Uniqueness of DRS as the 2 Operator Resolvent-Splitting and Impossibility of 3 Operator Resolvent-Splitting

Ernest K. Ryu

September 14, 2018

Abstract

Given the success of Dougals–Rachford splitting (DRS), it is natural to ask whether DRS can be generalized. Are there are other 2 operator resolvent-splittings? Can DRS be generalized to 3 operators? This work presents the answers: no and no. In a certain sense, DRS is the unique 2 operator resolvent-splitting, and generalizing DRS to 3 operators is impossible without lifting, where lifting roughly corresponds to enlarging the problem size. The impossibility result further raises a question. How much lifting is necessary to generalize DRS to 3 operators? This work presents the answer by providing a novel 3 operator resolvent-splitting with provably minimal lifting that directly generalizes DRS.

1 Introduction

In 1979, Lions and Mercier presented Dougals–Rachford splitting (DRS) which solves the monotone inclusion problem

$$\text{find } x \in \mathbb{R}^d \text{ s.t. } 0 \in (A + B)x$$

with

$$z^{k+1} = (1 - \theta/2)z^k + (\theta/2)(2J_{\alpha A} - I)(2J_{\alpha B} - I)z^k$$

for any $\alpha > 0$ and $\theta \in (0, 2)$, where $A$ and $B$ are maximal monotone operators and $J_{\alpha A}$ and $J_{\alpha B}$ are their resolvents [30] [18] [24]. Since its introduction, DRS has enjoyed great popularity and has provided great value to the field of optimization.

Given the success of DRS, one may ask the following two questions:

1. Are there other 2 operator resolvent-splittings?
2. Can we generalize DRS to 3 operators?

In fact, the second question has been a long-standing open problem posed by Lions and Mercier themselves: “[T]he convergence seems difficult to prove ... in the case of a sum of 3 operators.” After all, identifying why a tool works and generalizing it is a common and often fruitful exercise in mathematics.

This work presents the answers to these questions: no and no. In a certain sense, DRS is the unique 2 operator resolvent-splitting. In a certain sense, there is no 3 operator resolvent-splitting without lifting, where lifting roughly corresponds to enlarging the problem size.

This impossibility result further raises the following question:

3. To generalize DRS to 3 operators, how much lifting is necessary?

This work presents the answer by providing a novel 3 operator resolvent-splitting with provably minimal lifting.
**Background.** To discuss what constitutes a generalization of DRS, we first point out a few key properties of DRS. Perhaps a generalization of DRS should satisfy these as well.

1. DRS is a *resolvent-splitting* in that it is constructed with scalar multiplication, addition, and resolvents.
2. DRS is *frugal* in that it uses $J_{\alpha A}$ and $J_{\alpha B}$ only once per iteration.
3. DRS *converges unconditionally* in that it works for any maximal monotone $A$ and $B$.
4. DRS uses *no lifting* in that the fixed-point mapping maps from $\mathbb{R}^d$ to $\mathbb{R}^d$, where $x \in \mathbb{R}^d$. In other words, DRS does not enlarge the problem size.

Consider the proximal point method (PPM) [27, 28, 35, 3], which finds an $x \in \mathbb{R}^d$ such that $0 \in Ax$ with

$$x^{k+1} = J_{\alpha A}x^k$$

for any $\alpha > 0$ and maximal monotone $A$. DRS generalizes PPM, and both methods are frugal, converge unconditionally, use no lifting, and rely on resolvents. Therefore, to require the 4 properties in a generalization of DRS seems reasonable.

Many other splittings have been presented since DRS, and they have certainly provided great value to the field of optimization. These splittings solve a wide range of different problem classes and are designed to be effective under a wide range of different computational considerations. Many of them include DRS as a special case and therefore are generalizations of DRS, in that sense. However, they do not satisfy the 4 stated properties and therefore are not generalizations of DRS, in this sense.

Forward-backward splitting (FBS) [29],

$$x^{k+1} = J_{\alpha B}(I - \alpha A)x^k,$$

which requires $A$ to be cocoercive, is frugal, uses no lifting, but is not a resolvent-splitting. Primal-dual hybrid gradient method [43, 31, 21, 8], also known as Chambolle–Pock,

$$x^{k+1} = J_A(x^k - \alpha u^k)$$
$$u^{k+1} = (I - J_B)(u^k + \alpha(2x^{k+1} - x^k))$$

is frugal but uses lifting. Davis–Yin splitting (DYS) [17], which finds an $x \in \mathbb{R}^d$ such that $0 \in (A + B + C)x$, where $C$ is cocoercive,

$$z^{k+1} = (I - J_{\alpha B} + J_{\alpha A} \circ (2J_{\alpha B} - I - \alpha C \circ J_{\alpha B})))z^k$$

is frugal, uses no lifting, but is not a resolvent-splitting. Other methods, such as FBFS [40], PPXA [14], PDFP²/O/PAPC [25, 9, 19], RFBS [1], Condat–Vu [16, 41], GFBS [33], PD3O [42], PDFP [10], AFBA [23], FBHFS [6], FDRS [4, 32], FRB [26], projective splitting [20, 14, 22], and the methods of [5, 12] all fail to satisfy the 4 properties.

**Organization of the paper.** In Section 2, we show that DRS is the only frugal, unconditionally convergent resolvent-splitting without lifting for the 2 operator problem. We do so by characterizing all frugal resolvent-splittings without lifting and showing that DRS is the only one among them that unconditionally converges.

In Section 3, we show that there is no resolvent-splitting without lifting for the 3 operator problem, even if the splitting is not frugal and not convergent. In particular, we show such a scheme without lifting cannot be a fixed-point encoding.

In Section 4, we define and quantify the notion of lifting for the 3 operator problem. We then provide a novel frugal, unconditionally convergent resolvent-splitting with provably minimal lifting for the 3 operator problem that directly generalizes DRS.

**Definitions.** We briefly review some standard notation and results of operator theory. Interested readers can find in-depth discussion of these concepts in standard references such as [36, 2].

Write $\langle \cdot, \cdot \rangle$ for the standard Euclidean inner product in $\mathbb{R}^d$. We say $A$ is an operator on $\mathbb{R}^d$ if $A$ maps points of $\mathbb{R}^d$ to subsets of $\mathbb{R}^d$. Given a matrix $M \in \mathbb{R}^{d \times d}$ also write $M : \mathbb{R}^d \to \mathbb{R}^d$ to denote the linear operator defined by the
matrix $M$. In particular, write $I$ for both the identity operator and the identity matrix. Write $\mathcal{M}(\mathbb{R}^d)$ for the set of all maximal monotone operators on $\mathbb{R}^d$. For any maximal monotone operator $A$ and $\alpha > 0$, write
\[ J_{\alpha A} = (I + \alpha A)^{-1} \]
for the resolvent of $A$. A mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ is nonexpansive if
\[ \|Tx - Ty\|^2 \leq \|x - y\|^2 \]
for all $x, y \in \mathbb{R}^d$. A mapping $F : \mathbb{R}^d \to \mathbb{R}^d$ is firmly nonexpansive if
\[ \|Fx - Fy\|^2 \leq \langle x - y, Fx - Fy \rangle \]
for all $x, y \in \mathbb{R}^d$. Resolvents are firmly nonexpansive. Given a mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ and a starting point $z^0 \in \mathbb{R}^d$, we call
\[ z^{k+1} = Tz^k \]
the fixed-point iteration with respect to $T$. A fixed-point iteration with respect to a nonexpansive mapping need not converge. A mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ is averaged if it can be expressed as $T = (1 - \theta)I + \theta R$, where $R : \mathbb{R}^d \to \mathbb{R}^d$ is nonexpansive and $\theta \in (0, 1)$. Note that $R$ and $T$ share the same fixed points. The fixed-point iteration with respect to an averaged mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ converges in that $z^k \to z^*$ where $Tz^* = z^*$, if a fixed point exists.

For any $A \in \mathcal{M}(\mathbb{R}^d)$, write zer $A = \{x : x \in Ax\}$ for the set of zeros of $A$. Consider the monotone inclusion problem of finding an element of zer $(A + B)$, where $A, B \in \mathcal{M}(\mathbb{R}^d)$. Peaceman–Rachford splitting (PRS) \cite{30, 24} is the fixed-point iteration
\[ z^{k+1} = (2J_{\alpha A} - I)(2J_{\alpha B} - I)z^k \]
with $\alpha > 0$. PRS is not guaranteed to converge. Douglas–Rachford splitting (DRS) is the fixed-point iteration
\[ z^{k+1} = (1 - \theta/2)z^k + (\theta/2)(2J_{\alpha A} - I)(2J_{\alpha B} - I)z^k \]
with $\alpha > 0$ and $\theta \in (0, 2)$. (Some may call this “relaxed PRS”.

\[ \text{DRS is guaranteed to converge in the sense that } z^k \to z^* \text{ for some } z^* \text{ where } J_{\alpha B}z^* \in \text{zer}(A + B), \text{if zer}(A + B) \text{ is not empty.} \]

2 Uniqueness of DRS as the unique frugal, unconditional 2 operator resolvent-splitting without lifting

In this section, we define what a frugal, unconditional 2 operator resolvent-splitting without lifting is and prove DRS is the only such splitting.

2.1 Definitions

When reading the definitions, it is helpful to think of DRS as a specific example. In the terminology and notation we soon establish, DRS is an unconditionally convergent frugal resolvent-splitting without lifting and $d' = d$, $T(A, B, z) = (1 - \theta/2)I + (\theta/2)(2J_{\alpha A} - I)(2J_{\alpha B} - I)$, and $S(A, B, z) = J_{\alpha B}$.

Given a dimension $d$, define the problem class $(\text{2op-}\mathbb{R}^d)$ to be the collection of monotone inclusion problems of the form
\[ \text{find } x \in \mathbb{R}^d \text{ s.t. } 0 \in (A + B)x \]
(2op-$\mathbb{R}^d$)
with $A, B \in \mathcal{M}(\mathbb{R}^d)$.

