Douglas–Rachford Feasibility Methods for Matrix Compl. . .

F.J. Aragón Artacho · J.M. Borwein · M.K. Tam

School of Mathematical and Physical Sciences
University of Newcastle, Australia

CARMA/OCANA Seminar
23rd October, 2013
(Revised 17/10/13)

Many successful non-convex applications of the Douglas–Rachford ('splitting') method can be considered as matrix completion problems. In this talk we discuss recent successful applications of the Douglas–Rachford algorithm to a variety of (real) matrix reconstruction problems, both convex and non-convex.

In particular we shall consider matrix completion in the context of:

1. Positive semi-definite matrices.
2. Stochastic matrices.
3. Euclidean distance matrices arising in protein reconstruction.
4. Hadamard matrices together with their specializations.
5. Nonograms – a Japanese number painting game.
6. Sudoku – a Japanese number game.

The framework is flexible, and there are many other actual and potential applications!

Those Involved

Fran Aragón Jon Borwein

Matt Tam

Hilbert Stadt Streets

- 'A mathematical theory is not to be considered complete until you have it so clear that you can explain it to the first man whom you meet on the street.' – David Hilbert

Fran is now working in Luxembourg

Consider the Hilbert space $\mathbb{R}^{m\times n}$ equipped with inner product and induced (Frobenius) norm

$$\langle A, B\rangle := \operatorname{tr}(A^T B), \quad \|A\|_F := \sqrt{\operatorname{tr}(A^T A)} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2}.$$

- A partial matrix is an $m \times n$ array for which only entries in certain locations are known.
- A completion of the partial matrix $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, is a matrix $B = (b_{ij}) \in \mathbb{R}^{m\times n}$ such that if a_{ij} is specified then $b_{ij} = a_{ij}$.

The problem of matrix completion is the following:

Given a partial matrix, A, find a completion having certain properties of interest.

It is natural to formulate the problem of matrix completion as a feasibility problem.

$$\text{Find } X \in \bigcap_{i=1}^{N} C_i \subseteq \mathbb{R}^{m\times n}.$$

Let A be the partial matrix to be completed. We (mostly) take

- C_1 to be the set of all completions of A,
- $C_2, \dots, C_N$ such that their intersection has the properties of interest.

Throughout, let Ω denote the set of indices for which the ijth entry of A is known. Thus

$$C_1 := \{X \in \mathbb{R}^{m\times n} | X_{ij} = A_{ij} \text{ for all } (i,j) \in \Omega\}.$$

A Variational Toolkit

Let $S \subseteq \mathbb{R}^{m\times n}$. The (nearest point) projection onto S is the (set-valued) mapping,

$$P_S x := \operatorname*{argmin}_{s\in S} \|s - x\|.$$

The reflection w.r.t. S is the (set-valued) mapping,

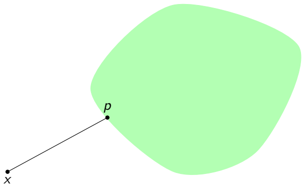
$$R_S := 2P_S - I.$$

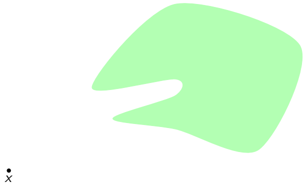
Let $S \subseteq \mathbb{R}^{m \times n}$. The (nearest point) projection onto S is the (set-valued) mapping,

$$P_S x := \operatorname*{argmin}_{s \in S} \|s - x\|.$$

The reflection w.r.t. S is the (set-valued) mapping,

$$R_S := 2P_S - I.$$





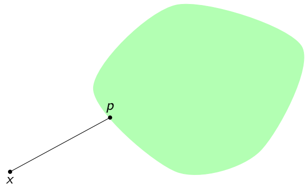
A Variational Toolkit

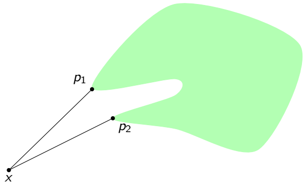
Let $S \subseteq \mathbb{R}^{m\times n}$. The (nearest point) projection onto S is the (set-valued) mapping,

$$P_S x := \operatorname*{argmin}_{s\in S} \|s - x\|.$$

The reflection w.r.t. S is the (set-valued) mapping,

$$R_S := 2P_S - I.$$



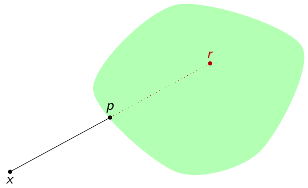


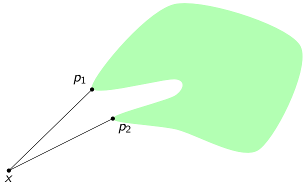
Let $S \subseteq \mathbb{R}^{m\times n}$. The (nearest point) projection onto S is the (set-valued) mapping,

$$P_S x := \operatorname*{argmin}_{s\in S} \|s - x\|.$$

The reflection w.r.t. S is the (set-valued) mapping,

$$R_S := 2P_S - I.$$



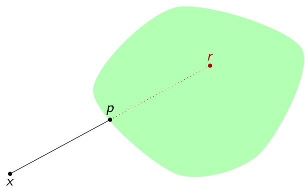


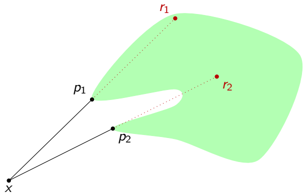
Let $S \subseteq \mathbb{R}^{m\times n}$. The (nearest point) projection onto S is the (set-valued) mapping,

$$P_S x := \operatorname*{argmin}_{s\in S} \|s - x\|.$$

The reflection w.r.t. S is the (set-valued) mapping,

$$R_S := 2P_S - I.$$





Theorem (Douglas–Rachford in finite dimensions)

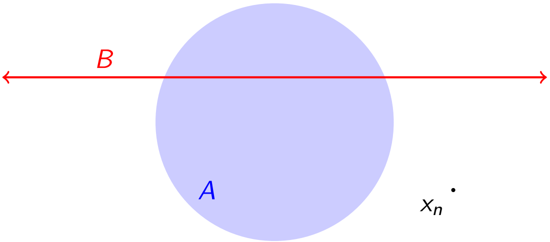
Suppose $A, B \subseteq \mathbb{R}^{m\times n}$ are closed and convex. For any $x_0 \in \mathbb{R}^{m\times n}$ define

$$x_{n+1} := T_{A,B}x_n \text{ where } T_{A,B} := \frac{I + R_B R_A}{2}.$$

Then if:

(a) $A \cap B \neq \emptyset$, (x_n) converges to a point x such that $P_A x \in A \cap B$.

(b) $A \cap B = \emptyset$, $\|x_n\| \to +\infty$.



