Supporting hyperplanes and dual quadratic programming for set intersection problems:

Algorithms, implementation and numerical results

C.H. Jeffrey Pang

July 4, 2013

Outline

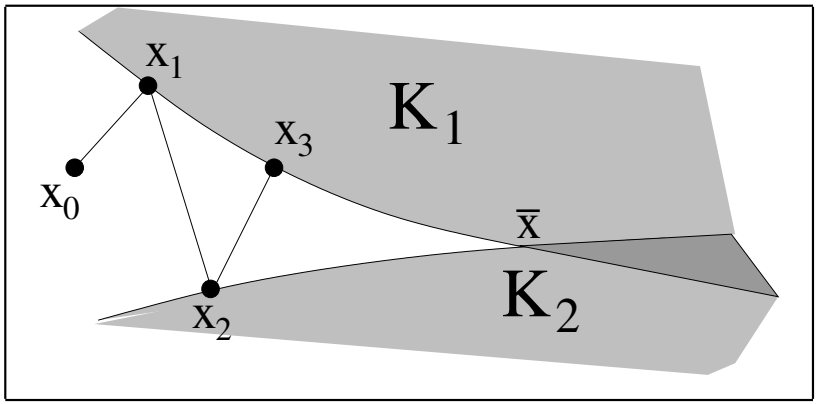
- Set intersection problems: Problem statement
- Algorithms
- Implementation
- Numerical results
- Theory

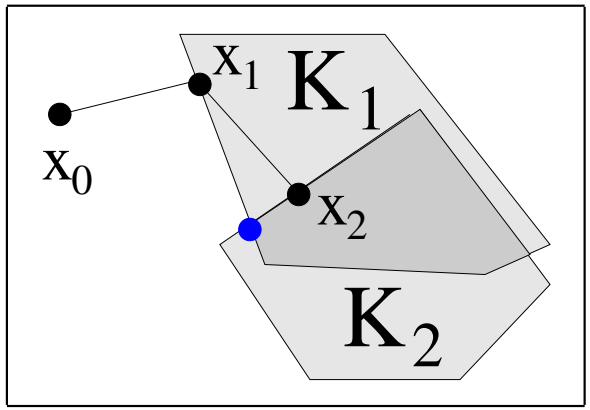
Problem

Let X be a Hilbert space,

Closed convex sets $K_l \subset X$, $i = 1, \ldots, r$, & $K := \cap_{l=1}^{r} K_l$.

- **Set Intersection Problem (SIP)**: Find $x \in K$.
  - For starting iterate $x_0 \in X$,
    Alternating Projections converge to some $x \in K$.





- **Best Approximation Problem (BAP)**:

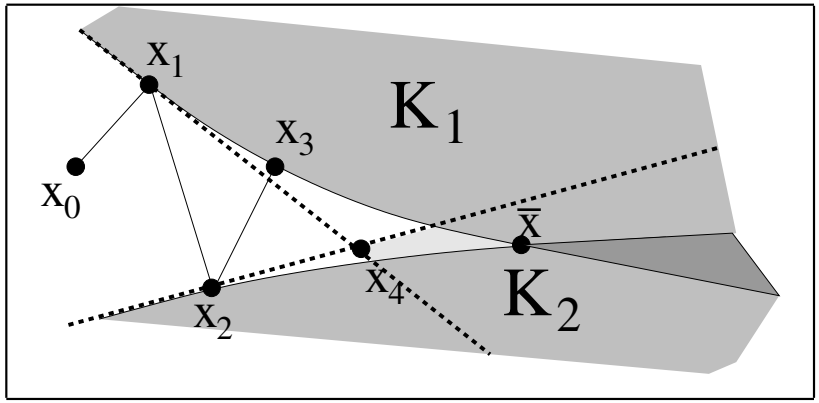
$$\text{For } x_0 \in X, \text{ find } P_K(x_0) := \arg\min_{x \in K} \|x - x_0\|.$$

  - Alternating Projections may not converge to $P_K(x_0)$.
  - **Dykstra's algorithm** reduces Best Approximation to alternating projection by adding correction vectors.