**Fixed-point encoding.** A pair of functions $(T, S)$ is a fixed-point encoding for the problem class $(\text{2op-}\mathbb{R}^d)$ if
\[ \exists z^* \in \mathbb{R}^{d'} \text{ such that } \begin{pmatrix} T(A, B, z^*) \nm S(A, B, z^*) \end{pmatrix} = \begin{pmatrix} z^* \nm x^* \end{pmatrix} \iff 0 \in (A + B)(x^*) \]
for all $A, B \in \mathcal{M}(\mathbb{R}^d)$. We call
\[ T : \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathbb{R}^{d'} \to \mathbb{R}^{d'} \]
the fixed-point mapping and
\[ S : M(\mathbb{R}^d) \times M(\mathbb{R}^d) \times \mathbb{R}^d \rightarrow \mathbb{R}^d, \]
the solution mapping. To clarify, a fixed-point encoding is defined for the entire problem class \([2op-\mathbb{R}^d]\), rather than a single instance of the monotone inclusion problem.

When we fix \(A, B \in M(\mathbb{R}^d)\), fixed points of \(T(A, B, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d\) correspond to zeros of \(A + B\). We say that points in \(\text{zer}(A + B)\) are encoded as fixed points of \(T(A, B, \cdot)\). For notational simplicity, we often drop the dependency on \(A\) and \(B\) and write \(Tz\) and \(Sz\) for \(T(A, B, z)\) and \(S(A, B, z)\).

In this section, we only consider \(d' = d\), as we limit our attention to fixed-point encodings without lifting (formally defined soon). In general, however, the dimension \(d\) of problems in \([2op-\mathbb{R}^d]\) and the dimension \(d'\) of the fixed-point mapping need not be the same. The purpose of allowing \(d' \neq d\) will become clearer later in Section 4, where an analogously defined \(d'\) is larger than \(d\).

Under this definition, DRS is a collection of fixed-point encodings. Each choice of \(d, \alpha > 0,\) and \(\theta \in (0, 2)\) in DRS gives a fixed-point encoding for the problem class \([2op-\mathbb{R}^d]\).

**Frugal resolvent-splitting without lifting.** Loosely speaking, \((T, S)\) is a resolvent-splitting for the problem class \([2op-\mathbb{R}^d]\) if it is a fixed-point encoding constructed with resolvents of \(A\) and \(B\), addition, and scalar multiplication. Loosely speaking, \((T, S)\) is frugal if it uses \(J_{\alpha A}\) and \(J_{\beta B}\) once, in that a single evaluation of \(J_{\alpha A}\) and a single evaluation of \(J_{\beta B}\) is used to evaluate both \(Tz\) and \(Sz\) for some \(z\). \((T, S)\) is without lifting if \(T(A, B, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d\) and \(S(A, B, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d\) for any \(A, B \in M(\mathbb{R}^d)\), i.e., if \(d' = d\).

We now make the definitions precise. Define the class of mappings
\[
F_0 = \{I\} \cup \{J_{\alpha A} \mid \alpha > 0\} \cup \{J_{\alpha B} \mid \alpha > 0\},
\]
where \(I\) is the “identity mapping” defined as \(I : M(\mathbb{R}^d) \times M(\mathbb{R}^d) \times \mathbb{R}^d \rightarrow \mathbb{R}^d\) and \(I(A, B, z) = z\) for any \(A, B \in M(\mathbb{R}^d)\) and \(z \in \mathbb{R}^d\). Recursively define
\[
F_{i+1} = \{F + G \mid F, G \in F_i\} \cup \{F \circ G \mid F, G \in F_i\} \cup \{\beta F \mid F \in F_i, \beta \in \mathbb{R}\}
\]
for \(i = 0, 1, 2, \ldots\). The “composition” \(F \circ G\) is defined with
\[
(F \circ G)(A, B, z) = F(A, B, G(A, B, z))
\]
for any \(z \in \mathbb{R}^d\) and \(A, B \in M(\mathbb{R}^d)\). Note that \(F_0 \subset F_1 \subset F_2 \subset \cdots\). Finally define
\[
F = \bigcup_{i=0}^{\infty} F_i.
\]
To clarify, elements of \(F\) map \(M(\mathbb{R}^d) \times M(\mathbb{R}^d) \times \mathbb{R}^d\) to \(\mathbb{R}^d\). If \(R \in F\) and \(A, B \in M(\mathbb{R}^d)\), then \(R(A, B, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d\).

As an aside, we could have defined \(F\) as the “near-ring” generated by \(J_{\alpha A}\) and \(J_{\beta B}\) for all \(\alpha > 0\) and \(\beta I\) for all \(\beta \in \mathbb{R}\). The set is not a ring because \(T \circ (U + V) \neq T \circ U + T \circ V\) for non-linear functions.

We say \((T, S)\) is a resolvent-splitting without lifting for the problem class \([2op-\mathbb{R}^d]\), if \((T, S)\) is a fixed-point encoding for the problem class \([2op-\mathbb{R}^d]\), and \(T, S \in F\).

When \(T, S \in F\), one can evaluate \(T(A, B, z)\) and \(S(A, B, z)\) for given \(z \in \mathbb{R}^d\) and \(A, B \in M(\mathbb{R}^d)\) in finitely many steps, where each step is scalar multiplication, vector addition, or a resolvent evaluation. We represent a procedure to evaluate \(T\) and \(S\) with a directed acyclic graph (DAG). Such evaluation procedures are not unique. As an example, Figure 1 shows two distinct ways to evaluate \(2J_{\alpha A}z\).

We say \((T, S)\), where \(T, S \in F\), is frugal if there is an evaluation procedure for \((T, S)\) that evaluates \(J_{\alpha A}\) once and \(J_{\beta B}\) once, for some \(\alpha > 0\) and \(\beta > 0\). More formally, if the evaluation procedure’s DAG has two nodes corresponding to the final values for \(Tz\) and \(Sz\) (i.e., the procedure evaluates both \(T\) and \(S\)) exactly one node evaluating \(J_{\alpha A}\) for some \(\alpha > 0\), and exactly one node evaluating \(J_{\beta B}\) for some \(\beta > 0\), then \((T, S)\) is frugal. See Figure 2 for an example.
Unconditional convergence. We say \((T, S)\) converges unconditionally for the problem class \(2\text{op-}\mathbb{R}^d\) if
\[
ST^k z^0 \to x^*, \quad x^* \in \text{zer}(A + B)
\]
for any \(z^0 \in \mathbb{R}^d\) and \(A, B \in \mathcal{M}(\mathbb{R}^d)\) as \(k \to \infty\), when \(\text{zer}(A + B)\) is not empty. To clarify, \(ST^k z^0 = S(A, B, T^k (A, B, z^0))\), and
\[
T^k = T \circ T \circ \cdots \circ T \quad k \text{ times}
\]

For example, with DRS, the \(z^k\)-iterates do not converge to a solution. Rather, \(z^k \to z^\star\), where \(J_{\alpha B} z^\star\) is a solution to the monotone inclusion problem, when a solution exists.

We say the convergence is unconditional, because there are no conditions on the monotone operators \(A, B \in \mathcal{M}(\mathbb{R}^d)\) or the starting point \(z^0 \in \mathbb{R}^d\). The fixed-point iteration with respect to \(T\) always, unconditionally finds a fixed point corresponding to a solution (provided the one condition that the solution set, \(\text{zer}(A + B)\), is nonempty holds).

Equivalence. Given a fixed-point iteration, we can scale it with a nonzero scalar to get another one that is essentially the same, i.e.,
\[
z^{k+1} = T(z^k) \iff az^{k+1} = aT(a^{-1}az^k)
\]
for any \(a \in \mathbb{R}\) such that \(a \neq 0\). Given resolvent-splitting, we can swap the role of \(A\) and \(B\) to get another one that is conceptually no different, i.e.,
\[(T(A, B, \cdot), S(A, B, \cdot)) \iff (T(B, A, \cdot), S(B, A, \cdot)).\]

Two resolvent-splittings without lifting are equivalent if one can be obtained from the other through scaling with a nonzero scalar and/or swapping the role of \(A\) and \(B\).

### 2.2 Uniqueness result

**Theorem 1.** Up to equivalence, \((T, S)\) is a frugal resolvent-splittings without lifting for the problem class \(2\text{op-}\mathbb{R}^d\) if and only if it is of the form
\[
x_1 = J_{\alpha A} z
\]
\[
x_2 = J_{\beta B} ((1 + \beta/\alpha)x_1 - (\beta/\alpha)z)
\]
\[
T(z) = z + \theta (x_2 - x_1)
\]
\[
S(z) = \eta x_1 + (1 - \eta)x_2
\]
for some $\alpha, \beta > 0$, $\theta \neq 0$, and $\eta \in \mathbb{R}$.

To clarify, if $(T, S)$ is a frugal resolvent-splittings without lifting for the problem class $2\text{op-R}^d$, then it is equivalent to a splitting stated in Theorem \[1\].

Since the splittings of Theorem \[1\] are fixed-point encodings, a fixed-point iteration with respect to $(T, S)$ of Theorem \[1\] stays at a fixed point (corresponding to a solution) if it starts at a fixed point (corresponding to a solution). However, we of course want a fixed-point iteration to converge to a fixed-point (corresponding to a solution) with other starting points. Theorem \[2\] characterizes the ones that do converge.

**Theorem 2.** If $d \geq 2$ $(T, S)$ of Theorem \[1\] converges unconditionally if and only if $\alpha = \beta$ and $\theta \in (0, 2)$.

When $\alpha = \beta$, the splitting $(T, S)$ of Theorem \[1\] is DRS (or relaxed PRS). Theorem \[1\] characterizes all frugal resolvent-splittings without lifting for the problem class $2\text{op-R}^d$, and Theorem \[2\] states that among them only DRS converges unconditionally, up to equivalence, when $d \geq 2$. Therefore, DRS is the collection of all frugal, unconditionally convergent resolvent splittings without lifting for the problem class $2\text{op-R}^d$, when $d \geq 2$.

When $d = 1$, the splittings $(T, S)$ of Theorem \[1\] may converge under more general conditions. However, we do not pursue this discussion, since monotone inclusion problems in 1 dimension are not very interesting.

### 2.3 Proof of Theorem \[1\]

Showing that $(T, S)$ of Theorem \[1\] is indeed a fixed-point encoding is straightforward. Let $x^* \in \mathbb{R}^d$ satisfy $0 \in (A + B)x^*$. Let $Ax^* \in Ax$ and $Bx^* \in Bx$ such that $Ax^* + Bx^* = 0$, and let $x_0 = x^* + \alpha Ax^*$. Then it is straightforward to verify that $x_1 = x_2 = x^*, Tz_0 = z_0, and Sz_0 = x^*$. On the other hand, assume $T(A, B, z^*) = z^*$. Then $x_1 = x_2$. Write $x^* = x_1 = x_2, Ax^* = (1/\alpha)(z^* - x^*), and Bx^* = (1/\alpha)(x^* - z^*)$. Then it is straightforward to verify $Ax^* \in Ax$, $Bx^* \in Bx$, and $Ax^* + Bx^* = 0$, which implies $x^* = S(z^*)$ is a solution.