$A := \{x \in \mathbb{R}^{m\times n} : \|x\| \leq 1\}$, $B := \{x \in \mathbb{R}^{m\times n} : \langle a, x\rangle = b\}$. (Phase retrieval)

Theorem (Douglas–Rachford in finite dimensions)

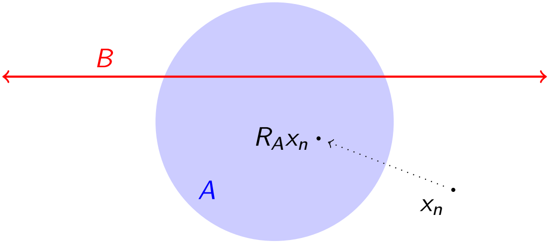
Suppose $A, B \subseteq \mathbb{R}^{m\times n}$ are closed and convex. For any $x_0 \in \mathbb{R}^{m\times n}$ define

$$x_{n+1} := T_{A,B}x_n \text{ where } T_{A,B} := \frac{I + R_B R_A}{2}.$$

Then if:

(a) $A \cap B \neq \emptyset$, (x_n) converges to a point x such that $P_A x \in A \cap B$.

(b) $A \cap B = \emptyset$, $\|x_n\| \to +\infty$.



$A := \{x \in \mathbb{R}^{m\times n} : \|x\| \leq 1\}$, $B := \{x \in \mathbb{R}^{m\times n} : \langle a, x\rangle = b\}$. (Phase retrieval)

The Douglas–Rachford Algorithm (1956–1979–)

Theorem (Douglas–Rachford in finite dimensions)

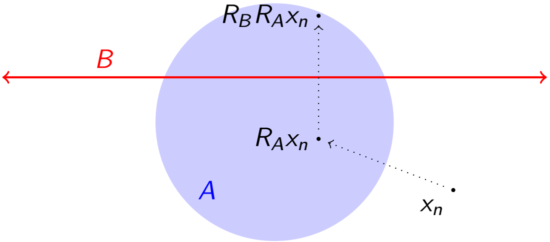
Suppose $A, B \subseteq \mathbb{R}^{m\times n}$ are closed and convex. For any $x_0 \in \mathbb{R}^{m\times n}$ define

$$x_{n+1} := T_{A,B} x_n \text{ where } T_{A,B} := \frac{I + R_B R_A}{2}.$$

Then if:

(a) $A \cap B \neq \emptyset$, (x_n) converges to a point x such that $P_A x \in A \cap B$.

(b) $A \cap B = \emptyset$, $\|x_n\| \to +\infty$.



$A := \{x \in \mathbb{R}^{m\times n} : \|x\| \leq 1\}$, $B := \{x \in \mathbb{R}^{m\times n} : \langle a, x\rangle = b\}$. (Phase retrieval)

Theorem (Douglas–Rachford in finite dimensions)

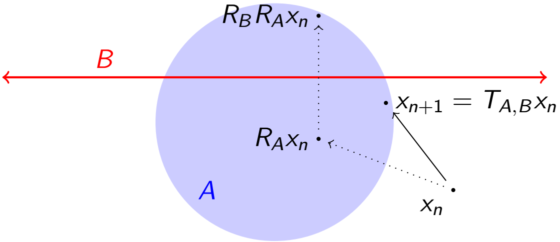
Suppose $A, B \subseteq \mathbb{R}^{m\times n}$ are closed and convex. For any $x_0 \in \mathbb{R}^{m\times n}$ define

$$x_{n+1} := T_{A,B}x_n \text{ where } T_{A,B} := \frac{I + R_B R_A}{2}.$$

Then if:

(a) $A \cap B \neq \emptyset$, (x_n) converges to a point x such that $P_A x \in A \cap B$.

(b) $A \cap B = \emptyset$, $\|x_n\| \to +\infty$.



$A := \{x \in \mathbb{R}^{m\times n} : \|x\| \leq 1\}$, $\quad B := \{x \in \mathbb{R}^{m\times n} : \langle a, x\rangle = b\}$. (Phase retrieval)

Theorem (Douglas–Rachford in finite dimensions)

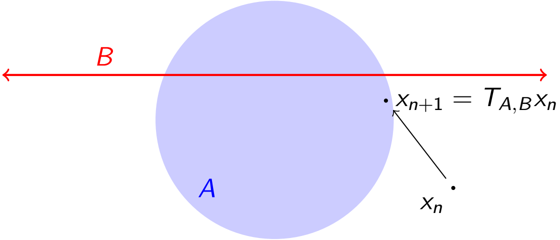
Suppose $A, B \subseteq \mathbb{R}^{m\times n}$ are closed and convex. For any $x_0 \in \mathbb{R}^{m\times n}$ define

$$x_{n+1} := T_{A,B}x_n \text{ where } T_{A,B} := \frac{I + R_B R_A}{2}.$$

Then if:

(a) $A \cap B \neq \emptyset$, (x_n) converges to a point x such that $P_A x \in A \cap B$.

(b) $A \cap B = \emptyset$, $\|x_n\| \to +\infty$.



$A := \{x \in \mathbb{R}^{m\times n} : \|x\| \leq 1\}$, $B := \{x \in \mathbb{R}^{m\times n} : \langle a, x\rangle = b\}$. (Phase retrieval)

Douglas and Rachford (and Hadamard)

Jim Douglas Jr (1927–)

H.H. Rachford Jr (192x–)

Hadamard (1865–1963)

Jim Douglas Jr (1927–)

H.H. Rachford Jr (192x–)

Hadamard (1865–1963)

- Some pictures (dates) are easier to find than others

For constraint sets $C_1, C_2, \dots, C_N$ define[1]

$$D := \{(x, x, \dots, x) \in E^N | x \in E\}, \quad C := \prod_{i=1}^{N} C_i.$$

We now have an equivalent feasibility problem with

$$x \in \bigcap_{i=1}^{N} C_i \iff (x, x, \dots, x) \in D \cap C.$$

Moreover, $T_{D,C}$ can be readily computed whenever $P_{C_1}, P_{C_2}, \dots, P_{C_N}$ can be since

$$P_D x = \left(\frac{1}{N}\sum_{i=1}^{N} x_i\right)^N, \quad P_C x = \prod_{i=1}^{N} P_{C_i} x_i.$$

[1] The set D is sometimes called the *diagonal*.

Positive semi-definite matrices

Denote the set of $n \times n$ real symmetric matrices by S_n, and the real positive semi-definite matrices by S_n^+. Set

$$C_2 := S_n^+ := \{X \in \mathbb{R}^{n \times n} : X = X^T, y^T X y \geq 0 \text{ for all } y \in \mathbb{R}^n\}.$$

Theorem (Higham 1986)

Let $X \in \mathbb{R}^{n \times n}$. Define $Y = (A + A^T)/2$ and let $Y = UP$ be a polar decomposition. Then

$$P_{C_2}(X) = \frac{Y + P}{2}.$$

(Note if X is symmetric then $Y = X$.)

