_1, T_2, \ldots, T_m$, are nonexpansive operators and $\omega \in R^m$ is a weight vector then

(i) the composition $T_m \cdots T_2 T_1$ is nonexpansive, and

(ii) the convex combination $\sum_{i=1}^m \omega_i T_i$ is nonexpansive.

Applying the last propositions to projections implies that a finite composition of projections as well as a convex combination of projections are nonexpansive operators.

3 PROJECTION ALGORITHMS

The algorithmic schemes for solving the convex feasibility problem that we study here employ an algorithmic operator that combines projections in a special manner. The algorithmic operator T and the algorithmic scheme itself are sequential, simultaneous or have both properties depending on the way that projections are combined. In all algorithms, the starting point $x^0 \in R^n$ is arbitrary. Sequential algorithms project the current point x^k (sequentially) onto the next set to produce x^{k+1}, i.e., $T = P_m \cdots P_2 P_1$, where P_i is the projection onto C_i, $i = 1, 2, \ldots, m$. Simultaneous algorithms project x^k onto **all** m sets simultaneously and produce x^{k+1} as a positive convex combination of those projections, i.e., $T = \sum_{i=1}^{m} \omega_i P_i$, where ω is a positive weight vector. Throughout this paper we use the following general algorithm.

Algorithm 3.1 (General Algorithm).
Initialization: $x^0 \in V$ is an arbitrary starting point.
Iterative Step: Given x^k compute x^{k+1} by

$$x^{k+1} = T\left(x^k\right). \tag{2}$$

Next we describe the string-averaging and the fixed-blocks BIP algorithms, followed by the general algorithmic scheme which generalizes them.

3.1 The String-Averaging Algorithm

Let $C_1, C_2, \ldots, C_m$, be nonempty closed convex subsets of V. Let the **string** I_t be a finite nonempty subset of $\{1, 2, \ldots, m\}$, for $t = 1, 2, \ldots, S$, of the form

$$I_t = \left(i_1^t, i_2^t, \ldots, i_{\gamma(I_t)}^t\right), \tag{3}$$

where the length of the string I_t, denoted by $\gamma(I_t)$, is the number of elements in I_t. The **projection along the string** I_t operator is defined as the composition of projections onto the sets indexed by I_t, that is,

$$T_t := P_{i_{\gamma(I_t)}^t} \cdots P_{i_2^t} P_{i_1^t}, \text{ for } t = 1, 2, \ldots, S. \tag{4}$$

Given a positive weight vector $\omega \in R^S$ we define the algorithmic operator

$$T = \sum_{t=1}^{S} \omega_t T_t. \tag{5}$$

Using this T in Algorithm 3.1 gives the string-averaging algorithm.

3.2 The Fixed-Blocks Block-Iterative Algorithm

Given a weight vector ω, a **block with respect to** ω is the subfamily of sets $\{C_i \mid \omega_i > 0\}$. The fixed-blocks BIP algorithm is a special case of the BIP algorithm of [1] which allows variable blocks, see also [13, Algorithm 5.6.1]. It projects the current iterate x^k successively onto B fixed blocks which are given by weight vectors $\omega^1, \omega^2, \ldots, \omega^B \in R^m$. The **projection onto the block** ω operator is defined by

$$P_\omega := \sum_{i=1}^{m} \omega_i P_i. \tag{6}$$

The algorithmic operator T is now

$$T = P_{\omega^B} \cdots P_{\omega^2} P_{\omega^1}, \tag{7}$$

and using this T in Algorithm 3.1 gives the fixed-blocks BIP algorithm. We restrict ourselves to fixed blocks because we can assure convergence for inconsistent problems only for this case. If the blocks are not fixed over the iterations and the problem is inconsistent, it is possible to find a series of blocks for each iteration that will toggle the point x^k between two or more non-overlapping sets.

3.3 The General Algorithmic Scheme

The general algorithmic scheme that we propose here performs projections along blocks of strings. At the end of each block of strings the end points of the strings are averaged and the averaged point is projected onto the next block of strings. The string-averaging algorithm is a special case where only one block of strings is used and the fixed-blocks BIP algorithm is a special case in which the strings in all blocks contain only one subset, i.e., the length of all strings is 1. To clarify the algorithmic structure we first discuss an example and proceed with the formal definition later. At this time we are unable to prove the convergence of this general algorithmic scheme.