We now need to show that any frugal resolvent-splitting without lifting for the problem class $2\text{op-R}^d$ is of the form of Theorem \[1\] up to equivalence. Before we do so, we will discuss the following lemma, which proof follows from standard linear algebra.

**Lemma 3.** Let $M \in \mathbb{R}^{m \times n}$ and $c \in \mathbb{R}^n$ be fixed coefficients, and let $v \in \mathbb{R}^n$ be a variable. Then the linear equalities $Mv = 0$ implies the linear equality $c^Tv = 0$ if and only if there is a $w \in \mathbb{R}^m$ such that $w^TM = c^T$.

To clarify, $Mv = 0$ implies $c^Tv = 0$ if all instances of the variable $v \in \mathbb{R}^n$ satisfying $Mv = 0$ satisfies $c^Tv = 0$. If $Mv = 0$ does not imply $c^Tv = 0$, there is an instance of $v \in \mathbb{R}^n$ such that $Mv = 0$ but $c^Tv \neq 0$.

**Proof of Lemma 3.** If there is a $w$ such that $w^TM = c^T$, then $Mv = 0$ implies $c^Tv = w^TMv = 0$. If there is no such $w$, then $c \notin \mathcal{R}(MT)$ and there is a $v \in \mathcal{N}(M)$ such that $c^Tv \neq 0$, where $\mathcal{R}$ and $\mathcal{N}$ respectively denote the range and null space. \[ \square \]

We now proceed onto the main proof. Let $(T, S)$ be any frugal resolvent-splitting without lifting.

Consider an evaluation procedure of $(T, S)$ and its DAG that establishes frugality. The DAG has one node each for the evaluations of $J_{\alpha A}$ and $J_{\beta B}$. With this DAG, we can find a step-by-step ordering of the computational steps. This ordering should respect the computational dependency represented in the DAG. Such an ordering is not unique. In this ordering, either $J_{\alpha A}$ or $J_{\beta B}$ is evaluated before the other. Without loss of generality, assume $J_{\alpha A}$ is evaluated before $J_{\beta B}$ in this ordering, since we can otherwise consider the equivalent splitting $(\tilde{T}, \tilde{S})$ defined with

$$
\tilde{T}(A, B, z) = T(B, A, z), \quad \tilde{S}(A, B, Z) = S(B, A, z).
$$

Consider the evaluation of $Tz_0$ and $Sz_0$ for $z_0 \in \mathbb{R}^d$. Write $z_1$ and $z_2$ for the points at which $J_{\alpha A}$ and $J_{\beta B}$ are evaluated. Write $x_1 = J_{\alpha A}z_1$ and $x_2 = J_{\beta B}z_2$. Define $\tilde{A}x_1$ and $Bx_2$ with $x_1 + \alpha \tilde{A}x_1 = z_1$ and $x_2 + \beta Bx_2 = z_2$. By definition of resolvents, we have $\tilde{A}x_1 \in A x_1$ and $Bx_2 \in B x_2$. 

6
Then we can express the evaluation of \( Tz_0 \) and \( Sz_0 \) as

\[
0 = \begin{bmatrix}
    *I_d & I_d & 0_d & 0_d & 0_d & 0_d & 0_d & 0_d & 0_d \\
    0_d & -I_d & I_d & 0_d & 0_d & 0_d & 0_d & 0_d & 0_d \\
    *I_d & *I_d & *I_d & I_d & 0_d & 0_d & 0_d & 0_d & 0_d \\
    0_d & 0_d & 0_d & -I_d & I_d & 0_d & 0_d & 0_d & \beta \cdot 0_d \\
    *I_d & *I_d & *I_d & *I_d & I_d & *I_d & *I_d & *I_d & *I_d \\
    \end{bmatrix}
\begin{bmatrix}
    z_0 \\
    z_1 \\
    x_1 \\
    z_2 \\
    x_2 \\
    Tz_0 \\
    \tilde{A}x_1 \\
    \tilde{B}x_2 \\
    Sz_0
\end{bmatrix},
\]

where the \( * \) denote unspecified scalar coefficients, \( I_d \) denotes the \( d \times d \) identity matrix, and \( 0_d \) denotes the \( d \times d \) zero matrix. We reduce the notational burden by replacing a scalar multiplied by \( I_d \) with just the scalar and write

\[
0 = \begin{bmatrix}
    * & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & -1 & 1 & 0 & 0 & \alpha & 0 & 0 & 0 \\
    * & * & * & 1 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & -1 & 1 & 0 & \beta & 0 & 0 \\
    * & * & * & * & 1 & * & * & * & 1
\end{bmatrix}
\begin{bmatrix}
    z_0 \\
    z_1 \\
    x_1 \\
    z_2 \\
    x_2 \\
    Tz_0 \\
    \tilde{A}x_1 \\
    \tilde{B}x_2 \\
    Sz_0
\end{bmatrix}.
\]

To clarify again, each scalar in the matrix should be interpreted as a \( d \times d \) block. With Gaussian elimination and reordering of the rows, we obtain the simpler equivalent system

\[
0 = \begin{bmatrix}
    -a & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    * & 0 & * & 1 & 0 & 0 & 0 & 0 & 0 \\
    * & 0 & * & 0 & * & 1 & 0 & 0 & 0 \\
    0 & -1 & 1 & 0 & 0 & \alpha & 0 & 0 & 0 \\
    0 & 0 & 0 & -1 & 1 & 0 & \beta & 0 & 0 \\
    * & 0 & * & 0 & * & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    z_0 \\
    z_1 \\
    x_1 \\
    z_2 \\
    x_2 \\
    Tz_0 \\
    \tilde{A}x_1 \\
    \tilde{B}x_2 \\
    Sz_0
\end{bmatrix}.
\]

We claim \( a \neq 0 \). Assume \( a = 0 \) for contradiction. Let

\[
A(x) = c_1x, \quad B(x) = c_2,
\]

where \( c_1 > 0 \) is unspecified and \( 0 \neq c_2 \in \mathbb{R}^d \). Then \( J_{\alpha A}0 = 0 \), and the mappings \( T \) and \( S \) are independent of the value of \( c_1 \). So the set of fixed points of \( T \) and the set of \( Sz^* \), where \( z^* \) is a fixed point of \( T \), do not depend on \( c_1 \).

On the other hand, since \( (T, S) \) is assumed to be a fixed-point encoding, there must be a \( z^* \) such that \( Tz^* = z^* \) and \( 0 \in (A + B)Sz^* \). So \(-c_1^{-1}c_2 = Sz^* \), which does depend on \( c_1 \), and we have a contradiction.

Knowing \( a \neq 0 \), we can absorb the top-left \( a \) into \( z_0 \) and left-multiply by an invertible matrix to get the equivalent system

\[
0 = \begin{bmatrix}
    1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & a & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 \\
    0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
    \end{bmatrix}
\begin{bmatrix}
    -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    * & 0 & * & 1 & 0 & 0 & 0 & 0 & 0 \\
    * & 0 & * & 0 & * & 1 & 0 & 0 & 0 \\
    0 & -1 & 1 & 0 & 0 & \alpha & 0 & 0 & 0 \\
    0 & 0 & 0 & -1 & 1 & 0 & \beta & 0 & 0 \\
    * & 0 & * & 0 & * & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    az_0 \\
    z_1 \\
    x_1 \\
    z_2 \\
    x_2 \\
    Tz_0 \\
    \tilde{A}x_1 \\
    \tilde{B}x_2 \\
    Sz_0
\end{bmatrix}.
\]
The **boldface symbols** denote where to pay attention in the linear systems. This further simplifies to the equivalent system

\[
0 = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
* & 0 & * & 1 & 0 & 0 & 0 & 0 \\
* & 0 & * & 0 & * & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
* & 0 & * & 0 & * & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
a_{z_0} \\
z_1 \\
x_1 \\
z_2 \\
x_2 \\
aT(a^{-1}a_{z_0}) \\
Ax_1 \\
Bx_2 \\
S(a^{-1}a_{z_0}) \\
\end{bmatrix}.
\]

By redefining \((T(z_0), S(z_0))\) to be the equivalent scaled splitting \((aT(a^{-1}a_{z_0}), S(a^{-1}a_{z_0}))\), we get the equivalent system

\[
0 = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\theta_1 & 0 & \theta_2 & 1 & 0 & 0 & 0 & 0 \\
\theta_3 & 0 & \theta_4 & 0 & \theta_5 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
\theta_6 & 0 & \theta_7 & 0 & \theta_8 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
x_1 \\
z_2 \\
x_2 \\
Tz_0 \\
Az_0 \\
Bz_0 \\
Sz_0 \\
\end{bmatrix},
\]

where \(\theta_1, \ldots, \theta_8\) are scalar parameters. To summarize our progress, we have shown that any frugal resolvent-splitting without lifting is equivalent to a frugal resolvent-splitting of the form \((1)\).

However, not all choices for the parameters \(\theta_1, \ldots, \theta_8\) make \((T, S)\) defined with \((1)\) a fixed-point encoding. We now analyze the requirements on \(\theta_1, \ldots, \theta_8\) that ensure \((T, S)\) is a fixed-point encoding, and thereby a frugal resolvent-splitting without lifting.

Consider the system of linear equalities with the fixed-point condition \(T(z_0) = z_0\) added

\[
0 = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\theta_1 & 0 & \theta_2 & 1 & 0 & 0 & 0 & 0 \\
\theta_3 & 0 & \theta_4 & 0 & \theta_5 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
\theta_6 & 0 & \theta_7 & 0 & \theta_8 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
x_1 \\
z_2 \\
x_2 \\
Tz_0 \\
Az_0 \\
Bz_0 \\
Sz_0 \\
\end{bmatrix}.
\]

System \((1)\) represents evaluation of \((T, S)\) at any point. System \((2)\) represents evaluation of \((T, S)\) at a fixed point, in the sense that when \(Mv = 0\), we have \(Tz_0 = z_0\) for \((T, S)\) defined with \((1)\).