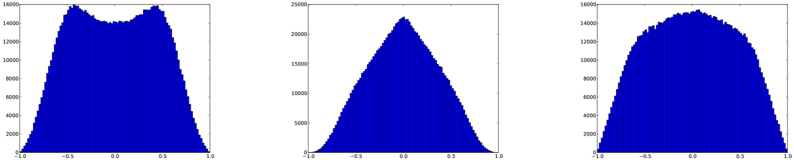
Then X is a positive semi-definite matrix that completes A if and only if $A \in C_1 \cap C_2$.

An important class of positive semi-definite matrices is the correlation matrices.

If $X_1, X_2, \dots, X_n$ are random variables, the ijth entry of the corresponding correlation matrices contains the correlation between X_i and X_j. Clearly,

$$(i,i) \in \Omega \text{ with } A_{ii} = 1 \text{ for } i = 1, 2, \dots, n. \tag{1}$$

Moreover, the entries of any matrices satisfying (1) can be shown to be contained in $[-1, 1]$.



$X_0 := Y$. $\quad X_0 := \frac{1}{2}(Y + Y^T) \in S_5$. $\quad X_0 := YY^T \in S_5$.

Figure. Distribution of entries for correlation matrices generated by choosing different initial points. Y is a random matrix in $[-1, 1]^{5\times5}$.

Stochastic matrices

Recall that a matrix $A = (A_{ij}) \in \mathbb{R}^{m\times n}$ is said to be doubly stochastic if

$$\sum_{i=1}^{m} A_{ij} = \sum_{j=1}^{n} A_{ij} = 1, A_{ij} \geq 0. \tag{2}$$

Such matrices describe the transitions of a Markov chain (in this case, $m = n$), amongst other things. The set of all doubly stochastic matrices can be represented as the intersection of

$$C_2 := \left\{ X \in \mathbb{R}^{m\times n} | \sum_{i=1}^{m} X_{ij} = 1 \text{ for } j = 1, \ldots, n \right\},$$

$$C_3 := \left\{ X \in \mathbb{R}^{m\times n} | \sum_{j=1}^{n} X_{ij} = 1 \text{ for } i = 1, \ldots, m \right\},$$

$$C_4 := \{ X \in \mathbb{R}^{m\times n} | X_{ij} \geq 0 \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n \}.$$

Then X is a double stochastic matrix that completes A if and only if $X \in C_1 \cap C_2 \cap C_3 \cap C_4$.

Euclidean Distance Matrices

Recall that $D = (d_{ij}) \in \mathbb{R}^{n \times n}$ is a Euclidean distance matrix (EDM) if there exists points $p_1, \ldots, p_n \in \mathbb{R}^r$ such that

$$d_{ij} = \|p_i - p_j\|^2.$$

Consider the problem of reconstructing an EDM, A, from a subset of its entries (A_{ij} for $(i, j) \in \Omega$). Of course, we may assume $A_{ij} = A_{ji}$ and $A_{ii} = 0$. We define

$$C_2 := \{X \in \mathbb{R}^{n \times n} : X \text{ is a EDM}\}.$$

Then X is an EDM that completes A if and only if $X \in C_1 \cap C_2$.

How do we compute $P_{C_2} X$?

Euclidean Distances Matrices

Use the following characterisation:

Theorem (Hayden–Wells 1988)

Let Q be the Householder matrix defined by

$$Q := I - \frac{2vv^T}{v^Tv}, \text{ where } v = \left[1, 1, \ldots, 1, 1 + \sqrt{n}\right]^T \in \mathbb{R}^n.$$

Then a distance matrix, X, is an EDM iff the $(n-1) \times (n-1)$ block, $\widehat{X}$, in

$$Q(-X)Q = \begin{bmatrix} \widehat{X} & d \\ d^T & \delta \end{bmatrix}$$

is positive semidefinite. In this case, X is irreducibly embeddable in $\mathbb{R}^r$ where $r = \operatorname{rank}(\widehat{X}) \leq n - 1$.

Main point: Use $\widehat{X}$ rather than X directly.

$$C_2 := \{X \in \mathbb{R}^{n\times n} : X \text{ is a EDM}\} = \{X \in \mathbb{R}^{n\times n} : \widehat{X} \in S_n^+\}.$$

Problem: Usually we know that the points defining our EDM lie in a space of given dimension (eg. $\mathbb{R}^2, \mathbb{R}^3$.).

Low-Rank Euclidean Distance Matrices

From a few slides ago:

Recall that $D = (d_{ij}) \in \mathbb{R}^{n \times n}$ is a Euclidean distance matrix (EDM) if there exists points $p_1, \ldots, p_n \in \mathbb{R}^r$ such that

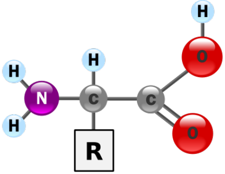
$$d_{ij} = \|p_i - p_j\|^2.$$

Furthermore, if this holds for a set of points in $\mathbb{R}^r$ then D is said to be embeddable in $\mathbb{R}^r$. If D is embeddable in $\mathbb{R}^r$, but not in $\mathbb{R}^{r-1}$, then D is said to be irreducibly embeddable in $\mathbb{R}^r$.

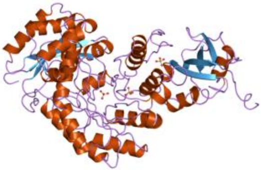
Low-rank constraints arise, for example, in the setting of compressed sensing.

Protein Confirmation Determination

Proteins are large biomolecules comprising of multiple amino acid chains.[2]



Generic amino acid



RuBisCO

- Proteins participate in virtually every cellular process!
- Protein structure $\rightarrow$ predicts how functions are performed.
- NMR spectroscopy (Nuclear Overhauser effect[3]) can be used to determine a subset of the interatomic distances (i.e. < 6Å without cellular damage).

A low-rank Euclidean distance matrix completion problem!

[2] RuBisCO (responsible for photosynthesis) has 550 amino acids (smallish).
[3] A coupling which occurs through space, rather than chemical bonds.

Protein Confirmation Determination

Let D denote the partial EDM, and $\Omega \subset \mathbb{N} \times \mathbb{N}$ the set of indices for known entries. We have the following constraints:

$$C_1 := \{X \in \mathbb{R}^{n\times n} | X_{ii} = 0, X_{ij} \geq 0, X_{ij} = X_{ji} = D_{ij} \text{ for all } (i,j) \in \Omega\}^4,$$
$$C_2 := \{X \in \mathbb{R}^{n\times n} | X \text{ is embeddable in } \mathbb{R}^3\}.$$

The reconstructed EDM is the solution to the feasibility problem

$$\text{Find } X \in C_1 \cap C_2.$$

Now,

- C_1 is a convex set (intersection of cone and affine subspace).
- C_2 is convex iff $n \leq 2$ (in which case $C_2 = \mathbb{R}^{n\times n}$).