Applications of Alternating Projection algorithm

1. Solving linear equations;
2. The Dirichlet problem which has in turn inspired the "domain decomposition" industry;
3. Probability and statistics;
4. Computing Bergman kernels;
5. Approximating multivariate functions by sums of univariate ones;
6. Least change secant updates;
7. Multigrid methods;
8. Conformal mapping;
9. Image restoration;
10. Computed tomography.

Supporting Hyperplanes & Quadratic Programming



- **Supporting hyperplanes** at x_1 and x_2
  - Rules out x_3
  - Polyhedron better than each K_l taken singly.
    [Projection of x_0 onto polyhedron (x_4) better than x_3.]
  - Faster convergence if few second order effects.
  - Projection is a **QP**: $\min\{\|x_0 - x\|^2 \mid x \text{ in polyhedron}\}$.
  - (García-Palomares '98, '01): $K_l = \{x \mid f_l(x) \leq 0\}$ case.

This talk:
Algorithms, **Implementation** and **Numerical results**
for general Best Approximation and Set Intersection Problems.

Dual active set quadratic programming

Consider

$$QP(A, y): \quad \min_{x \in \mathbb{R}^n} \quad q(x) := \frac{1}{2}\|y - x\|^2$$
$$\text{s.t.} \quad c_i^T x - b_i \geq 0 \text{ for all } i \in A,$$

Suppose solution of QP(A,y) known, & $|A|$ is increased by one

- Use Dual QP to find new solution (from the old solution)
- We recall method of Goldfarb-Idnani (1983)
  - Uses a step which we call the inner GI step.

We say (x, A) is an S-pair if

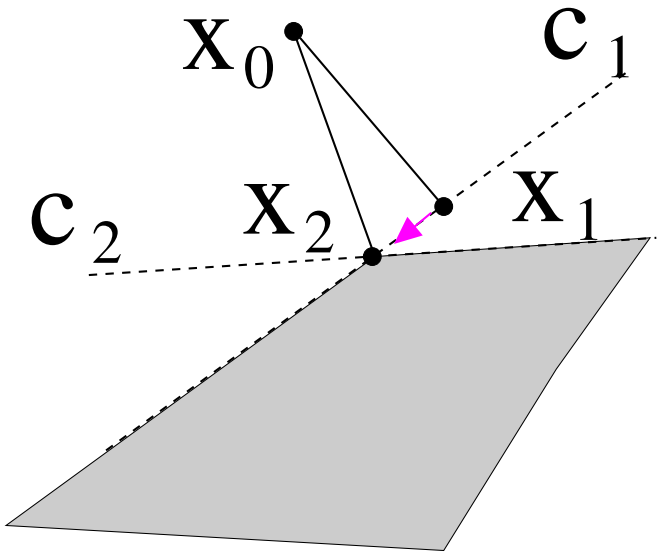
- $A \subset \mathbb{N} = \{1, 2, 3 \dots\}$ (The active set)
- $c_i^T x - b_i = 0$ for all $i \in A$.
- x is the optimal solution to $QP(A, y)$.

$$QP(A, y): \quad \min_{x \in \mathbb{R}^n} \quad q(x) := \frac{1}{2}\|y - x\|^2$$
$$\text{s.t.} \quad c_i^T x - b_i \geq 0 \text{ for all } i \in A,$$

Suppose (x, A) is an S-pair, and $i' \in \mathbb{N} \backslash A$.
Inner GI step:

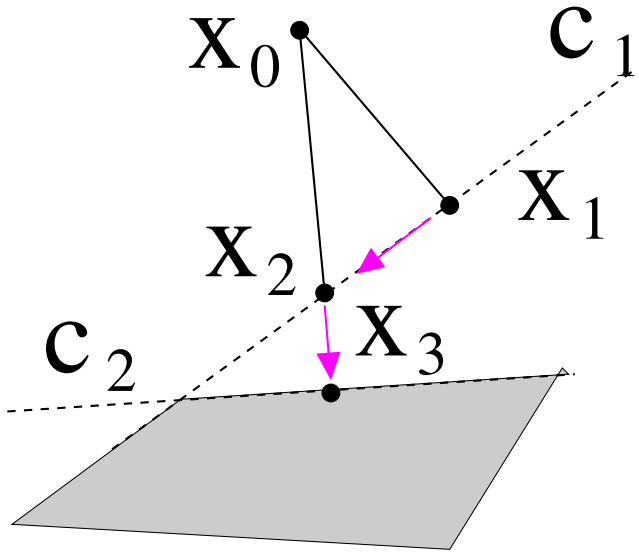
- Finds $A' \subset A \cup \{i'\}$ and x' such that
  - (x', A') is an S-pair
  - $q(x') > q(x)$
- Also updates KKT multipliers $u \geq 0$ s.t. $x - y = \sum_{i \in A} u_i c_i$.

