Why do these projection algorithms work on combinatorial feasibility problems?

Contact Quang:
Email: theepiccowoflife@gmail.com
Github: https://github.com/TheEpicCowOfLife

Quang Ong, supervised by Matthew Tam
January 2021

Contact Matthew Tam:
Email: matthew.tam@unimelb.edu.au
Website: https://matthewktam.github.io/

The problem class:

Let E be an Euclidean space, and consider N constraint sets C_1, C_2, …, $C_N \subseteq E$. A **feasibility problem** asks for an $x \in E$ so that

$$x \in C_1 \cap C_2 \cap \cdots \cap C_n$$

Projection operator

Given a set A, define the projection operator P_A as

$$P_A(x) = \{c \in A \mid \|x - c\| \text{is minimal}\}$$

In other words, $P_A(x)$ projects x onto the set of closest points in A.
Define the reflection operator, R_A as

$$R_A(x) = 2P_A(x) - x$$

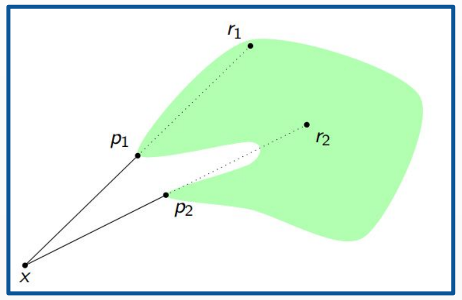


Illustration of projections $p_1, p_2 \in P_A(x)$ and reflections $r_1, r_2 \in R_A(x)$ Image credit: [3]

Douglas-Rachford (DR) algorithm

Suppose we had 2 constraint sets, $A, B \subseteq E$

Let $x_0 \in E$. Define the sequence x_k by

$$x_{k+1} = \frac{Id + R_B R_A}{2} x_k$$

It is known that when A and B are **convex**, and $A \cap B \neq \emptyset$, $P_A(x_k)$ converges to an $x \in A \cap B$.

To generalise to N sets $C_1 \dots C_N$, define new constraint sets

$$\mathbf{C} = C_1 \times \cdots \times C_N \text{ and } \mathbf{D} = \{(x, x, \dots, x) \mid x \in E\}$$

so $\boldsymbol{C}, \boldsymbol{D} \in E^N$ and run Douglas Ratchford on $\boldsymbol{C}$ and $\boldsymbol{D}$

Malitsky-Tam (MT) algorithm

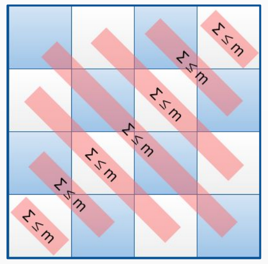
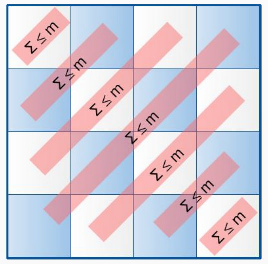
An algorithm recently devised to prove a theoretical bound and can be specialised to find the intersection of N **convex** sets. However, its numerical performance has not been as well studied as Douglas-Rachford. See page 9, [2] for the specific algorithm.

N-M Queens

N-M queens problem generalises the N queens problems. This problem asks for NM queens to be placed on an $N \times N$ chessboard such that at exactly M queens are placed on each row and column, and at most M queens are placed in each diagonal.

To formulate it as a feasibility problem, let $E = \mathbb{R}^{N \times N}$, such that for all $x \in E$, $x_{ij} = 1$ if (i,j) contains a queen and 0 otherwise. Then, we can define the following 4 constraint sets

- Each row must add to M
- Each column must add to M
- Each forward diagonal must add to at most M
- Each backward diagonal must add to at most M

We also impose that each constraint set contains only entries from 0-1, which empirically improves performance (See section 4.1, [1]

However, note these constraints are **not convex**, so there is no satisfactory mathematical justification for why this algorithm works.

Experiment:

Each trial proceeded as follows:

```
Initialise x_0 randomly
For k from 1 to 5000:
    Calculate x_k
    If x_k is a valid solution:
        Record the success, and
        value of k in the median
```

Noise optimization

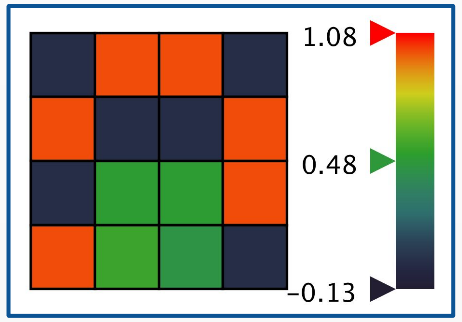
Adding a small amount of noise uniformly distributed within [-0.01, 0.01] to each cell each iteration has a profound effect on the performance of the algorithm.

Results:

For $(N,M) = (20,2)$, over 200 trials,

	Success rate	Median iterations	Average time per successful trial (ms)
DR, no noise	87.5%	189	12.789
DR, with noise	100%	112	9.700
MT, no noise	99.5%	1069	59.357
MT, with noise	15.5%	3165	202.903

A visualiser was coded to inspect the performance of the algorithm. With DR, many of the starting values failed because the algorithm would "get stuck" on a fixed point or small cycle, that was not a valid solution. However these fixed points are quite unstable, and adding noise would allow the algorithm to "escape", improving the performance.