For interesting problems, C_2 is **never convex**.

[4] Uncertainty can be incorporated by instead requiring $|X_{ij} - D_{ij}| \leq \epsilon, \forall (i,j) \in \Omega$.

Computing Projections and Reflections

The projection onto C_1 is given (point-wise) by

$$P_{C_1}(X)_{ij} = \begin{cases} D_{ij} & \text{if } (i,j) \in \Omega, \\ X_{ij} & \text{otherwise.} \end{cases}$$

Theorem (Hayden–Wells)

Let Q be the Householder matrix defined by

$$Q := I - \frac{2vv^T}{v^Tv}, \text{ where } v = \left[1, 1, \dots, 1, 1 + \sqrt{n}\right]^T \in \mathbb{R}^n.$$

Then a distance matrix, X, is a EDM iff the $(n-1) \times (n-1)$ block, $\widehat{X}$, in

$$Q(-X)Q = \begin{bmatrix} \widehat{X} & d \\ d^T & \delta \end{bmatrix}$$

is positive semidefinite. In this case, X is irreducibly embeddable in $\mathbb{R}^r$ where $r = \operatorname{rank}(\widehat{X}) \leq n - 1$.

Computing Projections and Reflections

A projection onto C_2 is given by

$$P_{C_2}(X) = -Q \begin{bmatrix} U\Lambda_+ U^T & d \\ d^T & \delta \end{bmatrix} Q,$$

where $X = U\Lambda U^T$ is a spectral decomposition with

$$\begin{aligned} \Lambda \; &:= \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n-1}) \quad \text{for } \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1}, \\ \Lambda_+ &:= \operatorname{diag}(0, \dots, 0, \max\{0, \lambda_{n-3}\}, \max\{0, \lambda_{n-2}\}, \max\{0, \lambda_{n-1}\}). \end{aligned}$$

i.e. Compute P_{C_2} from the rank 3 approximation to $\widehat{X}$ ($U\Lambda_+ U^T$).

Recall that a spectral decomposition of a real symmetric matrix, A, is given by $A = U\Lambda U^T$, where U is an orthogonal matrix, and Λ a diagonal matrix whose entries are eigenvalues of A.

Results: Six Proteins

Interatomic distances below 6Å typically constitute less than 8% of the total nonzero entries of the distance matrix.

Table. **Six Proteins: average (maximum) errors from five replications**.

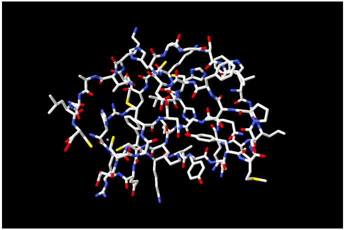
Protein	# Atoms	Rel. Error (dB)	RMSE	Max Error
1PTQ	404	-83.6 (-83.7)	0.0200 (0.0219)	0.0802 (0.0923)
1HOE	581	-72.7 (-69.3)	0.191 (0.257)	2.88 (5.49)
1LFB	641	-47.6 (-45.3)	3.24 (3.53)	21.7 (24.0)
1PHT	988	-60.5 (-58.1)	1.03 (1.18)	12.7 (13.8)
1POA	1067	-49.3 (-48.1)	34.1 (34.3)	81.9 (87.6)
1AX8	1074	-46.7 (-43.5)	9.69 (10.36)	58.6 (62.6)

$$\text{Rel. error} := 10\log_{10}\left(\frac{\|P_{C_2}P_{C_1}X_N - P_{C_1}X_N\|^2}{\|P_{C_1}X_N\|^2}\right),$$

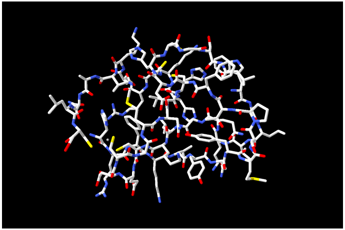
$$\text{RMSE} := \sqrt{\frac{\sum_{i=1}^m \|\hat{p}_i - p_i^{true}\|_2^2}{\#\text{ of atoms}}}, \qquad \text{Max} := \max_{1\le i\le m}\|\hat{p}_i - p_i^{true}\|_2.$$

The points $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ denote the best fitting of $p_1, p_2, \ldots, p_n$ if rotation, translation and reflections are allowed.

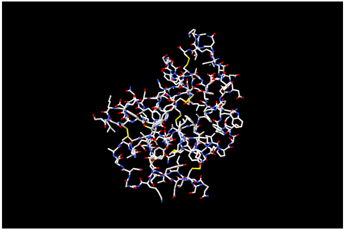
What do the reconstructions look like?



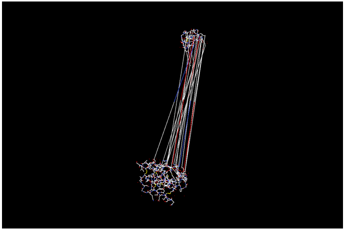
1PTQ (actual)



5,000 steps, -83.6dB (perfect)

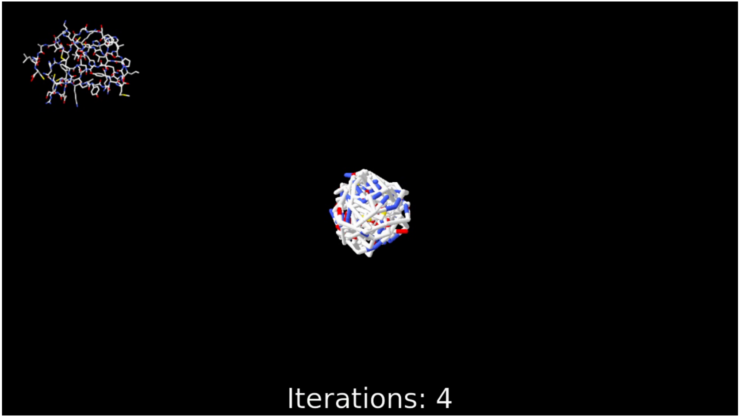


1POA (actual)



5,000 steps, -49.3dB (mainly good!)

What do reconstructions look like?



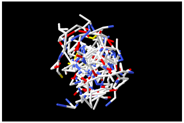
Video: First 3,000 steps of the 1PTQ reconstruction.

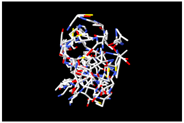
http://carma.newcastle.edu.au/DRmethods/1PTQ.html

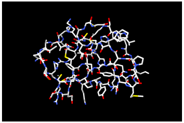
What do reconstructions look like?