Example 3.2 The algorithmic operator T that we use in this example is illustrated in Figure 1. The number of blocks of strings here is $B = 3$. For each block we define the strings that it uses, the positive weight vector for averaging, and some additional parameters. For each block of strings

$b = 1, 2, 3$, the number of strings in the block is denoted by S_b, the length of the block of strings is denoted by L_b and defined as the length of its longest string. For $b = 1, 2, 3$, the strings are denoted by $I_{b,t}$, where $t = 1, 2, \ldots, S_b$. These parameters are all summarized in the table given below.

Block of strings 1	Block of strings 2	Block of strings 3
$I_{1,1} = (2, 4, 2, 3)$	$I_{2,1} = (1, 5)$	$I_{3,1} = (3)$
$I_{1,2} = (1, 3)$	$I_{2,2} = (4, 2, 1)$	$I_{3,2} = (5, 1)$
$I_{1,3} = (5)$	$---$	$---$
$\omega^1 = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\right)$	$\omega^2 = \left(\frac{1}{3}, \frac{2}{3}\right)$	$\omega^3 = \left(\frac{1}{2}, \frac{1}{2}\right)$
$L_1 = 4$	$L_2 = 3$	$L_3 = 2$
$S_1 = 3$	$S_2 = 2$	$S_3 = 2$

The projection along the string $I_{b,t}$ operator $T_{b,t}$ is defined as in (4), for example, $T_{1,1} = P_3P_2P_4P_2$. The projection along the block operator T_b is defined as a convex combination of the end points of its strings, for example, $T_1 = \sum_{t=1}^{S_1} \omega_t^1 T_{1,t} = \frac{1}{4}T_{1,1} + \frac{1}{4}T_{1,2} + \frac{1}{2}T_{1,3}$. Finally, the algorithmic operator T is the composition of projections along the blocks, i.e., $T = T_3T_2T_1$.

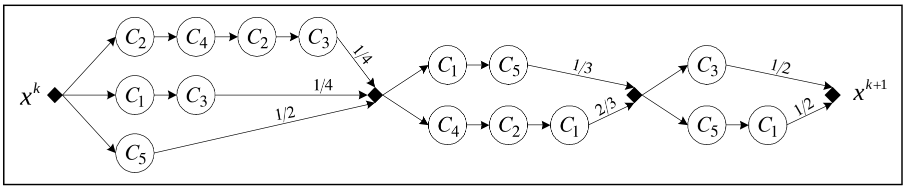

Figure 1: An example of the operator in the general algorithmic scheme.

We now give a formal definition of the algorithmic operator T of the general algorithmic scheme. This operator is composed of projections along B blocks of strings. Each block contains S_b, $b = 1, 2, \ldots, B$, strings, and has a positive weight vector

$$\omega^b = \left(\omega_1^b, \omega_2^b, \ldots, \omega_{S_b}^b\right) \in R^{S_b}, \ b = 1, 2, \ldots, B, \tag{8}$$

associated with it. We denote by $I_{b,t}$ the t-th string in the b-th block of strings. Each $I_{b,t}$ is a finite nonempty sequence whose elements belong to the set of $\{1, 2, \ldots, m\}$, i.e., for each $b = 1, 2, \ldots, B$,

$$I_{b,t} = \left(i_1^{b,t}, i_2^{b,t}, \ldots, i_{\gamma(I_{b,t})}^{b,t}\right), \ t = 1, 2, \ldots, S_b, \tag{9}$$

where $\gamma(I_{b,t})$ denotes the length of the string $I_{b,t}$. The elements of the sequence $I_{b,t}$ need not be distinct. For each block of strings, $b = 1, 2, \ldots, B$, and each string, $t = 1, 2, \ldots, S_b$, we define the projection along the string $I_{b,t}$ operator by

$$T_{b,t} := P_{i^{b,t}_{\gamma(I_{b,t})}} \cdots P_{i^{b,t}_2} P_{i^{b,t}_1}. \tag{10}$$

The projection along the block of strings b operator T_b is defined as a positive convex combination of the projections along the strings of the block, i.e.,

$$T_b := \sum_{\tau=1}^{S_b} \omega^b_\tau T_{b,\tau}, \; b = 1, 2, \ldots, B. \tag{11}$$

Finally, the algorithmic operator is defined by

$$T := T_B \cdots T_2 T_1. \tag{12}$$

The new general algorithmic scheme is formed by using the operator T of (12) in Algorithm 3.1 and yields the following.