Inner Goldfarb-Idnani (1983) step



- x_1 optimal solution if only one constraint c_1.
- If c_2 added, then move along pink direction to satisfy c_2.
- Reach x_2, which is optimal.

Inner Goldfarb-Idnani (1983) step

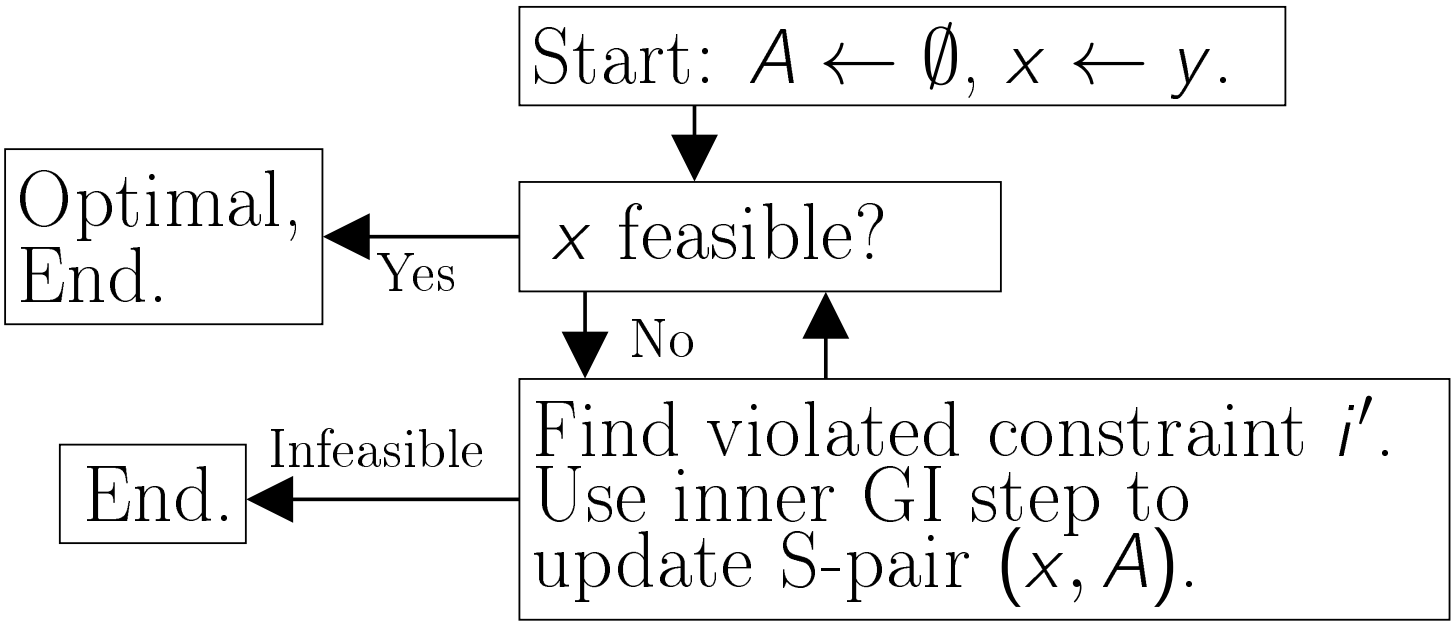


- In this case, move along same direction again.
- But the KKT multipler to c_1, which vary linearly on pink direction, is zero at x_2.
- Drop constraint c_1, and reach the optimum x_3.