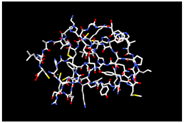
There are many projection methods, so why use Douglas-Rachford?

Douglas–Rachford method reconstruction:

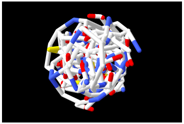

500 steps, -25 dB.

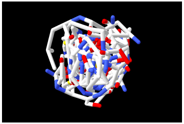

1,000 steps, -30 dB.

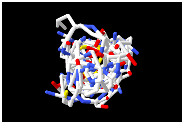

2,000 steps, -51 dB.

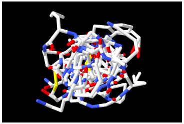

5,000 steps, -84 dB.

Alternating projection method reconstruction:


500 steps, -22 dB.


1,000 steps, -24 dB.


2,000 steps, -25 dB.


5,000 steps, -28 dB.

- Yet MAP works very well for optical abberation correction (Hubble, amateur telescopes). Why?

Hadamard Matrices

A matrix $H = (H_{ij}) \in \{-1, 1\}^{n\times n}$ is said to be a Hadamard matrix of order n if [5]

$$H^T H = nI.$$

A classical result of Hadamard asserts that Hadamard matrices exist only if $n = 1, 2$ or a multiple of 4. For orders 1 and 2, such matrices are easy to find. For example,

$$\begin{bmatrix}1\end{bmatrix}, \quad \begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}.$$

The Hadamard conjecture is concerned with the converse:

There is a Hadamard matrix of order 4n for all n?

[5] There are many equivalent characterizations and many local experts.

Hadamard Matrices

Consider now the problem of finding a Hadamard matrix of a given order – an important completion problem with structure restriction but no fixed entries. Define:

$$C_1 := \{X \in \mathbb{R}^{n\times n} | X_{ij} = \pm 1 \text{ for } i, j = 1, \ldots, n\},$$
$$C_2 := \{X \in \mathbb{R}^{n\times n} | X^T X = nI\}.$$

Then X is a Hadamard matrix if and only if $X \in C_1 \cap C_2$.

Proposition

Let $X = USV^T$ be a singular value decomposition. Then

$$\sqrt{n}UV^T \in P_{C_2}(X).$$

Hadamard Matrices

Let H_1 and H_2 be Hadamard matrices. We say H_1 are H_2 are distinct if $H_1 \neq H_2$. We say H_1 and H_2 are equivalent if H_2 can be obtained from H_1 by performing a sequence of row/column permutations, and/or multiplying row/columns by -1.

For order $4n$:

- Number of Distinct Hadamard matrices is OEIS A206712:

$$768, 4954521600, 20251509535014912000, \ldots$$

- Number of Inequivalent Hadamard matrices is OEIS A00729:

$$1, 1, 1, 1, 5, 3, 60, 487, 13710027, \ldots$$

With increasing order, the number of Hadamard matrices is a faster than exponentially decreasing proportion of total number of $\{+1, -1\}$ -matrices (there are 2^{n^2} for order n).

Hadamard Matrices

Table : Number of Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1371	534	8	1
4	1.0791	627	422	1
8	0.7368	996	996	1
12	7.1298	0	0	0
16	9.4228	0	0	0
20	20.6674	0	0	0

Checking if two Hadamard matrices are equivalent can be cast as a problem of graph isomorphism (McKay '79).

- In Sage use `is_isomorphic(graph1,graph2)`.

Hadamard Matrices

We give an alternative formulation. Define:

$$C_1 := \{X \in \mathbb{R}^{n\times n} | X_{ij} = \pm 1 \text{ for } i,j = 1,\dots,n\},$$
$$C_3 := \{X \in \mathbb{R}^{n\times n} | X^T X = \|X\|_F I\}.$$

Then X is a Hadamard matrix if and only if $X \in C_1 \cap C_2 = C_1 \cap C_3$.

Proposition

Let $X = USV^T$ be a singular value decomposition. Then

$$\sqrt{\|X\|_F}\, UV^T \in P_{C_3}(X).$$

Hadamard Matrices

Table : Number of Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1371	534	8	1
4	1.0791	627	422	1
8	0.7368	996	996	1
12	7.1298	0	0	0
16	9.4228	0	0	0
20	20.6674	0	0	0

Order	$C_1 \cap C_3$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	1.1970	505	8	1
4	0.2647	921	541	1
8	0.0117	1000	1000	1
12	0.8337	1000	1000	1
16	11.7096	16	16	4
20	22.6034	0	0	0

Recall that a matrix $X \in R^{n\times n}$ is skew-symmetric if $X^T = -X$. A skew-Hadamard matrix is a Hadamard matrix H such that $(I - H)$ is skew-symmetric. That is,

$$H + H^T = 2I.$$

Skew-Hadamard matrices are of interest, for example, in the construction of various combinatorial designs.

The number of inequivalent skew-Hadamard matrices of order $4n$ is OEIS A001119 (for $n = 2, 3, \dots$):

$$1, 1, 2, 2, 16, 54, \dots$$

Table : Number of skew-Hadamard matrices found from 1000 instances

Order	$C_1 \cap C_2$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	0.0003	1000	2	1
4	1.1095	719	16	1
8	0.7039	902	889	1
12	14.1835	43	43	1
16	19.3462	0	0	0
20	29.0383	0	0	0

Order	$C_1 \cap C_3$ Formulation			
	Ave Time (s)	Solved	Distinct	Inequivalent
2	0.0004	1000	2	1
4	1.6381	634	16	1
8	0.0991	986	968	1
12	0.0497	999	999	1
16	0.2298	1000	1000	2
20	20.0296	495	495	2

Nonograms

A nonogram puzzle consists of a blank $m \times n$ gride of pixels together with $(m+n)$ cluster-size sequences (i.e. for each row and each column). The goal is to paint the canvas with a picture subject to:

1. Each pixel must be either black or white.
2. If a row (resp. column) has a cluster-size sequences $s_1, \ldots, s_k$ then it must contain k cluster of black pixels, each separated by at least one white pixel. The ith leftmost (resp. uppermost) cluster contains s_i black pixels.