Algorithm 3.3 (The General Algorithmic Scheme).
Initialization: $x^0 \in V$ is an arbitrary starting point.
Iterative Step: Given x^k compute x^{k+1} by

$$x^{k+1} = T_B \cdots T_2 T_1 \left(x^k\right). \tag{13}$$

where T_b, $b = 1, 2, \ldots, B$, are as in (11).

A general convergence analysis for Algorithm 3.3 is still missing. In the sequel, however, we establish the new finding that the string-averaging algorithm, which was proposed and studied for the consistent case by Censor, Elfving and Herman [12], converges in the inconsistent case.

4 TOOLS FOR THE CONVERGENCE ANALYSIS

Our approach to proving convergence of the string-averaging algorithm in the inconsistent case is based on showing that the algorithmic operator has a fully sequential equivalent in a product space. Then we apply our generalization (Theorem 4.4 below) of Gubin, Polyak and Raik's convergence theorem [18] for sequential algorithms in the inconsistent case.

4.1 Cyclic Convergence of Sequential Algorithms in the Inconsistent Case

Consider the following cyclically controlled sequential iterative projection algorithm for the convex feasibility problem.

Algorithm 4.1 (The Successive Projections Algorithm).
Initialization: $x^0 \in V$ is an arbitrary starting point.
Iterative Step: Given x^k compute x^{k+1} by

$$x^{k+1} = P_{(k \bmod m)+1}\left(x^k\right). \tag{14}$$

where m is the number of sets.

This algorithm originates in Bregman [6] and is also known by the name "projections onto convex sets" (POCS), see, e.g., Stark and Yang [24] or [13, Chapter 5]. Gubin, Polyak and Raik proved convergence of this algorithm, regardless of the consistency of the given convex feasibility problem [18, Theorem 2]. They showed that any cyclic subsequence of points lying in the same subset converges to a limit point in the subset. As a result, the limiting sequence is a cyclic m fixed-points sequence, where m is the number of subsets, i.e., the limiting sequence consists of the points $\{x^{*,1}, x^{*,2}, \ldots, x^{*,m}\}$ such that $x^{*,2} = P_2\left(x^{*,1}\right), \ x^{*,3} = P_3\left(x^{*,2}\right), \ \ldots, \ x^{*,1} = P_1\left(x^{*,m}\right)$.

Theorem 4.2 (Theorem 2 of [18]). Let $C_1, C_2, \ldots, C_m$, be nonempty closed convex subsets of V, such that at least one of them (for explicitness, say C_1) is bounded. Then there exist points $x^{*,i} \in C_i, \ i = 1, 2, \ldots, m$, such that $P_{i+1}(x^{*,i}) = x^{*,i+1}, \ i = 1, 2, \ldots, (m-1)$, and $P_1(x^{*,m}) = x^{*,1}$, and for $i = 1, 2, \ldots, m$, we have

$$\lim_{k\to\infty} x^{km+i+1} - x^{km+i} = x^{*,i+1} - x^{*,i}, \tag{15}$$

$$\lim_{k\to\infty} x^{km+i} = x^{*,i}, \tag{16}$$

where $\left\{x^k\right\}_{k\geq 0}$ is any sequence generated by Algorithm 4.1.

We will use the fixed-point theorem of Browder and Petryshyn [10], given below, (see also Ortega and Rheinboldt [20, theorem 5.1.4]) instead of Browder's theorem, in the proof of Theorem 4.2.

Theorem 4.3 (Browder and Petryshyn's Theorem [10]). If T is a nonexpansive operator on a closed convex subset C such that $T(C) \subseteq C$ then T has a fixed-point in C if and only if the sequence $x^{k+1} = T\left(x^k\right),\ k = 0, 1, 2, \ldots$ is bounded for at least one $x^0 \in C$.