Goldfarb- Idnani (1983) dual QP algorithm

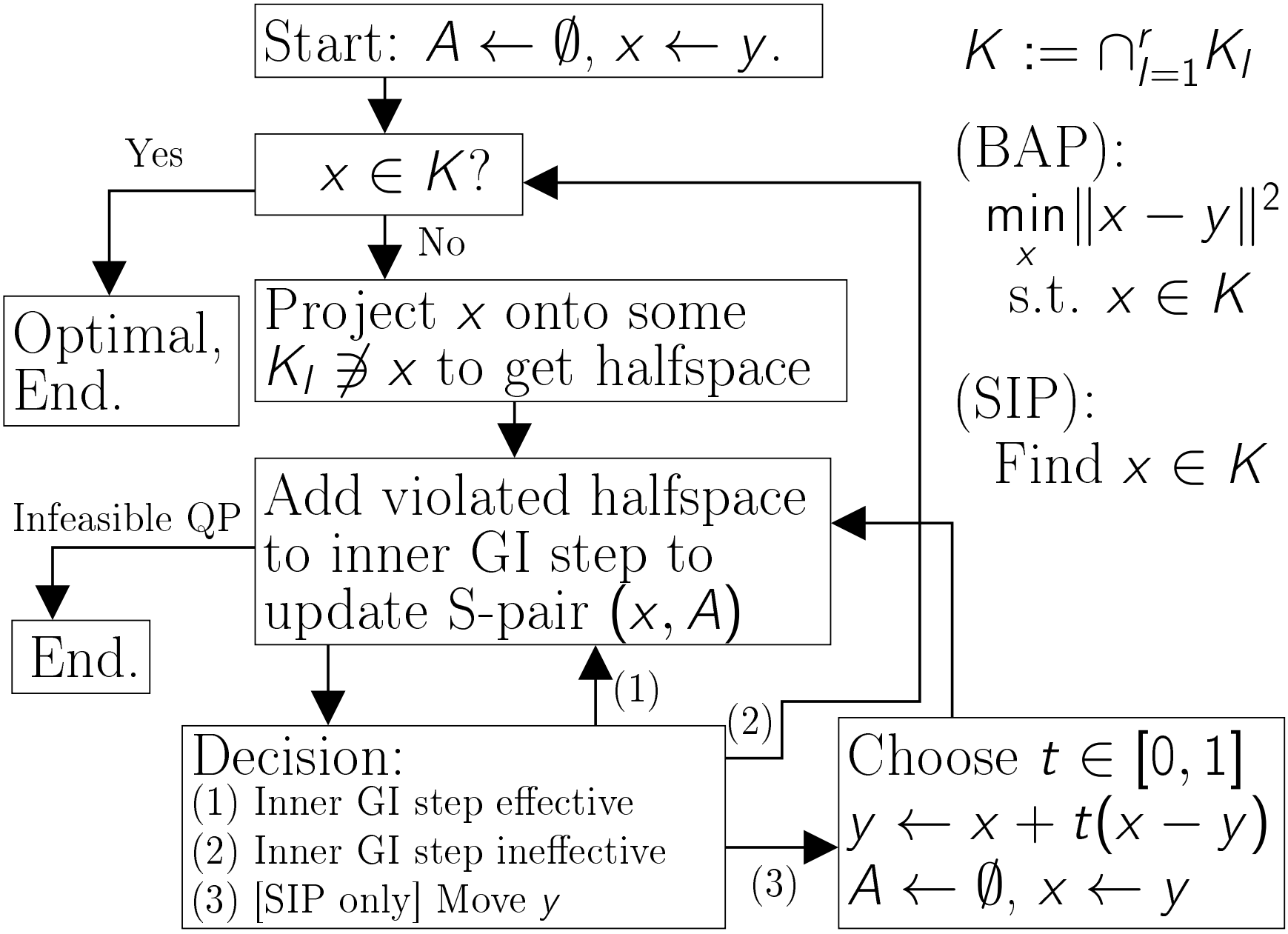
$QP(\{1,\ldots,m\}, y)$: $\min_{x\in\mathbb{R}^n} q(x) := \frac{1}{2}\|y-x\|^2$

s.t. $c_i^T x - b_i \geq 0$ for all $i \in \{1,\ldots m\}$,



- Converges in finitely many inner GI steps because
  - Dual QP: x feasible $\Longrightarrow$ KKT and Optimality
  - $q(\cdot)$ strictly increasing, and
  - only finitely many possibilities for $A \subset \{1,\ldots,m\}$.
- When new constraint added, do more inner GI steps.