						1			
			2			4	1	2	2
2	3	1	1	5	4	1	5	2	1

1	2
	2
	1
	1
	2
2	4
2	6
	8
1	1
2	2

Nonograms

We model nonograms as a binary feasibility problem. The $m \times n$ grid is represented as a matrix $A \in \mathbb{R}^{m \times n}$. We define

$$A[i,j] = \begin{cases} 0 & \text{if the } (i,j)\text{-th entry of the grid is white,} \\ 1 & \text{if the } (i,j)\text{-th entry of the grid is black.} \end{cases}$$

Let $\mathcal{R}_i \subset \mathbb{R}^m$ (resp. $\mathcal{C}_j \subset \mathbb{R}^n$) denote the set of vectors having cluster-size sequences matching row i (resp. column j). The constraints are:

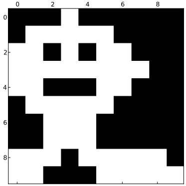
$$C_1 = \{A : A[i,:] \in \mathcal{R}_i \text{ for } i = 1, \ldots, m\},$$
$$C_2 = \{A : A[:,j] \in \mathcal{C}_j \text{ for } j = 1, \ldots, n\}.$$

Given an incomplete nonogram puzzle, A is a solution if and only if

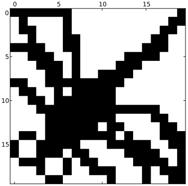
$$A \in C_1 \cap C_2.$$

Nonograms

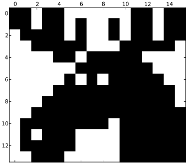
From 1000 random replication, the following nonograms were solved in every instance.



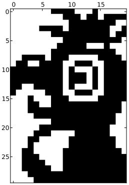
A spaceman.



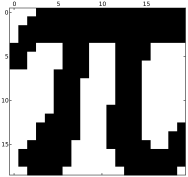
A dragonfly.



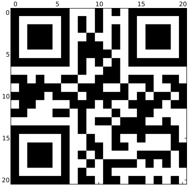
A moose.



A parrot.



The number π.



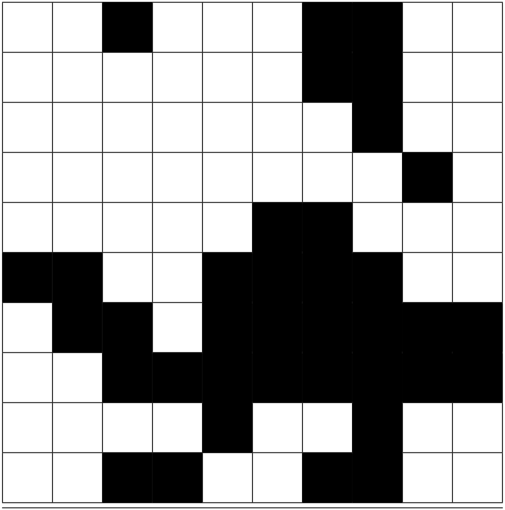
"Hello from CARMA".

Nonograms

- Computing the projections onto C_1 and C_2 is not easy.
- We do not know an efficient way to do so.
- Approach: Pre-compute all legal cluster size sequences (slow).
- Only a few Douglas–Rachford iterations are required to solve (fast).

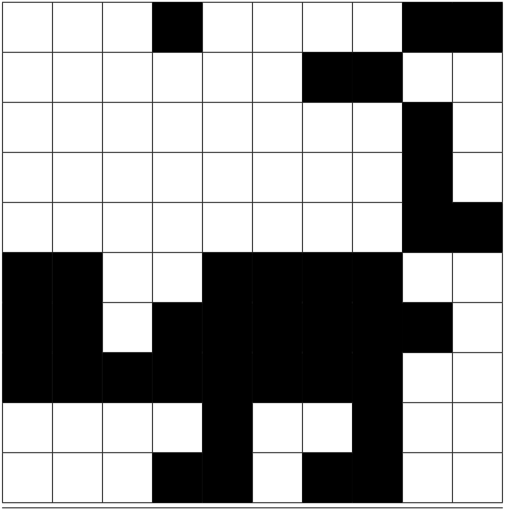
Other problems, have simple projections but require many more iterations.

> Trade-off between simplicity of projection operators and the number of iterations required.



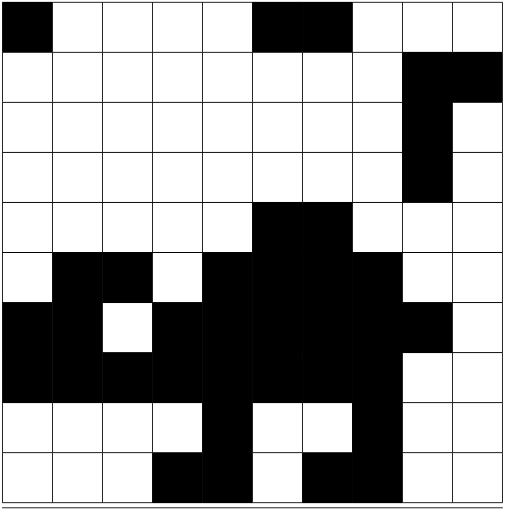
Iteration: 1

Nonograms: An example



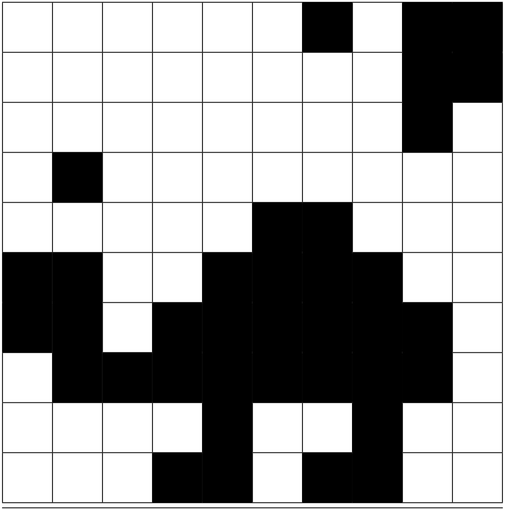
Iteration: 2

Nonograms: An example



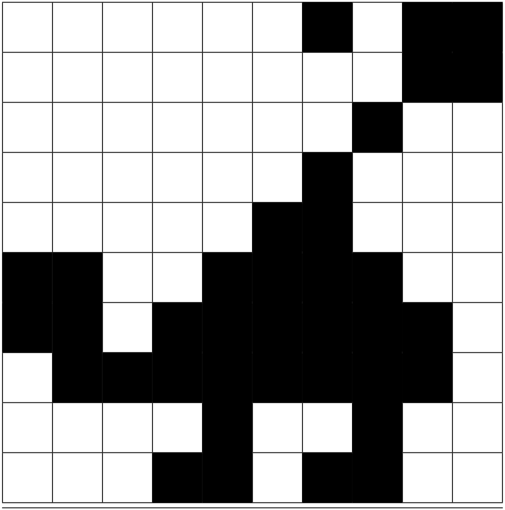
Iteration: 3

Nonograms: An example



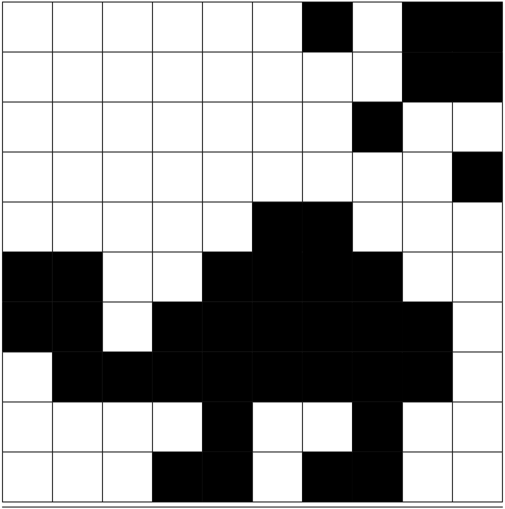
Iteration: 4

Nonograms: An example



Iteration: 5

Nonograms: An example



Iteration: 6 (solved)

Sudoku Puzzles

In Sudoku the player fills entries of an incomplete Latin square subject to the constraints:

- Each row contains the numbers 1 through 9 exactly once.
- Each column contains the numbers 1 through 9 exactly once.
- Each 3×3 sub-block contains the numbers 1 through 9 exactly once.