The generalization of Theorem 4.2 that we arrive at is as follows.

Theorem 4.4 Let $C_1, C_2, \ldots, C_m,$ be nonempty closed convex subsets of V. If for at least one set (for explicitness, say C_1) the cyclic subsequence (of points in C_1) $\left\{x^{km+1}\right\}_{k\geq 0}$ of a sequence $\{x^k\}_{k\geq 0}$, generated by Algorithm 4.1, is bounded for at least one $x^0 \in R^n$ then there exist points $x^{*,i} \in C_i,\ i = 1, 2, \ldots, m,$ such that $P_{i+1}(x^{*,i}) = x^{*,i+1},\ i = 1, 2, \ldots, (m-1),$ and $P_1(x^{*,m}) = x^{*,1}$, and for $i = 1, 2, \ldots, m,$ we have

$$\lim_{k\to\infty} x^{km+i+1} - x^{km+i} = x^{*,i+1} - x^{*,i}, \tag{17}$$

$$\lim_{k\to\infty} x^{km+i} = x^{*,i}, \tag{18}$$

where $\left\{x^k\right\}_{k\geq 0}$ is any sequence generated by Algorithm 4.1.

Proof. Careful analysis of the proof of Theorem 4.2, shows that Gubin, Polyak and Raik use the assumption on the boundedness of one of the sets, namely C_1, for precisely two purposes. Firstly, they rely on Browder's theorem [8, Theorem 1] (see also [9, Thoerem 8.1]) which states that if V is a uniformly convex Banach space and T is a nonexpansive operator from a bounded closed convex subset C of V to itself, then T has a fixed-point in C. They define

$$T = P_1 P_m \cdots P_3 P_2, \tag{19}$$

which is a nonexpansive operator from a bounded closed convex subset C_1 to itself and establish in this way the existence of the fixed-point $x^{*,1} \in C_1$ of T. This is taken care of in our theorem by the application of Theorem 4.3.

Secondly, they deduce from the boundedness of C_1 that the cyclic subsequence $\{x^{km+1}\}_{k\geq 0}$ of any sequence $\{x^k\}_{k\geq 0}$, generated by Algorithm 4.1, is always (regardless of the initial point x^0) bounded, thus has a convergent subsequence. We deduce this by observing that the operator T of (19) is Lipshitzian with Lipshitz constant one, thus also its powers T^k are, for every

$k \geq 0$. If $\widehat{x}^0 \in R^n$ is any other initial point for a sequence $\left\{\widehat{x}^k\right\}_{k\geq 0}$ generated by Algorithm 4.1, then

$$\| T^k(x^0) - T^k(\widehat{x}^0) \| \leq \| x^0 - \widehat{x}^0 \| \tag{20}$$

and, therefore, any sequence $\{x^k\}_{k\geq 0}$, generated by Algorithm 4.1, is always (regardless of the initial point x^0) bounded. In this respect see also Goebel and Reich [17, Theorem 5.2]. The remainder of the proof is identical to the proof of Theorem 4.2 in [18, pp. 12–13]. ■

The requirement in Theorem 4.2 that at least one subset (i.e., C_1) is bounded is only a sufficient condition for the conditions on C_1 in our Theorem 4.4, therefore, Theorem 4.4 generalizes Theorem 4.2. An important case occurs when all subsets are affine to which Theorem 4.2 is obviously not applicable. However, we can rely on Aharoni, Duchet and Wajnryb [2] (and later Meshulam [19]) who proved that any sequence produced by successive projections on finitely many affine subsets is bounded. Hence, our Theorem 4.4 covers this case and supplies an alternative proof for this case to Bauschke, Borwein and Lewis [4, Theorem 5.6.1] result on the limiting cycle of cyclic projections onto convex polyhedra. This is a significant improvement of Theorem 4.2 since many real-world applications are modeled as convex feasibility problems with all subsets being affine. For an exhaustive text on projection methods for systems of linear or nonlinear equations see, e.g., Brezinski [7].