The SHDQP algorithm (Look at BAP first)



$K := \cap_{l=1}^{r} K_l$

(BAP):

$$\min_x \|x - y\|^2 \quad \text{s.t. } x \in K$$

(SIP):

Find $x \in K$

Set Intersection Problems

Recall the problem of finding an $x \in K = \cap_{l=1}^{r} K_l$.
Let $[i]$ be the integer in $\{1, \ldots, r\}$ such that r divides $i - [i]$.

- Alternating projections:
  Starting from $y_0 \in \mathbb{R}^n$, iterates $\{y_i\}$ generated by

$$y_{i+1} = P_{K_{[i]}}(y_i) \text{ for } i = 0, 1, 2, \ldots,$$

- Alternating reflections:

$$y_{i+1} = 2P_{K_{[i]}}(y_i) - y_i \text{ for } i = 0, 1, 2, \ldots.$$

- Douglas-Rachford:

$$\begin{aligned} y_{i+1} &= 2P_{K_1}(y_i) - y_i, \\ y_{i+2} &= \frac{1}{2}\left[[2P_{K_2}(y_{i+1}) - y_{i+1}] + y_i\right] \text{ for } i = 0, 2, 4, \ldots. \end{aligned}$$

(Letting $R_C(x) := 2P_C(x) - x$, D-R scheme also
$y_{i+2} = \frac{1}{2}[R_{K_2} \circ R_{K_1}(y_i) + y_i]$ for $i = 0, 2, 4, \ldots.$)

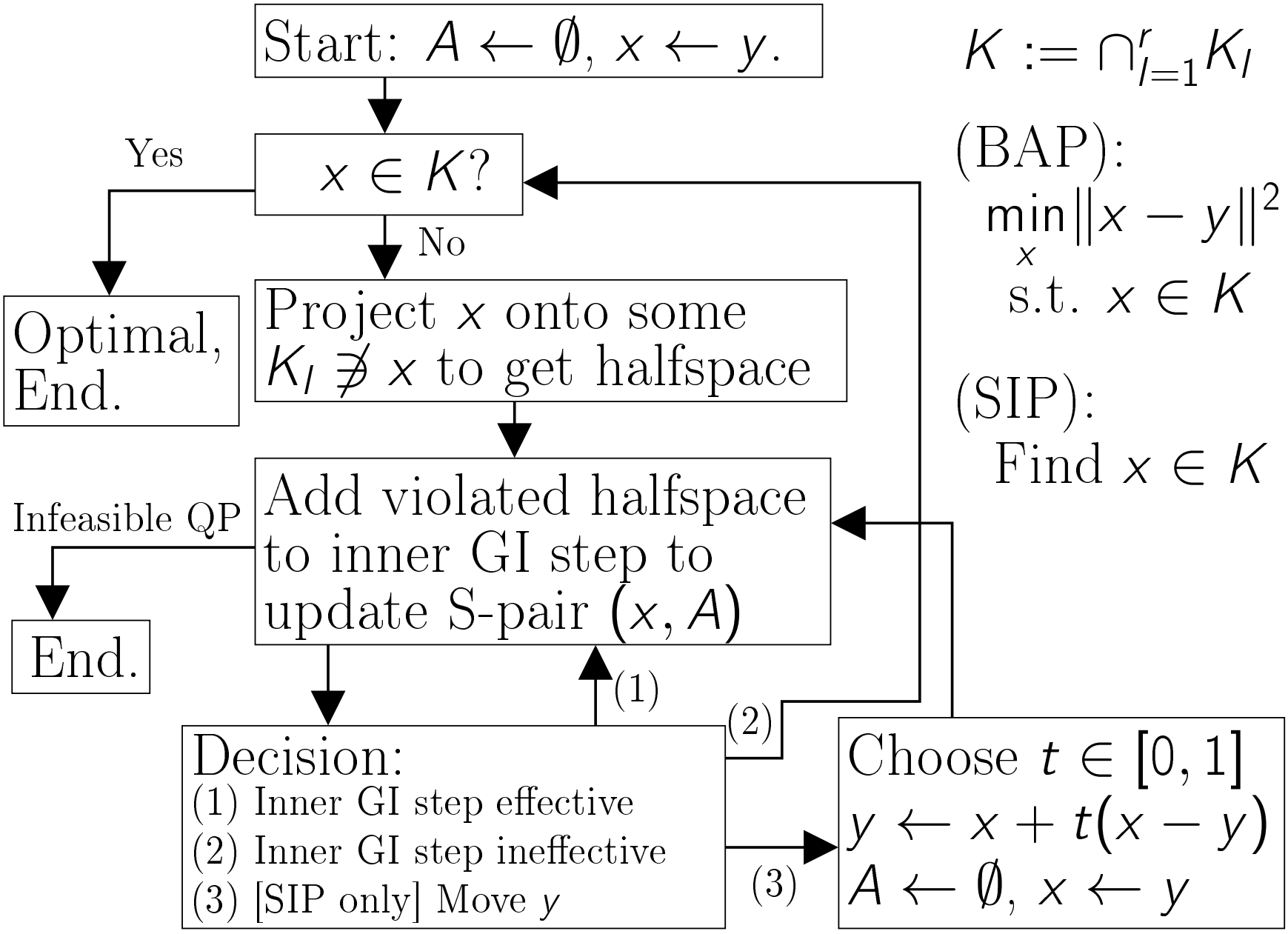
A possible adaptation of alternating projections is as follows:

Let $y = y_0$, $m = 0$
While y not close enough to $K = \cap_{l=1}^{r} K_l$
    $m \leftarrow m + 1$
    Find l s.t. $y \notin K_l$, and project y onto K_l to get
        $c_m \in \mathbb{R}^n$ & $b_m \in \mathbb{R}$ s.t. $\{\tilde{x} \mid c_m^T \tilde{x} - b_m \geq 0\}$ supports K_l.
    $x \leftarrow P_{K_A}(y)$ (K_A: polyhedron defined by $A \subset \{1, \ldots, m\}$.)
    Choose $t \in \{-1\} \cup [0, 1]$ and update $y \leftarrow x + t(x - y)$.
end While.

Particular cases of above algorithm:

- $t = -1$: $x + t(x - y) = y$, so no update.
- $A \equiv \{m\}$ and $t = 0$ always: Alternating projections.
- $A \equiv \{m\}$ and $t = 1$ always: Alternating reflections.
- $t \in [0, 1]$ ensures limit of y is in K (Fejér monotonicity).

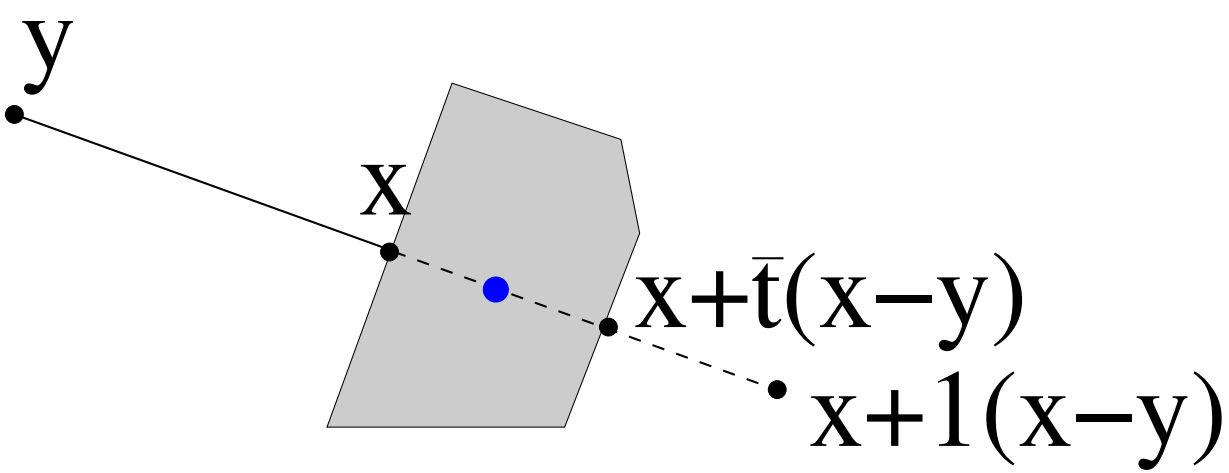
The SHDQP algorithm



A small wrinkle in SHDQP (SIP)