		5	3					
8							2	
	7			1		5		
4					5	3		
	1			7				6
		3	2				8	
	6		5					9
		4					3	
					9	7		

1	4	5	3	2	7	6	9	8
8	3	9	6	5	4	1	2	7
6	7	2	9	1	8	5	4	3
4	9	6	1	8	5	3	7	2
2	1	8	4	7	3	9	5	6
7	5	3	2	9	6	4	8	1
3	6	7	5	4	2	1	8	9
9	8	4	7	6	1	2	3	5
5	2	1	8	3	9	7	6	4

Figure. An incomplete Sudoku (left) and its unique solution (right).

- The Douglas–Rachford algorithm applied to the natural integer feasibility problem fails (exception: $n^2 \times n^2$ Sudokus where $n = 1, 2$).

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

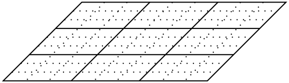
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^9 \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times9\times9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



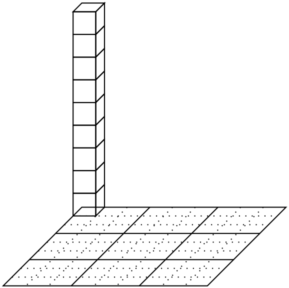
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



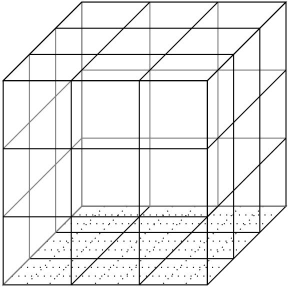
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^9 \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$$C_1 = \{X : X_{ij} \in E\}$$
$$C_2 = \{X : X_{ik} \in E\}$$
$$C_3 = \{X : X_{jk} \in E\}$$
$$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$$
$$C_5 = \{X : X \text{ matches original puzzle}\}$$

A solution is any $X \in \bigcap_{i=1}^5 C_i$.

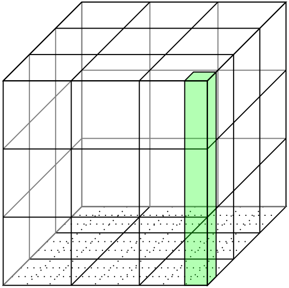
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^9 \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

A solution is any $X \in \bigcap_{i=1}^5 C_i$.

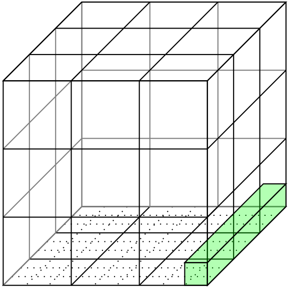
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$$C_1 = \{X : X_{ij} \in E\}$$
$$C_2 = \{X : X_{ik} \in E\}$$
$$C_3 = \{X : X_{jk} \in E\}$$
$$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$$
$$C_5 = \{X : X \text{ matches original puzzle}\}$$

A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

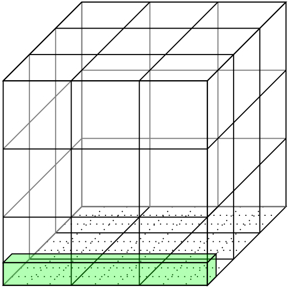
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

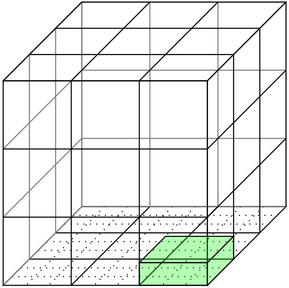
[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: A Binary Model[5]

Let $E = \{e_j\}_{j=1}^{9} \subset \mathbb{R}^9$ be the standard basis. Define $X \in \mathbb{R}^{9\times 9\times 9}$ by

$$X_{ijk} = \begin{cases} 1 & \text{if } ij\text{th entry of the Sudoku is } k, \\ 0 & \text{otherwise.} \end{cases}$$

The idea: Reformulate integer entries as binary vectors.



The constraints are:

$C_1 = \{X : X_{ij} \in E\}$

$C_2 = \{X : X_{ik} \in E\}$

$C_3 = \{X : X_{jk} \in E\}$

$C_4 = \{X : \text{vec}(3 \times 3 \text{ submatrix}) \in E\}$

$C_5 = \{X : X \text{ matches original puzzle}\}$

A solution is any $X \in \bigcap_{i=1}^{5} C_i$.

[5] Veit Elser was the first to realise the usefulness of this binary formulation for solving Sudoku via Douglas–Rachford methods.

Sudoku Puzzles: Computing projections

Proposition (projections onto permutation sets)

Denote by $\mathcal{C} \subset \mathbb{R}^m$ the set of all vector whose entries are permutations of $c_1, c_2, \dots, c_m \in \mathbb{R}$. Then for any $x \in \mathbb{R}^m$,

$$P_{\mathcal{C}} x = [\mathcal{C}]_x,$$

where $[\mathcal{C}]_x$ is the set of vectors $y \in \mathcal{C}$ such that ith largest index of y has the same index in y as the ith largest entry of x, for all indices i.

- $[\mathcal{C}]_x$ be computed efficiently using sorting algorithms.
- Choosing $c_1 = 1$ and $c_2 = \cdots = c_m = 0$ gives[6]

$$P_E x = \{e_i : x_i = \max\{x_1, \dots, x_m\}\}.$$

  Formulae for $P_{C_1}, P_{C_2}, P_{C_3}$ and P_{C_4} easily follow.
- P_{C_5} is given by setting the entries corresponding to those in the incomplete puzzle to 1, and leaving the remaining untouched.

[6] A direct proof of this special case appears in Jason Schaad's Masters thesis.