Pictured is $P_D(x_k)$, for a 4-2 queens problem generated by DR with no noise . For k > 100 this board remains relatively unchanged.

MT on the other hand, does not appear to suffer from getting stuck in fixed points, and adding noise simply disrupts the entire algorithm. Hence DR's tolerance to noise is actually quite notable!

Learn more about this project: explorations in sudoku, and more details!

There are more marvellous insights for which this poster is too narrow to contain, so here is a link to my blog post!
https://theepiccowoflife.github.io/2022/02/05/projalgo

References

[1] Aragón Artacho, F.J., Campoy, R. & Tam, M.K. The Douglas–Rachford algorithm for convex and nonconvex feasibility problems. Math Meth Oper Res 91, 201–240 (2020). https://doi.org/10.1007/s00186-019-00691-9

[2] Yura Malitsky, Matthew Tam, Resolvent Splitting for Sums of Monotone Operators with MInimal Lifting, https://arxiv.org/abs/2108.02897

[3] Matthew Tam, Douglas-Ratchford for Combinatorial Optimisation, Accessed 04/02/2022 https://matthewktam.github.io/talks/mtam13-amssc.pdf

draft lol

Quang Ong, supervised by Matthew Tam
January 2021

The problem class:

Let E be an euclidean space, and $A_1, A_2 \dots A_n$ be sets $A_i \subseteq E$. A **feasibility problem** asks for an $x \in E$ such that

$$x \in A_1 \cap A_2 \cap \dots \cap A_n$$

Projection, Reflection operator

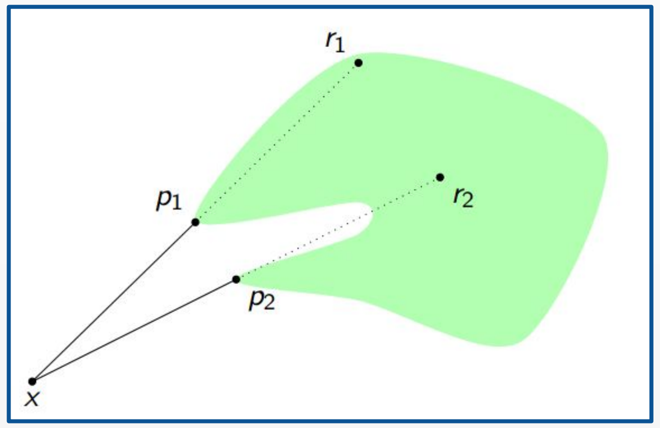
Define the (possibly set valued) projection operator P_A as

$$P_A(x) = \{c \in A \mid \|x-c\| \text{ is minimal}\}$$

In other words, $P_A(x)$ projects x to the closest point in A.

The reflection operator R_A is the (set valued mapping) defined as

$$R_A = 2P_A - Id$$S



The green blob is our set A, and $p_1, p_2 \in R$

Douglas-Ratchford algorithm

Suppose we had 2 sets, $A, B \subset E$

Let $x_0 \in E$. Define the sequence x_k by

$$x_{k+1} = \frac{Id + R_B R_A}{2}x_k$$

It is known that when A and B are **convex**, and $A \cap B \neq \emptyset$, $P_A x_k$ converges to an $x \in A \cap B$.

To generalise to n sets $A_1, \dots, A_n$, define new constraint sets

$$C = A_1 \times \dots \times A_n \text{ and } D = \{(x,x,\dots,x) \mid x \in E\}$$

so $C,D \in E^n$ and run Douglas Ratchford on C and D.

Malitsky-Tam algorithm

An algorithm recently devised to prove a theoretical bound for a more general form of this class of problem. However, its numerical performance has not been as well studied as Douglas-Ratchford. See page 9, [x] for the algorithm

Suppose we had N constraint sets, $A, B \subset E$

Let $x_0 \in E$. Define the sequence x_k by

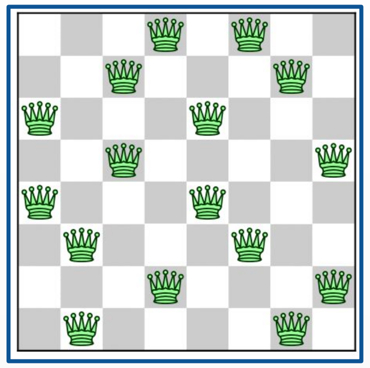
$$x_{k+1} \in \frac{\text{Id} + R_B R_A}{2} x_k$$

It is known that when A and B are **convex**, and $A \cap B \neq \emptyset$, $P_A x_k$ converges to an $x \in A \cap B$.

To generalise to N sets see [x] for the product space formulation

N-M Queens

N-M queens problem generalises the N queens problems. This problem asks for nm queens to be placed on an nxn chessboard such that at exactly m queens are placed on each row and column, and at most m queens are placed in each diagonal.



(A valid solution: will take a better photo later)

To formulate it as a feasibility problem, let $E = R^{(n \times n)}$, where for $x \in E$, $x_{ij} = 1$ if (i,j) contains a queen and 0 otherwise. Then, 4 constraint sets can be defined for rows, column

$$x_{k+1} = \frac{Id + R_B R_A}{2} x_k$$