4.2 The Product Space Setup

The product space setup, proposed by Pierra [21], enables the conversion of a simultaneous algorithm into a sequential one to which convergence results for sequential algorithms can be applied. Let the product space be $\mathsf{V} := V^m$ for some positive integer m. A vector $\mathsf{x} \in \mathsf{V}$ is $\mathsf{x} =(x^1, x^2, \ldots, x^m)$ where $x^i \in V$, $i = 1, 2, \ldots, m$. Given a positive weight vector $\omega \in R^m$, the scalar product in V, denoted and defined by

$$\langle\langle \mathsf{x}, \mathsf{y} \rangle\rangle := \sum_{i=1}^{m} \omega_i \left\langle x^i, y^i \right\rangle, \quad \text{for all } \mathsf{x}, \mathsf{y} \in \mathsf{V}, \tag{21}$$

induces the norm $||| \cdot |||$ and the distance function d (observe our notational convention to use bold-face letters for quantities associated with the product space). It is easy to verify that for all $\mathsf{x} \in \mathsf{V}$ we have $|||\mathsf{x}|||^2 = \sum_{i=1}^m \omega_i \left\|x^i\right\|^2$. The diagonal subset, defined by

$$\mathsf{D} := \{\mathsf{x} \in \mathsf{V} \mid \mathsf{x} = (x, x, \dots, x),\ x \in V\}, \tag{22}$$

is a subspace of V and the canonical mapping $\delta : V \to \mathsf{D}$ is defined by

$$\delta(x) := (x, x, \dots, x) \in \mathsf{D}. \tag{23}$$

Projections in the product space V can be characterized. Let $C_1, C_2, \dots, C_m$, be nonempty closed convex subsets of V and define

$$\mathsf{C} := C_1 \times C_2 \times \cdots \times C_m = \prod_{i=1}^m C_i. \tag{24}$$

It is clear that C is a nonempty closed convex subset of V since all C_i enjoy these properties in V. The following lemma characterizes the projections onto the product set C and onto the diagonal set D (see [21]).

Lemma 4.5 If, for all $i = 1, 2, \dots, m$, P_i is the projection onto the set C_i in V, and $\mathsf{x} \in \mathsf{V}$ then

(i) the projection of $\mathsf{x} \in \mathsf{V}$ onto C is

$$\mathsf{P}_{\mathsf{C}}(\mathsf{x}) = \left(P_1(x^1), P_2(x^2), \dots, P_m(x^m)\right), \tag{25}$$

(ii) if ω is the positive vector in the definition of the scalar product in V (21) then the projection of $\mathsf{x} \in \mathsf{V}$ onto D is

$$\mathsf{P}_{\mathsf{D}}(\mathsf{x}) = \left(\sum_{i=1}^m \omega_i x^i, \sum_{i=1}^m \omega_i x^i, \dots, \sum_{i=1}^m \omega_i x^i\right). \tag{26}$$

Equation (25) characterizes the projection onto a product set as the concatenated vector of projections onto the individual sets, while (26) characterizes the projection onto the diagonal set via the (positive) linear combination of the vector components of the given point.

5 CONVERGENCE OF THE STRING-AVERAGING ALGORITHM IN THE INCONSISTENT CASE

The product space is defined here by $\mathsf{V} = V^S$, where S is the number of strings of the form (3). The positive weight vector $\omega \in R^S$ which is used to average the strings in (5) is used also in the definition of the inner product (21). The length of the algorithmic operator T, denoted by L, is the length of the longest string, i.e.,

$$L := \max \{\gamma(I_t) \mid t = 1, 2, \ldots, S\}. \tag{27}$$

Strings shorter then L are extended with copies of V, and product sets, denoted by $\mathsf{C}_j,\ j = 1, 2, \ldots, L$, are defined by

$$\mathsf{C}_j := \prod_{t=1}^{S} C_{j,t} \tag{28}$$

where

$$C_{j,t} := \begin{cases} C_{i_j^t}, & \text{if } 1 \le j \le \gamma(I_t), \\ V, & \text{otherwise.} \end{cases} \tag{29}$$

Example 5.3 might help to clarify (29). The algorithmic operator T in the product space is defined by

$$\mathsf{T} := \mathsf{P}_{\mathsf{D}} \mathsf{P}_L \cdots \mathsf{P}_2 \mathsf{P}_1, \tag{30}$$

where $\mathsf{P}_j = \mathsf{P}_{\mathsf{C}_j},\ j = 1, 2, \ldots, L$, and P_{D} is the projection onto the diagonal subset D. We proceed by showing that the algorithmic operators in the original space V and in the product space V are equivalent.