We claim that the linear equalities \((2)\) must imply \(x_1 = x_2, Sz_0 = x_1, \text{ and } Ax_1 + Bx_2 = 0\). We prove these three implications one-by-one by assuming otherwise and constructing counter examples.

Assume for contradiction that \((2)\) does not imply the linear equality \(x_1 = x_2\). By Lemma \([\text{Lem}3]\) this means there is a specific instance

\[
v' = (z_0', x_1', z_2', x_2', T(z_0'), \tilde{Ax}_1', \tilde{Bx}_2, S(z_0')) \in \mathbb{R}^{9d}
\]

such that \(Mv' = 0\) but \(x_1' \neq x_2'\). Define

\[
A(x) = x - x'_1 + \tilde{Ax}_1', \quad B(x) = x - x'_2 + \tilde{Bx}_2'.
\]

\(A\) and \(B\) are monotone operators constructed to match the evaluations \(A(x_1') = \tilde{Ax}_1'\) and \(B(x_2') = \tilde{Bx}_2'.\) This implies \(T(A, B, z_0') = z_0'.\) Write \(x^* = S(A, B, z_0')\). Since \((T, S)\) is a fixed-point encoding, we have

\[
0 = (A + B)x^*.
\]
However, $x'_1 \neq x'_2$, so either $x'_1 \neq x^*$ or $x'_2 \neq x^*$ or both. Without loss of generality assume $x'_1 \neq x^*$. Define 

$$C(x) = 2(x - x'_1) + \tilde{A}x'_1.$$ 

Since $C(x'_1) = \tilde{A}x'_1$, we still have $T(C, B, z'_0) = z'_0$ and $S(C, B, z'_0) = x^*$. However, 

$$0 \notin (C + B)x^*,$$

since $C(x^*) \neq A(x^*)$. In other words, $T(C, B, z'_0) = z'_0$, but $S(C, B, z'_0)$ is not a zero of $C + B$. So $(T, S)$ fails to be a fixed-point encoding for $C, B \in \mathcal{M}(\mathbb{R}^d)$, and we have a contradiction.

Next, assume for contradiction that (2) does not imply the linear equality $S_{z_0} = x_1$. This means there is a specific instance 

$$v' = (z'_0, z'_1, x'_1, x'_2, T(z'_0), \tilde{A}x'_1, \tilde{B}x'_2, S(z'_0)) \in \mathbb{R}^{2d}$$ 

such that $Mv' = 0$ but $S(z'_0) \neq x'_1 = x'_2$. (We now know that $x'_1 = x'_2$.) Using the same definition of $A, B, C$, the same arguments carry over and we can establish $S(A, B, z'_0) = S(C, B, z'_0)$. Since we assumed (for contradiction) that $S(A, B, z'_0) \neq x'_1$, we have 

$$(A + B)(S(A, B, z'_0)) \neq (C + B)(S(C, B, z'_0)).$$

Remember that $A, B$, and $C$ are single-valued. So it is not possible for $S(A, B, z'_0) = S(C, B, z'_0)$ to be in both $\text{zer}(A + B)$ and $\text{zer}(C + B)$. Therefore $(T, S)$ fails to be a fixed-point encoding for the instance $A, B \in \mathcal{M}(\mathbb{R}^d)$ or $C, B \in \mathcal{M}(\mathbb{R}^d)$, and we have a contradiction.

Finally, assume for contradiction that (2) does not imply the linear equality $Ax_1 + \tilde{B}x_2 = 0$. This means there is a specific instance 

$$v' = (z'_0, z'_1, x'_1, x'_2, T(z'_0), \tilde{A}x'_1, \tilde{B}x'_2, S(z'_0)) \in \mathbb{R}^{2d}$$ 

such that $Mv' = 0$ but $\tilde{A}x'_1 + \tilde{B}x'_2 \neq 0$. We now know that $x'_1 = x'_2 = S(z'_0)$. Define 

$$A(x) = x - x'_1 + \tilde{A}x'_1, \quad B(x) = x - x'_2 + \tilde{B}x'_2.$$ 

Then $T(A, B, z_0) = z_0$, but 

$$(A + B)(S(z'_0)) = \tilde{A}x'_1 + \tilde{B}x'_2 \neq 0.$$ 

So $(T, S)$ fails to be a fixed-point encoding for the instance $A, B \in \mathcal{M}(\mathbb{R}^d)$, and we have a contradiction.

With the assertions proved, we proceed to complete the proof. Gaussian elimination on (2) gives us the equivalent system 

$$0 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \theta_1 & 0 & \theta_2 & 1 & 0 & 0 & 0 & 0 & 0 \\ \theta_3 + 1 & 0 & \theta_4 & 0 & \theta_5 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ x_1 \\ z_2 \\ x_2 \\ T_{z_0} \\ \tilde{A}x_1 \\ \tilde{B}x_2 \\ S_{z_0} \end{bmatrix}.$$ 

(3)

Because the system of linear equalities must imply $x_1 = x_2$ and because of where the zeros and nonzeros are placed, we have $\theta_3 = -1$ and $\theta_4 = -\theta_5 = \theta$ for some $\theta \neq 0$.

Let us further spell out this argument. The linear equality $x_1 = x_2$ can be expressed as 

$$0 = \begin{bmatrix} z_0 \\ z_1 \\ x_1 \\ z_2 \\ x_2 \\ T_{z_0} \\ \tilde{A}x_1 \\ \tilde{B}x_2 \\ S_{z_0} \end{bmatrix}.$$ 

(4)

9
By Lemma 5, the system of linear equalities (3) implies (4) if and only if we can linearly combine the rows of (3) to get (4). Row 7 of (3) cannot be used in the linear combination, as any nonzero contribution from row 7 will place a nonzero component in the 9th column. Row 6 of (3) also cannot be used in the linear combination, as any nonzero contribution from row 6 will place a nonzero component in the 8th column. Repeating this argument tells us that rows 7, 6, 5, 4, 2, and 1 cannot be used in the linear combination. Therefore, a scalar multiple of row 3 of (3) must equal (4), and this tells us \( \theta_3 = -1 \) and \( \theta_4 = -\theta_5 = \theta \) for some \( \theta \neq 0 \). We use a similar argument twice more in this proof, and once in the proof of Theorem 4.

Because the linear equality must imply \( x_1 = S z_0 \) and because of where the zeros and nonzeros are placed, \( \theta_6 = 0 \), \( \theta_7 = -1 + \eta \), and \( \theta_8 = -\eta \) for some \( \eta \in \mathbb{R} \). Plugging these in, we get

\[
0 = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\theta_1 & 0 & \theta_2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \theta & 0 & -\theta & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & \alpha & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & \beta & 0 \\
0 & 0 & -1 + \eta & 0 & -\eta & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
x_1 \\
z_2 \\
x_2 \\
Tz_0 \\
Ax_1 \\
Bx_2 \\
S z_0 \\
\end{bmatrix}.
\]

We perform Gaussian elimination again to get the equivalent system

\[
0 = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\theta_1 & 0 & \theta_2 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 + \theta_1 + \theta_2 & 0 & 0 & 0 & 0 & -1(1 + \theta_2)\alpha & \beta & 0 \\
-1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & \alpha & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & \beta & 0 \\
0 & 0 & -1 + \eta & 0 & -\eta & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
x_1 \\
z_2 \\
x_2 \\
Tz_0 \\
Ax_1 \\
Bx_2 \\
S z_0 \\
\end{bmatrix}.
\]

Because the linear equality must imply \( \tilde{A} x_1 + \tilde{B} z_2 = 0 \) and because of where the zeros and nonzeros are placed, \( \theta_1 = \beta/\alpha \) and \( \theta_2 = -1 - \beta/\alpha \). Finally, plugging in the parameters and expressing the splitting in functional form, we get the splitting of Theorem 4.

\[\square\]

### 2.4 Proof of Theorem 2

When \( \alpha = \beta \), the splitting \((T, S)\) of Theorem 1 reduces to the setup of DRS. It is well known that the iteration converges for all maximal monotone \( A \) and \( B \) if and only if \( \theta \in (0, 2) \) in this case.

Now assume \( \alpha \neq \beta \). We provide counter examples, single-valued maximal monotone operators \( A \) and \( B \) such that \( \{0\} = \text{zer}(A + B) \) and \( T^k z^0 \) diverges for any \( z^0 \neq 0 \). Note that the parameters \( \alpha \) and \( \beta \) are fixed and are provided by the splitting. Our counter examples rely on \( \alpha \) and \( \beta \).

For the moment, consider the case \( d = 2 \). Consider the problem

\[
\text{find } x \in \mathbb{R}^2 \quad 0 = (A + B)x,
\]

where

\[
A = \begin{bmatrix}
0 & \tan(\omega)/\alpha \\
-\tan(\omega)/\alpha & 0 \\
\end{bmatrix} \quad B = \begin{bmatrix}
0 & \tan(\omega)/\beta \\
\tan(\omega)/\beta & 0 \\
\end{bmatrix}
\]

and \( \alpha, \beta > 0 \), and \( \omega \in (0, \pi/2) \). We identify \( A \) and \( B \) as maximal monotone operators from \( \mathbb{R}^2 \to \mathbb{R}^2 \). Note that \( x^* = 0 \) is the unique solution.

With basic algebra, we can show that

\[
T z = \begin{bmatrix}
1 & \theta(\alpha - \beta) \cos(\omega) \sin(\omega) \\
-\theta(\alpha - \beta) \cos(\omega) \sin(\omega) & 1 \\
\end{bmatrix} z.
\]
With basic eigenvalue computation, we get
\[ |\lambda_1|^2 = |\lambda_2|^2 = 1 + \left(\theta(1 - \beta/\alpha) \cos(\omega) \sin(\omega)\right)^2 > 1, \]
where \(\lambda_1, \lambda_2\) are the eigenvalues of the matrix that defines \(T\). So if \(z^0 \neq 0\), the iteration \(z^{k+1} = Tz^k\) diverges in that \(\|z^k\| \to \infty\) and \(\|Sz^k\| \to \infty\).

When \(d > 2\), we arrive at the same conclusion with
\[
\begin{bmatrix}
A & 0 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix} \in \mathbb{R}^{d \times d},
\begin{bmatrix}
B & 0 & 0 & \cdots & 0 \\
0 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \ddots & \ddots & 0 \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix} \in \mathbb{R}^{d \times d},
\]
which is the same counter example embedded into \(d\) dimensions.