Sudoku Puzzles: The Algorithm

1. Initialize: $x_0 := (y, y, y, y, y) \in D$ for some random $y \in [0,1]^{9\times9\times9}$.
2. Iteration: By setting

$$x_{n+1} := T_{D,C}x_n = \frac{x_n + R_C R_D x_n}{2}.$$

3. Termination: Either if a solution is found, or 10000 iteration have been performed. More precisely, round$(P_D x_n)$ ($P_D x_n$ pointwise rounded to the nearest integer) is a solution if

$$\text{round}(P_D x_n) \in C \cap D.$$

Taking round$(\cdot)$ is valid since the solution is binary.

Sudoku Puzzles: An Experiment

We consider the following test libraries frequently used by programmers to test their solvers.

1. Dukuso's top95 and top1465.
2. First 1000 puzzles from Gordan Royle's minimum Sudoku – puzzles with 17 entries (the best known lower bound on the entries required for a unique solution).
3. reglib-1.3 – 1000 test puzzle suited to particular human style techniques.
4. ksudoku16 and ksudoku25 – a collection around 30 instances (various difficulties) generated with *KSudoku*. Contains larger 16×16 and 25×25 puzzles.[7]

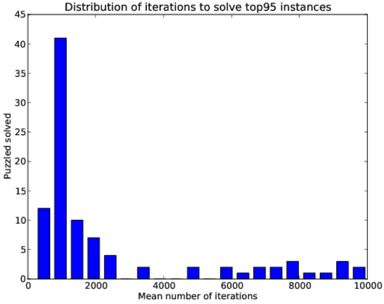
[7] Generating "hard" instances is a difficult problem.

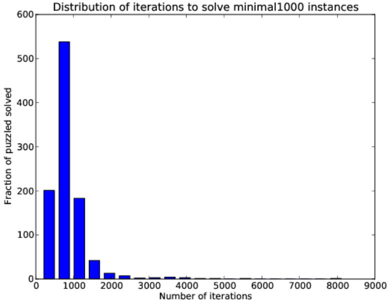
Computational Results: Success Rate

From 10 random replications of each puzzle:

Table. % Solved by the Douglas–Rachford method

top95	top1465	reglib-1.3	minimal1000	ksudoku16	ksudoku25
86.53	93.69	99.35	99.59	92	100





- If a instance was solved, the solution was usually found within the first 2000 iterations.

Computational Example: A 'Nasty' Sudoku

This 'nasty' Sudoku[8] cannot be solved reliably (20.2% success rate) by the Douglas–Rachford method.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

[8] This is a modified version of an example due to Veit Elser.

Computational Example: A 'Nasty' Sudoku

This 'nasty' Sudoku[8] cannot be solved reliably (20.2% success rate) by the Douglas–Rachford method.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Other "difficult" Sudoku puzzles do not cause the Douglas–Rachford method any trouble.

- AI escargot = 98.5% success rate.

[8] This is a modified version of an example due to Veit Elser.

Computational Example: A 'Nasty' Sudoku

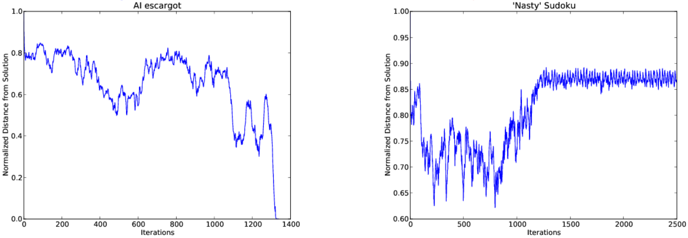
This 'nasty' Sudoku[8] cannot be solved reliably (20.2% success rate) by the Douglas–Rachford method.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Other "difficult" Sudoku puzzles do not cause the Douglas–Rachford method any trouble.

- AI escargot = 98.5% success rate.

Figure. Distance to the solution by iterations



[8] This is a modified version of an example due to Veit Elser.

Computational Example: A 'Nasty' Sudoku

We considered solving the puzzles obtained by removing any single entry from the 'Nasty' Sudoku.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Computational Example: A 'Nasty' Sudoku

We considered solving the puzzles obtained by removing any single entry from the 'Nasty' Sudoku.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Success rate when any single entry is removed:

- Top left 7 = 24%
- Any other entry = 99%

Computational Example: A 'Nasty' Sudoku

We considered solving the puzzles obtained by removing any single entry from the 'Nasty' Sudoku.

7					9		5	
	1						3	
		2	3			7		
		4	5				7	
8						2		
					6	4		
	9			1				
	8			6				
		5	4					7

Success rate when any single entry is removed:

- Top left 7 = 24%
- Any other entry = 99%

Number of solutions when any single entry is removed:

- Top left 7 = 5
- Any other entry = 200–3800

Is the Douglas–Rachford method hindered by an abundance of 'near' solutions?

Computational Results: Performance Comparison

We compared the Douglas–Rachford method to the following solvers:

1. Gurobi binary program – Solves the same binary model using integer programming techniques.
2. YASS (Yet another Sudoku solver) – First applies a reasoning algorithm to determine possible candidates for each empty square. If this does not completely solve the puzzle, a deterministic recursive algorithm is used.
3. DLX – Solves an exact cover formulation using the *Dancing Links* implementation of Knuth's *Algorithm X* (non-deterministic, depth-first, back-tracking).

Table. Average Runtime (seconds).[9]

	top95	reglib-1.3	minimal1000	ksudoku16	ksudoku25
DR	1.432	0.279	0.509	5.064	4.011
Gurobi	0.063	0.059	0.063	0.168	0.401
YASS	2.256	0.039	0.654	-	-
DLX	1.386	0.105	3.871	-	-

[9]Some solvers are only designed to handle 9×9 puzzles.

Concluding Remarks

- We presented with a feasibility problem, it is well worth seeing if Douglas–Rachford can deal with it – it is conceptually simple and easy to implement.
- Optimised implementations of the algorithm. For instance, in the protein reconstruction

Concluding Remarks

- We presented with a feasibility problem, it is well worth seeing if Douglas–Rachford can deal with it – it is conceptually simple and easy to implement.
- Optimised implementations of the algorithm. For instance, in the protein reconstruction
  - Update projection with heuristics (keeping Q fixed) or infrequently.
  - Impose more constraints on protein distances.
  - Exploit symmetry better.
  - 'profile' the successful and unsuccessful cases.

---

1. **Douglas–Rachford feasibility methods for matrix completion problems**. F.J. Aragón Artacho, J.M. Borwein & M.K. Tam. *Submitted* (2013). http://arxiv.org/abs/1308.4243.
2. **Recent Results on Douglas-Rachford methods for combinatorial optimization problems**. F.J. Aragón Artacho, J.M. Borwein & M.K. Tam. *Submitted* (2013). http://arxiv.org/abs/1305.2657.

Many resources can be found at the companion website:

`http://carma.newcastle.edu.au/DRmethods/`