Theorem 5.1 If T and T are the projection operators in V and V, defined in (5) and (30), respectively, then

$$\delta(T(x)) = \mathsf{T}(\delta(x)), \text{ for all } x \in V, \tag{31}$$

where δ is the canonical mapping as in (23).

Proof. Let $x \in V$ and $\mathsf{x} = \delta(x)$. By Lemma 4.5(i) and (28) we have, for all $j = 1, 2, \ldots, L$,

$$\mathsf{P}_j(\mathsf{x}) = (P_{j,1}(x), P_{j,2}(x), \ldots, P_{j,S}(x)) \tag{32}$$

where $P_{j,t} = P_{C_{j,t}}$. Using Lemma 4.5(i) repeatedly we may write

$$\begin{aligned} \mathsf{y} &:= \mathsf{P}_L \cdots \mathsf{P}_2 \mathsf{P}_1(\mathsf{x}) \\ &= (P_{L,1} \cdots P_{2,1} P_{1,1}(x),\ P_{L,2} \cdots P_{2,2} P_{1,2}(x),\ \ldots,\ P_{L,S} \cdots P_{2,S} P_{1,S}(x)). \end{aligned} \tag{33}$$

By Lemma 4.5(ii) averaging is done by projecting onto the diagonal subset D, thus, obtaining

$$\mathsf{P}_{\mathsf{D}}(\mathsf{y}) = \delta\left(\sum_{t=1}^{S} \omega_t y^t\right) = \delta(T(x)), \tag{34}$$

and the result follows. ■

Now we are ready to prove convergence of the string-averaging algorithm without consistency assumption. The idea of the proof is based on transforming the string-averaging algorithm into a sequential algorithm in the product space, applying Theorem 4.4 in that space and then translating the conclusion back to the original space. Identifying D with the set C_1 of Theorem 4.4 we obtain the following result.

Theorem 5.2 Let $C_1, C_2, \ldots, C_m$, be nonempty closed convex subsets of V. If for at least one $x^0 \in V$ the sequence $\{x^k\}_{k\geq 0}$, generated by the string-averaging algorithm (Algorithm 3.1 with T as in (5)), is bounded then it converges for any $x^0 \in V$.

Proof. Define $\mathsf{x}^0 = \delta(x^0) \in \mathsf{D}$ and let T be as in (30). Identifying D with the set C_1 of Theorem 4.4, and using also Theorem 5.1, we reach, by Theorem 4.4, the conclusion that any sequence $\{\mathsf{x}^k\}_{k\geq 0}$ generated by

$$\mathsf{x}^{k+1} = \mathsf{T}(\mathsf{x}^k) \tag{35}$$

converges in the product space. Using again Theorem 5.1, we conclude the required result. ■

Alternatively, we may identify the set C_1 of Theorem 4.4 with one of the sets $\{\mathsf{C}_j\}_{j=1}^{L}$ in the product space and obtain another set of sufficient conditions. The (appropriately modified versions of) comments made after Theorem 4.2 apply here. In particular, if one of the sets $\{C_i\}_{i=1}^{m}$ is bounded then all assumptions of Theorem 5.2 hold.

Example 5.3 Let V be a finite-dimensional Euclidean space and let $\{C_i\}_{i=1}^{5}$ be nonempty closed convex subsets of V. Define three strings $I_1 = (2, 4, 2, 3)$, $I_2 = (1, 3)$ and $I_3 = (5)$. The string projection operators are $T_1 = P_3P_2P_4P_2$, $T_2 = P_3P_1$ and $T_3 = P_5$. Choosing $\omega = (\frac{1}{4}, \frac{1}{4}, \frac{1}{2})$ we get $T = \frac{1}{4}T_1 + \frac{1}{4}T_2 + \frac{1}{2}T_3$. See Figure 2. The number of strings $S = 3$ and the length of the operator,