3 Impossibility of 3 operator resolvent-splitting without lifting

Define the problem class \((3\text{op-}\mathbb{R}^d)\) to be the collection of monotone inclusion problems of the form

\[
\text{find } x \in \mathbb{R}^d \quad 0 \in (A + B + C)x
\]

with \(A, B, C \in \mathcal{M}(\mathbb{R}^d)\). A pair of functions \((T, S)\) is a fixed-point encoding for the problem class \((3\text{op-}\mathbb{R}^d)\) if

\[
\exists z^* \in \mathbb{R}^d \text{ such that } \begin{bmatrix} T(A, B, C, z^*) \\ S(A, B, C, z^*) \end{bmatrix} = \begin{bmatrix} z^* \\ x^* \end{bmatrix} \iff 0 \in (A + B + C)(x^*). \]

We call

\[
T : \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathbb{R}^d \rightarrow \mathbb{R}^d
\]
the fixed-point mapping and

\[
S : \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathbb{R}^d \rightarrow \mathbb{R}^d,
\]
the solution mapping. The four key terms, resolvent-splitting, frugal, unconditional convergence, and no lifting, are defined analogously.

To define the notion of resolvent-splitting without lifting for the problem class \((3\text{op-}\mathbb{R}^d)\), we define the class of mappings \(\mathcal{G}\) similarly to how we defined \(\mathcal{F}\). More specifically,

\[
\mathcal{G}_0 = \{I\} \cup \{J_\alpha A \mid \alpha > 0\} \cup \{J_\alpha B \mid \alpha > 0\} \cup \{J_\alpha C \mid \alpha > 0\},
\]
\(G_1, G_2, \ldots\) are defined recursively in an analogous manner, and

\[
\mathcal{G} = \bigcup_{i=0}^{\infty} \mathcal{G}_i.
\]

Elements of \(\mathcal{G}\) map \(\mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathbb{R}^d\) to \(\mathbb{R}^d\). If \(R \in \mathcal{F}\) and \(A, B, C \in \mathcal{M}(\mathbb{R}^d)\), then \(R(A, B, C, \cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d\). If \((T, S)\) is a fixed-point encoding for the problem class \((3\text{op-}\mathbb{R}^d)\) and \(T, S \in \mathcal{G}\), then \((T, S)\) is a resolvent-splitting without lifting for the problem class \((3\text{op-}\mathbb{R}^d)\).

Frugality is defined analogously. We could make the definition formal with the notion of evaluation procedures and DAGs, but there is no need, since we only use the notion informally for the problem class \((3\text{op-}\mathbb{R}^d)\).

Unconditional convergence is also defined analogously. We say \((T, S)\) converges unconditionally for the problem class \((3\text{op-}\mathbb{R}^d)\) if

\[
ST^k z^0 \rightarrow x^*, \quad x^* \in \text{zer}(A + B + C)
\]
for any \(z^0 \in \mathbb{R}^d\) and \(A, B, C \in \mathcal{M}(\mathbb{R}^d)\), when \(\text{zer}(A + B + C)\) is not empty.
3.1 Impossibility result

If one could find a frugal, unconditionally convergent resolvent-splitting without lifting for \((3\text{-op-}\mathbb{R}^d)\), it would be a satisfying generalization of DRS to 3 operators. However, this is impossible. Even if we drop frugality and convergence as requirements, this is impossible.

**Theorem 4.** There is no resolvent-splitting without lifting for \((3\text{-op-}\mathbb{R}^d)\).

**Clarification.** Assume \(T(A, B, C, \cdot) : \mathbb{R}^d \to \mathbb{R}^d\) and \(S(A, B, C, \cdot) : \mathbb{R}^d \to \mathbb{R}^d\) are constructed with finitely many resolvents,

\[
J_{\alpha(1)A}, J_{\alpha(2)A}, \ldots, J_{\alpha(n_A)A} \\
J_{\beta(1)B}, J_{\beta(2)B}, \ldots, J_{\beta(n_B)B} \\
J_{\gamma(1)C}, J_{\gamma(2)C}, \ldots, J_{\gamma(n_C)C}
\]

where the parameters \(\alpha(i), \beta(j), \gamma(k)\) may be different. Theorem 4 states that \((T, S)\) will fail to be a fixed-point encoding.

**Clarification.** Another way to state Theorem 4 is to say that no element of the near-ring \(G\) is a fixed-point encoding for \((3\text{-op-}\mathbb{R}^d)\).

3.2 Proof of Theorem 4

Assume for contradiction that \((T, S)\) is a resolvent-splitting without lifting. Let \(n\) be the total number of resolvent evaluations required to compute \(T\) and \(S\). The specific value of \(n\) depends on how you count, i.e., whether you simplify things and whether some resolvent evaluations are counted redundantly. All that matters is that \(n\) is finite.

Since \(T, S \in G\), there is an evaluation procedure for \((T, S)\), and we can find a sequential ordering for the resolvent evaluations. We label the resolvents with this ordering \(J_1, J_2, \ldots, J_n\), where \(J_i\) is one of \(J_{\alpha A}, J_{\beta B}, \text{ or } J_{\gamma C}\) for some \(\alpha > 0, \beta > 0, \text{ or } \gamma > 0\) for each \(i = 1, \ldots, n\). We call \(z_i\) the point at which \(J_i\) is evaluated and \(x_i = J_i(z_i)\) for \(i = 1, \ldots, n\). In the process of evaluating \(Tz_0\) and \(Sz_0\), we get \(z_0, z_1, z_2, x_1, x_2, \ldots, z_n, x_n\). By nature of the ordering, \(z_i\) is defined as a linear combination of \(z_0, z_1, x_1, z_2, x_2, \ldots, z_{i-1}, x_{i-1}\) for each \(i = 1, \ldots, n\). Likewise, \(Tz_0\) can be expressed as a linear combination of \(z_0, z_1, x_1, z_2, x_2, \ldots, z_n, x_n\). Assume \(J_{\alpha A}, J_{\beta B}, \text{ and } J_{\gamma C}\) are all used least once in \(T\) or \(S\) with some \(\alpha > 0, \beta > 0, \text{ and } \gamma > 0\). Otherwise, if, for example, \(J_{\alpha A}\) is never used, we let \(z_{n+1} = 0\) and \(J_{n+1} = J_A\) to fix the issue.

Say \(J_{\alpha A}, J_{\beta B}, \text{ and } J_{\gamma C}\) are evaluated \(n_A, n_B, \text{ and } n_C\) times, respectively. So \(n_A + n_B + n_C = n\). Define the indices \(a(1), a(2), \ldots, a(n_A) \in \{1, 2, \ldots, n\}\) and the parameters \(\alpha(1), \alpha(2), \ldots, \alpha(n_A) > 0\) so that

\[
x_{a(t)} = J_{a(t)}(z_{a(t)}) = J_{\alpha A}(a(t), z_{a(t)}).
\]

In other words, \(x_{a(1)}, x_{a(2)}, \ldots, x_{a(n_A)}\) are the outputs of the resolvents of \(A\). (Define \(a(1), a(2), \ldots, a(n_A)\) so that they are distinct, i.e., \(x_{a(1)}, x_{a(2)}, \ldots, x_{a(n_A)}\) must cover all \(n_A\) outputs of the resolvents of \(A\).) Define the indices \(b(1), b(2), \ldots, b(n_B) \in \{1, 2, \ldots, n\}\) and \(c(1), c(2), \ldots, c(n_C) \in \{1, 2, \ldots, n\}\) and the parameters \(\beta(1), \beta(2), \ldots, \beta(n_B) > 0\) and \(\gamma(1), \gamma(2), \ldots, \gamma(n_C) > 0\) likewise.

We express the evaluation of \(Tz_0\) with the following system of linear and nonlinear equalities:

\[
0 = \begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
* & * & 1 & 0 & 0 & \cdots & 0 & 0 \\
* & * & * & * & \cdots & 1 & \cdots & 0 & 0 \\

dots & & & & & & & & \\
* & * & * & * & \cdots & * & \cdots & 1 & \cdots & 0 & 0 \\
* & * & * & * & \cdots & * & \cdots & * & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
z_2 \\
z_3 \\
z_4 \\
z_5 \\
z_6 \\
z_7 \\
z_8 \\
z_9 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
x_9 \\
x_{10} \\
x_{11} \\
x_{12} \\
x_{13} \\
x_{14} \\
x_{15} \\
x_{16} \\
x_{17} \\
x_{18} \\
x_{19} \\
x_{20}
\end{bmatrix}
\]

\[
x_1 = J_1(z_1), \quad x_2 = J_2(z_2), \ldots, \quad x_n = J_n(z_n),
\]
where the * denote unspecified scalar coefficients. (Each scalar in the matrix should be interpreted as a $d \times d$ block. We have seen this notation in the proof of Theorem 1.) Each linear equality except the last one defines $z_i$ for $i = 1, \ldots, n$. The last linear equality defines $T(z_0)$. With Gaussian elimination, we obtain the simpler equivalent system

$$
0 = \begin{bmatrix}
* 1 0 0 0 \cdots 0 0 0 \\
* 0 * 1 0 0 \cdots 0 0 0 \\
* 0 * 0 * 1 \cdots 0 0 0 \\
\vdots \\
* 0 * 0 * 0 \cdots 1 0 0 \\
* 0 * 0 * 0 \cdots 0 * 1
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
z_2 \\
z_3 \\
z_n \\
x_n \\
Tz_0
\end{bmatrix}
$$

$x_1 = J_1(z_1)$, $x_2 = J_2(z_2)$, \ldots, $x_n = J_n(z_n)$.

Now we add the fixed-point condition $z_0 = Tz_0$ and perform Gaussian elimination:

$$
0 = \begin{bmatrix}
* 1 0 0 0 \cdots 0 0 0 \\
* 0 * 1 0 0 \cdots 0 0 0 \\
* 0 * 0 * 1 \cdots 0 0 0 \\
\vdots \\
* 0 * 0 * 0 \cdots 1 0 0 \\
* 0 * 0 * 0 \cdots 0 * 0
\end{bmatrix}
\begin{bmatrix}
z_0 \\
z_1 \\
z_2 \\
z_3 \\
z_n \\
x_n \\
Tz_0
\end{bmatrix}
$$

$x_1 = J_1(z_1)$, $x_2 = J_2(z_2)$, \ldots, $x_n = J_n(z_n)$.

The **boldface symbols** denote where to pay attention in the linear system. Consider the system of linear equalities $Mv = 0$ of (5) combined with the linear equalities

$$
x_{a(1)} = x_{a(2)} = \cdots = x_{a(n_A)} \\
x_{b(1)} = x_{b(2)} = \cdots = x_{b(n_B)} \\
x_{c(1)} = x_{c(2)} = \cdots = x_{c(n_C)}.
$$

(6)

It is impossible to perform Gaussian elimination on these linear equalities to establish $x_{a(1)} = x_{b(1)} = x_{c(1)}$. Every row of $M$, except the last one, cannot be used in Gaussian elimination to prove a linear equality only involving the $x$’s, and the linear equalities of (6) do not help either. The last row of $M$ can establish $x_{a(1)} = x_{b(1)}$ or $x_{b(1)} = x_{c(1)}$ (or neither) but not both.

Let us further spell out this argument. By Lemma 3, the linear equalities of (5) and (6) imply $x_{a(1)} = x_{b(1)}$ and $x_{b(1)} = x_{c(1)}$ if and only if we can linearly combine the linear equalities of (5) and (6) to get $x_{a(1)} = x_{b(1)}$ and $x_{b(1)} = x_{c(1)}$. Note that $x_{a(1)} = x_{b(1)}$ and $x_{b(1)} = x_{c(1)}$ are linear equalities that do not involve $z_1, \ldots, z_n$ or $Tz_0$. All rows in (5) except the last two cannot be used in this linear combination, since they would introduce a nonzero corresponding to $z_i$ for some $i = 1, \ldots, n$. The second to last row of (5) cannot be used in this linear combination, since it would introduce a nonzero corresponding to $Tz_0$. This leaves us with only the last row of (5), which is only one row. Depending on what its unspecified coefficients are, the last row of (5) can establish $x_{a(1)} = x_{b(1)}$ or $x_{b(1)} = x_{c(1)}$, but not both. Adding the additional linear equalities (6) to the system does not help establish $x_{a(1)} = x_{b(1)}$ and $x_{b(1)} = x_{c(1)}$. (Adding (6) to the linear system of equalities is needed for constructing the counter example.)
Loosely speaking, the argument is that the fixed-point condition $Tz_0 = z_0$ must imply that the operators $A$, $B$, and $C$ are evaluated at the same point, but the fixed-point condition $Tz_0 = z_0$ does not have enough degrees of freedom to ensure this. DRS works, because it has one row to establish the one equality $x_a = x_b$, where $x_a$ and $x_b$ are the points where $A$ and $B$ are evaluated at. The splitting with 2-fold lifting presented in Section 4 works, because it has two rows (coming from $T_1(z_1, z_2) = z_1$ and $T_2(z_1, z_2) = z_2$) to establish the two equalities $x_a = x_b$ and $x_b = x_c$, where $x_a$, $x_b$, and $x_c$ are the points where $A$, $B$, and $C$ are evaluated at.

To summarize our progress, we have established that the linear equalities of $\mathcal{L}$, which includes the assumption that $z_0$ is a fixed point, fail to imply both linear equalities $x_{a(1)} = x_{b(1)}$ and $x_{b(1)} = x_{c(1)}$. Even if we add the linear equalities $\mathcal{L}$, the combined system fail to imply both linear equalities $x_{a(1)} = x_{b(1)}$ and $x_{b(1)} = x_{c(1)}$.

However, the fact that we cannot establish $x_{a(1)} = x_{b(1)} = x_{c(1)}$ with the linear equalities does not immediately imply that we cannot do so if we also use the nonlinear equalities. Therefore we construct a counter example, $A, B, C \in \mathcal{M}(\mathbb{R}^d)$ such that $(T, S)$ fails to be a fixed-point encoding.

We construct the counter example for the case $d = 1$. When $d > 1$, we can use the same 1 dimensional construction repeated for the $d$ coordinates. More specifically, if $A \in \mathcal{M}(\mathbb{R})$, then $\tilde{A}$ is defined with

$$\tilde{A}(x) = (A(x_1), A(x_2), \ldots, A(x_d)),$$

where $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$, satisfies $\tilde{A} \in \mathcal{M}(\mathbb{R}^d)$. This sort of construction based on the 1 dimensional counter example will provide a $d$ dimensional counter example.

Assume $d = 1$. By Lemma 3 there is a specific instance

$$v' = (z_0', z_1', x_1', z_2', x_2', \ldots, z_n', x_n', T(z_0')) \in \mathbb{R}^{2n}$$

that satisfies $Mv' = 0$ and the linear equalities of $\mathcal{L}$, but $x_{a(1)}' \neq x_{b(1)}'$ or $x_{b(1)}' \neq x_{c(1)}'$ or both. Without loss of generality, say $x_{b(1)}' \neq x_{c(1)}'$.

Define $A$ such that

$$J_{\alpha(i)}(z_{a(1)}') = x_{a(1)}'$$

for all $i = 1, \ldots, n_A$. In particular, we achieve this by defining

$$A(x_{a(1)}') = \left[ \min_{i=1, \ldots, n_A} \left( z_{a(i)}' - x_{a(i)}' \right) / \alpha(i), \max_{i=1, \ldots, n_A} \left( z_{a(i)}' - x_{a(i)}' \right) / \alpha(i) \right].$$

For the moment, leave $A(x)$ for $x \neq x_{a(1)}'$ unspecified. Define $B(x_{b(1)}')$ and $C(x_{c(1)}')$ likewise. By construction, $z_0' = T(A, B, C, z_0')$, even though $A, B,$ and $C$ are not yet fully specified. Write $x' = S(z_0')$. We have $x' \neq x_{b(1)}'$ or $x' \neq x_{c(1)}'$ since $x_{b(1)}' \neq x_{c(1)}'$. Without loss of generality, let $x' \neq x_{c(1)}'$.

Now we define

$$A(x) = \begin{cases} (x - x_{a(1)}) + \min \{ A(x_{a(1)}) \} & \text{for } x < x_{a(1)}' \\ (x - x_{a(1)}) + \max \{ A(x_{a(1)}) \} & \text{for } x > x_{a(1)}' \end{cases}$$

and

$$B(x) = \begin{cases} (x - x_{b(1)}) + \min \{ A(x_{b(1)}) \} & \text{for } x < x_{b(1)}' \\ (x - x_{b(1)}) + \max \{ A(x_{b(1)}) \} & \text{for } x > x_{b(1)}' \end{cases}.$$

(This makes $A$ and $B$ maximal monotone.) By construction, $(A + B)(x')$ is a bounded subset of $\mathbb{R}$, and $C(x')$ is unspecified. Depending on whether $x' < x_{c(1)}'$ or $x' > x_{c(1)}'$, we can make $C(x')$ an arbitrarily small or large value, respectively (and still have $C$ be monotone). In either case, we make $C(x')$ single-valued and so small or so large that $0 \notin (A + B + C)(x')$. We extend the definition of $C$ to all of $\mathbb{R}$ to make it maximal monotone.

So we have maximal monotone operators $A$, $B$, and $C$, such that $z_0' = T(A, B, C, z_0')$ but the $x' = S(z_0')$ does not satisfy $0 \in (A + B + C)x'$. This contradicts the assumption that $(T, S)$ is a fixed-point encoding.

\section{Attainment of 3 operator resolvent-splitting with minimal lifting}

Loosely speaking, we say a resolvent-splitting $(T, S)$ for the problem class $\mathcal{3op-\mathbb{R}^d}$ uses $\ell$-fold lifting if

$$T(A, B, C, \cdot) : \mathbb{R}^{\ell d} \to \mathbb{R}^{\ell d}, \quad S(A, B, C, \cdot) : \mathbb{R}^{\ell d} \to \mathbb{R}^d.$$ 

Note that 1-fold lifting corresponds to no lifting. Theorem 4 states a resolvent-splitting for $\mathcal{3op-\mathbb{R}^d}$ requires lifting. Then how much? The answer is 2-fold lifting.
4.1 Definitions

We formally define the notion of resolvent-splitting with \( \ell \)-fold lifting. Although the following definitions are not directly used in Theorem 5, they are useful for making the qualitative discussion concrete and for making the definition of minimal lifting formal.

Use the notation
\[
 z = (z_1, \ldots, z_\ell) \in \mathbb{R}^{\ell d}.
\]

Define \( E_i : \mathcal{M} \times \mathcal{M} \times \mathcal{M} \times \mathbb{R}^{\ell d} \to \mathbb{R}^d \) to be the block selection function,
\[
 E_i(A, B, C, z) = z_i,
\]
for \( i = 1, \ldots, \ell \). Define \( L_0 = \{E_1, \ldots, E_\ell\} \).

Recursively define
\[
 L_{i+1} = \{J_{\alpha A} \circ F | F \in L_i, \alpha > 0\} \cup \{J_{\alpha B} \circ F | F \in L_i, \alpha > 0\} \cup \{F + G | F, G \in L_i\} \cup \{\beta F | F \in L_i, \beta \in \mathbb{R}\},
\]
and let
\[
 L = \bigcup_{i=0}^{\infty} L_i.
\]

To clarify, \( L \) contains mappings from \( \mathcal{M} \times \mathcal{M} \times \mathcal{M} \times \mathbb{R}^{\ell d} \) to \( \mathbb{R}^d \).

We say \((T, S)\) is a resolvent-splitting with \( \ell \)-fold lifting for the problem class \(3\text{op-}\mathbb{R}^d\) if \((T, S)\) is a fixed-point encoding for the problem class \(3\text{op-}\mathbb{R}^d\),
\[
 T : \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathbb{R}^{\ell d} \to \mathbb{R}^{\ell d},
\]
\[
 S : \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathcal{M}(\mathbb{R}^d) \times \mathbb{R}^{\ell d} \to \mathbb{R}^d,
\]
\[
 T = (T_1, \ldots, T_\ell) \text{ with } T_1, \ldots, T_\ell \in L, \text{ and } S \in L. 
\]
Frugality is defined analogously, and unconditional convergence was defined in Section 3.