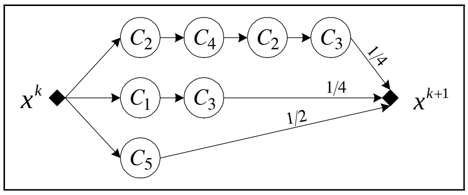

Figure 2: An illustration of the string-averaging algorithmic operator.

defined by its longest string, is $L = 4$. We show now how the operator T, defined in V, can be translated to a fully sequential operator T in some product space. Define $\mathsf{V} = V^5$ with a scalar product as in (21) using the weight vector ω. We use an $L \times S$ matrix Γ to construct L product sets to be projected onto. The matrix is constructed in the following manner. In the first column of Γ we write the subsets of the first string I_1, in the second column we write the subsets of the second string I_2, etc. We extend strings shorter then L by copies of V, thus, obtaining

$$\Gamma = \begin{bmatrix} C_2 & C_1 & C_5 \\ C_4 & C_3 & V \\ C_2 & V & V \\ C_3 & V & V \end{bmatrix}. \tag{36}$$

We produce L product sets, denoted by C_j, $j = 1, 2, \ldots, L$, as product of the sets in the rows of Γ yielding the sets (37).

$$\begin{array}{|l|}\hline \mathsf{C}_1 = C_2 \times C_1 \times C_5, \\ \hline \mathsf{C}_2 = C_4 \times C_3 \times V, \\ \hline \mathsf{C}_3 = C_2 \times V \times V, \\ \hline \mathsf{C}_4 = C_3 \times V \times V. \\ \hline \end{array} \tag{37}$$

The algorithmic operator T is then defined as,

$$\mathsf{T}(\mathsf{x}) := \mathsf{P}_{\mathsf{D}} \mathsf{P}_{\mathsf{C}_4} \mathsf{P}_{\mathsf{C}_3} \mathsf{P}_{\mathsf{C}_2} \mathsf{P}_{\mathsf{C}_1}, \tag{38}$$

where P_{D} is the projection onto the diagonal subset D of V. We need to show that the original operator T and the product space operator T are equivalent. We do so by showing that $\delta(T(x)) = \mathsf{T}(\delta(x))$ for every $x \in V$. Let $\mathsf{x} \in \mathsf{D}$, i.e., $\mathsf{x} = (x, x, x)$. By Lemma 4.5(i),

$$\mathsf{P}_{\mathsf{C}_1}(\mathsf{x}) = (P_{C_2}(x), P_{C_1}(x), P_{C_5}(x)). \tag{39}$$

Using Lemma 4.5(i) repeatedly and noting that $P_V(x) = x$ yields

$$\begin{aligned} \mathsf{y} &= \mathsf{P}_{\mathsf{C}_4} \mathsf{P}_{\mathsf{C}_3} \mathsf{P}_{\mathsf{C}_2} \mathsf{P}_{\mathsf{C}_1}(\mathsf{x}) \\ &= (P_{C_3} P_{C_2} P_{C_4} P_{C_2}(x),\ P_V P_V P_{C_3} P_{C_1}(x),\ P_V P_V P_V P_{C_5}(x)) \\ &= (P_{C_3} P_{C_2} P_{C_4} P_{C_2}(x),\ P_{C_3} P_{C_1}(x),\ P_{C_5}(x)) \\ &= (T_1(x),\ T_2(x),\ T_3(x)). \end{aligned} \tag{40}$$

Finally, averaging the strings is done, using Lemma 4.5(ii), by projection onto the diagonal subset D, thus,

$$\mathsf{T}(\mathsf{x}) = \mathsf{P}_{\mathsf{D}}(\mathsf{y}) = \delta\left(\sum_{t=1}^{S} \omega_t y^t\right) = \delta\left(\sum_{t=1}^{S} \omega_t T_t(x)\right) = \delta(T(x)). \tag{41}$$

Acknowledgments

We appreciate the careful reviews of two anonymous referees that helped us to properly revise and correct the paper. We thank Simeon Reich from the Department of Mathematics at the Technion in Haifa for his useful comments. This work was done at the Center for Computational Mathematics and Scientific Computation (CCMSC) of the University of Haifa and was supported by grant 592/00 of the Israel Science Foundation, founded by the Israel Academy of Sciences and Humanities.