A standard trick to solve \(3\text{op-}\mathbb{R}^d\) is to “copy” variables to form an enlarged problem
\[
 \text{find } x_1, x_2, x_3 \in \mathbb{R}^d, \quad 0 \in \begin{bmatrix} Ax_1 \\ Bx_2 \\ Cx_3 \end{bmatrix} + N((x_1, x_2, x_3) | x_1 = x_2 = x_3) (x_1, x_2, x_3),
\]
where \( N_K \) is the normal cone operator with respect to the set \( K \), and apply DRS to get
\[
 \bar{z} = (1/3)(z_A + z_B + z_C) \\
 T_A(z) = z_A + J_{\alpha B}(2\bar{z} - z_A) - \bar{z} \\
 T_B(z) = z_B + J_{\alpha C}(2\bar{z} - z_B) - \bar{z} \\
 T_C(z) = z_C + J_{\alpha A}(2\bar{z} - z_C) - \bar{z} \\
 S(z) = J_{\alpha A}(2\bar{z} - z_A),
\]
where \( \alpha > 0 \) and \( z = (z_A, z_B, z_C) \). This method is also an instance of Spingarn’s method of partial inverse [37] or parallel proximal algorithm (PPXA) [14, 15]. This frugal, unconditionally convergent resolvent-splitting uses 3-fold lifting, since \( T = (T_A, T_B, T_C) : \mathbb{R}^{3d} \to \mathbb{R}^{3d} \).

So constructing a resolvent-splitting for \(3\text{op-}\mathbb{R}^d\) is impossible with 1-fold lifting, but it is possible with 3-fold lifting. It turns out that 2-fold lifting is sufficient, and we therefore call 2-fold lifting the minimal lifting for \(3\text{op-}\mathbb{R}^d\).
4.2 Attainment result

Theorem 5. The pair \((T, S)\), where \(T : \mathbb{R}^{2d} \to \mathbb{R}^{2d}\) and \(S : \mathbb{R}^{2d} \to \mathbb{R}^d\) are defined as

\[
\begin{align*}
x_1 &= J_{\alpha A}(z_1) \\
x_2 &= J_{\alpha B}(x_1 + z_2) \\
x_3 &= J_{\alpha C}(x_1 - z_1 + x_2 - z_2) \\
T_1(z) &= z_1 + \theta(x_3 - x_1) \\
T_2(z) &= z_2 + \theta(x_3 - x_2) \\
S(z) &= (1/3)(x_1 + x_2 + x_3)
\end{align*}
\]

with \(z = (z_1, z_2)\) and \(T = (T_1, T_2)\), is a fixed-point encoding, and \((T, S)\) converges unconditionally for \(\theta \in (0, 1)\) and \(\alpha > 0\).

Therefore, \((T, S)\) for any \(\theta \in (0, 1)\) is a frugal, unconditionally convergent, resolvent-splitting with minimal lifting for \(3\text{op-}\mathbb{R}^d\). When \(B = 0\), the splitting of Theorem 5 reduces to DRS. In this sense, this splitting is a direct generalization of DRS with minimal lifting.

Remark. To the best of the author’s knowledge, the splitting of Theorem 5 cannot be reduced to an instance of a known splitting method. This is why Theorem 5 is proved from first principles.

4.3 Proof of Theorem 5

Throughout the proof, write \(z = (z_1, z_2)\). Without loss of generality, assume \(\alpha = 1\). We first show that \((T, S)\) is a fixed-point encoding.

Assume \(z\) is a fixed point of \(T\). Write

\[
\begin{align*}
a &= z_1 - x_1 \\
b &= x_1 + z_2 - x_2 \\
c &= x_1 - z_1 + x_2 - z_2 - x_3.
\end{align*}
\]

Add the three and use \(x_1 = x_2 = x_3\) to get

\[a + b + c = 0.\]

Since \(a \in Ax_1, b \in Bx_2,\) and \(c \in Cx_3,\) by the definitions of \(x_1, x_2,\) and \(x_3,\) this proves \(x_1 = x_2 = x_3\) is a solution to \(3\text{op-}\mathbb{R}^d\).

Now assume \(x^*\) is a solution to \(3\text{op-}\mathbb{R}^d\), and let \(a \in Ax^*, b \in Bx^*,\) and \(c \in Cx^*\) so that \(a + b + c = 0\). We then define

\[z^* = (a + x^*, b)\]

It is straightforward to verify that \(T(z^*) = z^*\) and \(S(z) = x^*\).

Next we show that \((T, S)\) converges unconditionally for \(\theta \in (0, 1)\). We show this by showing \(T\) is nonexpansive for \(\theta = 1\).

Let \(\theta = 1\). Define \(M : \mathbb{R}^{2d} \to \mathbb{R}^{2d}\) with

\[
M(x_1, x_2) = \begin{bmatrix} x_1 + x_2 \\ x_1 + x_2 \end{bmatrix},
\]

which is the linear operator corresponding to the matrix

\[
\begin{bmatrix} I & I \\ I & I \end{bmatrix} \in \mathbb{R}^{2d \times 2d}.
\]

Define \(U\) as

\[
U(z) = \begin{bmatrix} J_A(z_1) - z_1 \\ J_B(z_2 + x_1) - z_2 \end{bmatrix} \in \mathbb{R}^{2d},
\]

16
and we can write
\[
T = -U + \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U.
\]

Define
\[
N = \begin{bmatrix} -I & I \\ I & -I \end{bmatrix} \in \mathbb{R}^{2d \times 2d}.
\]

Then
\[
\|T(y) - T(z)\|^2 \\
= \|U(y) - U(z)\|^2 + \left\| \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(y) - \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(z) \right\|^2 \\
- 2 \left\langle U(y) - U(z), \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(y) - \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(z) \right\rangle \\
\leq \|U(y) - U(z)\|^2 \\
+ \left\langle M \circ (U(y) - U(z)), \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(y) - \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(z) \right\rangle \\
- 2 \left\langle U(y) - U(z), \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(y) - \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(z) \right\rangle \\
\leq \|U(y) - U(z)\|^2 \\
+ (U(y) - U(z))^T N \left( \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(y) - \begin{bmatrix} J_C \\ J_C \end{bmatrix} \circ M \circ U(z) \right) \\
\leq \|U(y) - U(z)\|^2.
\]

The inequality follows from firm-nonexpansiveness of \(J_C\). Next we have
\[
\|U(y) - U(z)\|^2 \\
= \|y - z\|^2 + \left\| \begin{bmatrix} J_A(y_1) - J_A(z_1) \\ J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1)) \end{bmatrix} \right\|^2 \\
- 2 \left\langle y - z, \begin{bmatrix} J_A(y_1) - J_A(z_1) \\ J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1)) \end{bmatrix} \right\rangle \\
\leq \|y - z\|^2 \\
+ \left\| \begin{bmatrix} J_A(y_1) - J_A(z_1) \\ J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1)) \end{bmatrix} \right\|^2 \\
- 2 \left\langle y - z, \begin{bmatrix} J_A(y_1) - J_A(z_1) \\ J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1)) \end{bmatrix} \right\rangle \\
\leq \|y - z\|^2 - \|J_A(y_1) - J_A(z_1)\|^2 - \|J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1))\|^2 \\
+ 2(J_A(y_1) - J_A(z_1), J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1))) \\
= \|y - z\|^2 - \|J_A(y_1) - J_A(z_1) + J_B(y_2 + J_A(y_1)) - J_B(z_2 + J_A(z_1))\|^2 \\
\leq \|y - z\|^2.
\]

The first two inequalities follow from firm-nonexpansiveness of \(J_A\) and \(J_B\). \(\square\)

### 4.4 Numerical examples

Whether the splitting of Theorem 5 is fast or efficient is somewhat besides the point, as the purpose of Theorem 5 is to establish attainment of minimal lifting. Nevertheless, we run 3 computational experiments to establish that the splitting of Theorem 5 is, at least in some cases, competitive with or even better than the existing approach of copying variables and applying DRS.
Signal denoising with outliers. Consider the problem

$$\text{minimize}_{x \in \mathbb{R}^d} \quad \|x_S - a\|_1 + \lambda\|Ux - b\|_1$$
subject to \quad x \geq 0,$$

where \( S \subseteq \{1, 2, \ldots, d\} \), \( a \in \mathbb{R}^{|S|} \), \( b \in \mathbb{R}^d \), and \( U \in \mathbb{R}^{d \times d} \) is a unitary matrix representing a wavelet transform.

The statistical interpretation is that we noisily observe \( x \) on a subset \( S \) of its indices, noisily observe \( x \) in the wavelet domain, and have a priori knowledge that \( x \) is nonnegative. The \( \ell^1 \)-norm is used for robustness against outliers. We reformulate this problem as

$$\text{minimize}_{x \in \mathbb{R}^d} \quad \|x_S - a\|_1 + \lambda\|Ux - b\|_1 + \delta_{\mathbb{R}^d}(x)$$
and apply the splittings of (7) and of Theorem 5 with \( A = \partial f \), \( B = \partial g \), and \( C = \partial h \). Because \( U \) is unitary, \( J_{\alpha \partial g} \) has a closed-form formula. For the experiments, we used synthetic data with \( d = 2^{20} \) and \( |S| \approx d/5 \). The code for data generation and optimization is provided in the author’s website for scientific reproducibility.

Figure 3 shows the results. We can see that the splitting of Theorem 5, which uses 2-fold lifting, is competitive with the splitting of (7). For all methods, the step size parameters were roughly tuned for best performance. We do not plot distance to solution, since solution does not seem to be unique.