References

[1] R. Aharoni and Y. Censor (1989). Block-iterative projection methods for parallel computation of solutions to convex feasibility problems. *Linear Algebra and its Applications*, **120**, 165–175.

[2] R. Aharoni, P. Duchet and B. Wajnryb (1984). Successive projections on hyperplanes. *Journal of Mathematical Analysis and Applications*, **103**, 134–138.

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

[4] H.H. Bauschke, J.M. Borwein and A. Lewis (1997). The method of cyclic projections for closed convex sets in Hilbert space. In: *Recent Developments in Optimization Theory and Nonlinear Analysis*. Y. Censor and S. Reich (Editors), Contemporary Mathematics, Vol. **204**, American Mathematical Society, Providence, RI, USA, 1–38.

[5] H.H. Bauschke, E. Matoušková and S. Reich. Projection and proximal point methods: convergence results and counterexamples. Technical report, June 14, 2003.

[6] L.M. Bregman (1965). The method of successive projection for finding a common point of convex sets. *Soviet Mathematics Doklady*, **6**, 688–692.

[7] C. Brezinski (1997). *Projection Methods for Systems of Equations*. Elsevier Science Publishers, Amsterdam, The Netherlands.

[8] F.E. Browder (1965). Nonexpansive nonlinear operators in a Banach space. *Proceedings of the National Academy of Sciences (USA)*, **54**, 1041–1044.

[9] F.E. Browder (1967). *Nonlinear Operators and Nonlinear Equations of Evolution in Banach Spaces*. American Mathematical Society, Providence, Rhode Island, USA.

[10] F.E. Browder and W.V. Petryshyn (1966). The solution by iteration of nonlinear functional equations in Banach spaces. *Bulletin of the American Mathematical Society*, **72**, 571–575.

[11] D. Butnariu, Y. Censor, and S. Reich (Editors) (2001). Inherently Parallel Algorithms in Feasibility and Optimization and their Applications. Elsevier Science Publishers, Amsterdam, The Netherlands.

[12] Y. Censor, T. Elfving and G.T. Herman (2001). Averaging strings of sequential iterations for convex feasibility problems. In: Inherently Parallel Algorithms in Feasibility and Optimization and their Applications. D. Butnariu, Y. Censor, and S. Reich (Editors), Elsevier Science Publishers, Amsterdam, The Netherlands, 101–114.

[13] Y. Censor and S.A. Zenios (1997). Parallel Optimization: Theory, Algorithms, and Applications. Oxford University Press, New York, NY, USA.

[14] W. Cheney and A.A. Goldstein (1959). Proximity maps for convex sets. Proceedings of the American Mathematical Society, 10, 448–450.

[15] P.L. Combettes (1994). Inconsistent signal feasibility problems: Least-square solutions in a product space. IEEE Transactions on Signal Processing, 42, 2955–2966.

[16] G. Crombez (2002). Finding common fixed points of strict paracontractions by averaging strings of sequential iterations. Journal of Nonlinear and Convex Analysis, 3, 345–351.

[17] K. Goebel and S. Reich (1984). Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings. Marcel Dekker, New York, NY, USA.

[18] L.G. Gubin, B.T. Polyak and E.V. Raik (1967). The method of projections for finding a common point of convex sets. Computational Mathematics and Mathematical Physics, 7, 1–24.

[19] R. Meshulam (1996). On products of projections. Discrete Mathematics, 154, 307–310.

[20] J.M. Ortega and W.C. Rheinboldt (1971). Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York , NY, USA.

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

[22] S. Reich (1983). A limit theorem for projections. Linear and Multilinear Algebra, **13**, 281–290.

[23] H. Rhee (2003). Application of the string averaging method to one-sided best simultaneous approximation, Journal of the Korea Society of Mathematical Education, Series B: The Pure and Applied Mathematics, **10**, 49–56.

[24] H. Stark and Y. Yang (1998). Vector Space Projections: A Numerical Approach to Signal and Image Processing. Neural Nets, and Optics, John Wiley & Sons, New York, NY, USA.