Portfolio optimization. Consider the Markowitz portfolio optimization [7] problem

$$\text{minimize}_{x \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^n (a_i^T x - b)^2$$
subject to \quad \mu^T x \geq b,$$

where \( d \) is the number of assets, \( a_1, \ldots, a_n \in \mathbb{R}^d \) are \( n \) realizations of the returns on the assets, \( \Delta = \{ x \in \mathbb{R}^d \mid x_1, \ldots, x_d \geq 0, x_1 + \cdots + x_d = 1 \} \) is the standard simplex for portfolios with no short positions, \( \mu \in \mathbb{R}^d \) is the (estimated) average return of the assets, and \( b \in \mathbb{R} \) is the desired expected return. We reformulate this problem as

$$\text{minimize}_{x \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^n (a_i^T x - b)^2 + \delta_{\mathbb{R}^d}(x)$$
and apply the splittings of (7) and of Theorem 5 with \( A = \nabla f \), \( B = \partial g \), and \( C = \partial h \). We also run DYS [17] and Condat–Vũ [16, 41], for which direct evaluations of \( \nabla f \) were used instead of \( J_{\alpha \nabla f} \). To compute the projection onto the simplex, we use the algorithm and code of [11]. For the experiments, we used synthetic data with \( n = 30000 \).
Figure 4: |Objective value suboptimality|, |f(x^{k+1}) - f(x^k)|/f(x^k), and distance to solution vs. iterations for the portfolio optimization problem. We take the absolute value in the first plot, because the slightly infeasible iterates produce objective values lower than the optimal value. The rough cost per iteration is 0.025s for CV and DYS and 0.15s for the splitting of Theorem 5 and the splitting of 7.
Figure 5: Objective value, $|f(x^{k+1}) - f(x^k)|/f(x^k)$, and distance to solution vs. iterations for the fused lasso problem. Theorem 5 proves convergence for $\theta \in (0, 1)$. Indeed, the splitting diverges for $\theta = 2$.

and $d = 10000$, which make the data approximately 2GB in size. The code for data generation and optimization is provided in the author’s website for scientific reproducibility.

Figure 4 shows the results. The splitting of Theorem 5, which uses 2-fold lifting, is competitive with and even better than the splitting using 3-fold lifting of (7). However, DYS and Condat–Vũ are faster than the splittings that only use resolvents. Run-time measurements of the splitting of Theorem 5 and the splitting of (7) exclude the time it took to compute a Cholesky factorization, about 35.9s. An Intel Core i7-2600 CPU operating at 3.40GHz was used for the experiments. For all methods, the step size parameters were roughly tuned for best performance.

**Fused lasso.** Consider the fused lasso problem

$$\minimize_{x \in \mathbb{R}^d} \left( \frac{\lambda}{2n} \right) \|Ax - b\|^2 + \sum_{i=1}^{d-1} |x_{i+1} - x_i|$$

where $A \in \mathbb{R}^{n \times d}$, $b \in \mathbb{R}^n$, and $\lambda > 0$. For simplicity, assume $d$ is odd. We reformulate this problem as

$$\minimize_{x \in \mathbb{R}^d} \left( \frac{\lambda}{2n} \right) \|Ax - b\|^2 + \sum_{i=1}^{d-1} |x_{i+1} - x_i| + \sum_{i=2}^{d-1} |x_{i+1} - x_i|$$

and apply the splittings of (7) and of Theorem 5 with $A = \nabla f$, $B = \partial g$, and $C = \partial h$. We also run DYS [17] and Condat–Vũ [16, 41], for which direct evaluations of $\nabla f$ were used instead of $J_\alpha \nabla f$. For the experiments, we used synthetic data with $n = 3000$, $d = 1001$, and $\lambda = 10$. The code for data generation and optimization is provided in the author’s website for scientific reproducibility.

Figure 5 shows the results. The splitting of Theorem 5, which uses 2-fold lifting, is competitive with and even better than the splitting using 3-fold lifting of (7). For this setup, the splitting of Theorem 5 is also competitive with
DYS and Condat–Vũ. Run-time measurements of the splitting of Theorem 5 and the splitting of (7) exclude the time it took to compute a Cholesky factorization, about 0.13s. An Intel Core i7-2600 CPU operating at 3.40GHz was used for the experiments. For all methods, the step size parameters were roughly tuned for best performance.

5 Conclusion

This work establishes that DRS is the unique frugal, unconditionally convergent resolvent-splitting without lifting for the 2 operator problem and that there is no resolvent-splitting without lifting for the 3 operator problem. Furthermore, this work presents a novel, frugal, unconditionally convergent resolvent-splitting for the 3 operator problem that directly generalizes DRS. This splitting proves that 2-fold lifting is the minimal lifting necessary for the 3 operator problem. In other words, the presented splitting is optimal in terms of frugality and lifting.

The potential for future work based on the ideas presented in this work is large. Analyzing and establishing uniqueness or optimality of other splittings is one direction of future work. Characterizing all splittings of a given setup is another. In particular, there is no reason to believe the splitting of Theorem 5 is unique, so characterizing all frugal, unconditionally convergent resolvent-splittings for the 3 operator problem would be interesting.

Acknowledgements

I would like to thank Wotao Yin for helpful comments and suggestions. I would also like to thank the anonymous associate editor and referees whose comments improved the paper significantly. In particular, the signal denoising numerical example was suggested by one of the anonymous reviewers. This work is supported in part by NSF grant DMS-1720237 and ONR grant N000141712162.

References

[1] Banert, S.: A relaxed forward-backward splitting algorithm for inclusions of sums of monotone operators. Master’s thesis, Technische Universität Chemnitz (2012)

[2] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces, 2nd edn. Springer New York (2017)

[3] Brezis, H., Lions, P.L.: Produits infinis de resolvantes. Isr. J. Math. 29(4), 329–345 (1978)

[4] Briceño-Arias, L.M.: Forward-Douglas–Rachford splitting and forward-partial inverse method for solving monotone inclusions. Optimization 64(5), 1239–1261 (2015)

[5] Briceño-Arias, L.M., Combettes, P.L.: A monotone+skew splitting model for composite monotone inclusions in duality. SIAM J. Optim. 21(4), 1230–1250 (2011)

[6] Briceño-Arias, L.M., Davis, D.: Forward-backward-half forward algorithm with non self-adjoint linear operators for solving monotone inclusions. SIAM Journal on Optimization (2018)

[7] Brodie, J., Daubechies, I., De Mol, C., Giannone, D., Loris, I.: Sparse and stable Markowitz portfolios. Proc. Natl. Acad. Sci. USA 106(30), 12267–12272 (2009)

[8] Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. 40(1), 120–145 (2011)

[9] Chen, P., Huang, J., Zhang, X.: A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration. Inverse Problems 29(2), 025011 (2013)

[10] Chen, P., Huang, J., Zhang, X.: A primal-dual fixed point algorithm for minimization of the sum of three convex separable functions. Fixed Point Theory and Applications (2016)

[11] Chen, Y., Ye, X.: Projection onto a simplex. arXiv (2011)
[12] Combettes, P.L., Condat, L., Pesquet, J.C., Vũ, B.C.: A forward-backward view of some primal-dual optimization methods in image recovery. IEEE Int. Conf. Image Process. (2014)

[13] Combettes, P.L., Eckstein, J.: Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions. Mathematical Programming 168(1), 645–672 (2018)

[14] Combettes, P.L., Pesquet, J.C.: A proximal decomposition method for solving convex variational inverse problems. Inverse Problems 24(6) (2008)

[15] Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, H. Wolkowicz (eds.) Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 185–212. Springer (2011)

[16] Condat, L.: A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. J. Optim. Theory Appl. 158(2), 460–479 (2013)

[17] Davis, D., Yin, W.: A three-operator splitting scheme and its optimization applications. Set-Valued Var. Anal. 25(4), 829–858 (2017)

[18] Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Trans. Amer. Math. Soc. 82, 421–439 (1956)

[19] Drori, Y., Sabach, S., Teboulle, M.: A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. Oper. Res. Lett. 43(2), 209–214 (2015)

[20] Eckstein, J.: A simplified form of block-iterative operator splitting and an asynchronous algorithm resembling the multi-block alternating direction method of multipliers. Journal of Optimization Theory and Applications 173(1), 155–182 (2017)

[21] Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. SIAM J. Imaging Sci. 3(4), 1015–1046 (2010)

[22] Johnstone, P.R., Eckstein, J.: Projective splitting with forward steps: Asynchronous and block-iterative operator splitting. arXiv (2018)

[23] Latafat, P., Patrinos, P.: Asymmetric forward–backward–adjoint splitting for solving monotone inclusions involving three operators. Comput. Optim. Appl. 68(1), 57–93 (2017)

[24] Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. 16(6), 964–979 (1979)

[25] Loris, I., Verhoyen, C.: On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. Inverse Problems 27(12), 125007 (2011)

[26] Malitsky, Y., Tam, M.K.: A forward-backward splitting method for monotone inclusions without cocoercivity. arXiv (2018)

[27] Martinet, B.: Régularisation d’inéquations variationnelles par approximations successives. Rev. Fr. d’Inform. Rech. Oper. Sér. Rouge 4(3), 154–158 (1970)

[28] Martinet, B.: Determination approchée d’un point fixe d’une application pseudo-contractante. C. R. l’Acad. Sci. Sér. A 274, 163–165 (1972)

[29] Passty, G.B.: Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. J. Math. Anal. Appl. 72(2), 383–390 (1979)

[30] Peaceman, D.W., Rachford, H.H.: The numerical solution of parabolic and elliptic differential equations. J. Soc. Ind. Appl. Math. 3(1), 28–41 (1955)

[31] Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the mumford-shah functional. IEEE Int. Conf. Comput. Vis. (2009)
[32] Raguet, H.: A note on the forward-Douglas–Rachford splitting for monotone inclusion and convex optimization. Optimization Letters (2018)

[33] Raguet, H., Fadili, J., Peyré, G.: A generalized forward-backward splitting. SIAM Journal on Imaging Sciences 6(3), 1199–1226 (2013)

[34] Rapaport, F., Barillot, E., Vert, J.P.: Classification of arrayCGH data using fused SVM. Bioinformatics 24(13), i375–i382 (2008)

[35] Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. 14(5), 877–898 (1976)

[36] Ryu, E.K., Boyd, S.: Primer on monotone operator methods. Appl. Comput. Math. 15, 3–43 (2016)

[37] Spingarn, J.E.: Applications of the method of partial inverses to convex programming: Decomposition. Mathematical Programming 32(2), 199–223 (1985)

[38] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. J. R. Stat. Soc. Ser. B. Stat. Methodol. 67(1), 91–108 (2005)

[39] Tibshirani, R., Wang, P.: Spatial smoothing and hot spot detection for CGH data using the fused lasso. Biostatistics 9(1), 18–29 (2008)

[40] Tseng, P.: A modified forward-backward splitting method for maximal monotone mappings. SIAM J. Control Optim. 38(2), 431–446 (2000)

[41] Vû, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. Adv. Comput. Math. 38(3), 667–681 (2013)

[42] Yan, M.: A new primal–dual algorithm for minimizing the sum of three functions with a linear operator. Journal of Scientific Computing 76(3), 1698–1717 (2018)

[43] Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report 08-34 (2008)