There is some $\bar{t} \in [0, \infty]$ s.t. the interval $[x, x + \bar{t}(x - y)]$ is contained in polyhedron produced so far.

- If $\bar{t} < 2$, then should choose midpoint of this line segment. (In this diagram, $\bar{t} < 1$.)



Implementation in Matlab

Syntax of Matlab program SHDQP:

```
f_list={@(x)nmap_cir(x,[1;0],3),...
        @(x)nmap_cir(x,[-1;0],3)};
[x status C b q A r u Q R]=...
     shdqp(y,f_list,[1 1],iter,P,w);
```

Inputs

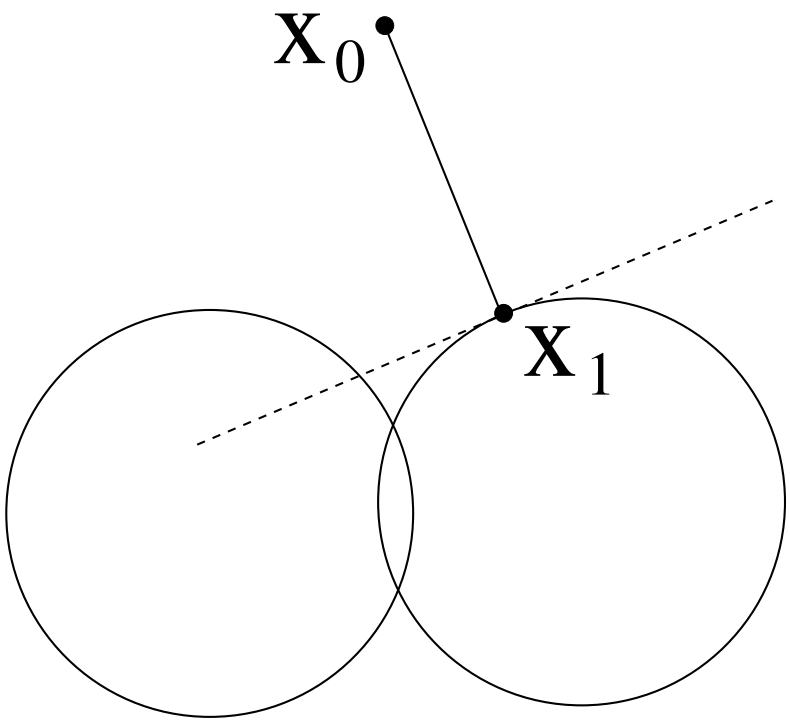
- `y`: Starting point
- `f_list`: List of normal mapping functions
  - This example: Projecting onto $\mathbb{B}\big((\pm 1, 0), 3\big)$.
- `iter`: Number of iterations
- `P`: Type of problem (0: BAP, 1: SIP)
- `w`: Parameter in $[0, 1]$ for SIP

```
f_list={@(x)nmap_cir(x,[1;0],3),...
        @(x)nmap_cir(x,[-1;0],3)};
[x status C b q A r u Q R]=...
     shdqp(y,f_list,[1 1],iter,P,w);
```

Outputs:

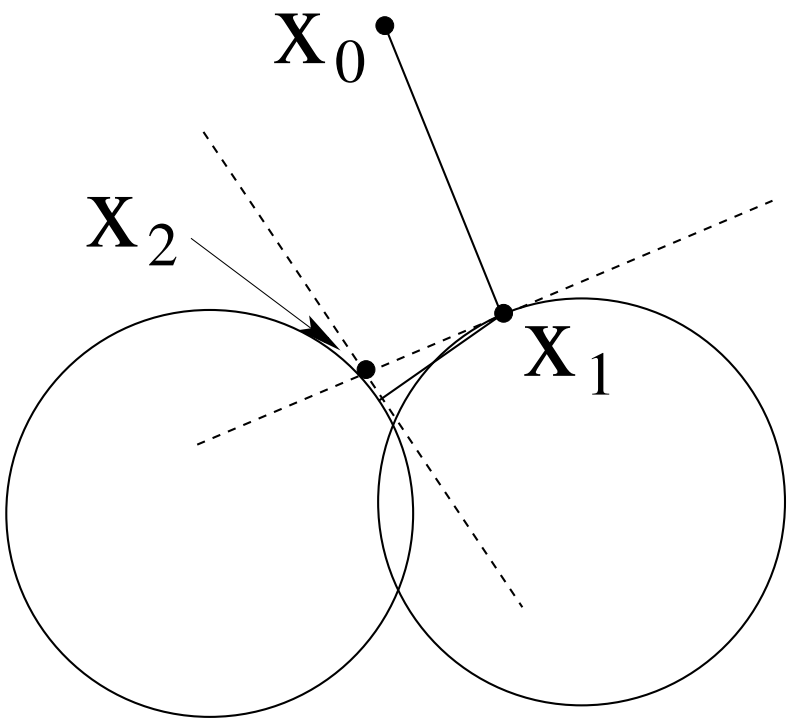
- `x`: Some x close to K (If BAP, $x = P_K(y)$).
- `status`: Indicates whether algorithm ends before or at iteration limit, or infeasible
- `C, b`: The matrices defining constraint $C^T x - b \geq 0$
- `q`: Number of active constraints
- `A`: The set `A(1:q)` is set of active constraints
- `r`: In infeasible case, gives Farkas lemma type multipliers certifying infeasibility of $C^T x - b \geq 0$.
- `u`: KKT multipliers (for BAP)
- `Q, R`: QR factorization of `C(:,A(1:q))`.

Performance of the algorithm



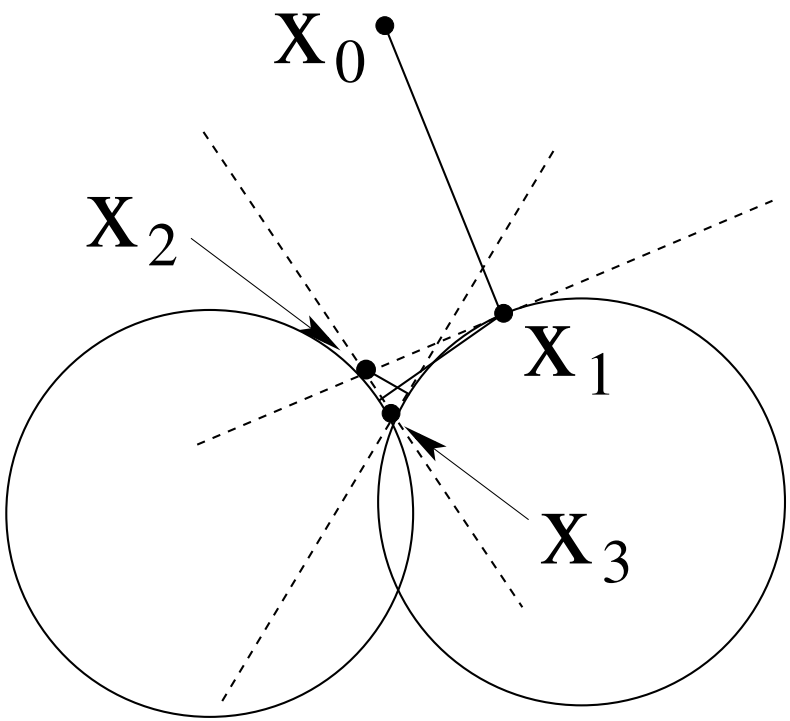
- Observations from projecting $(0, 10)$ onto $\mathbb{B}\big((1,0),3\big) \cap \mathbb{B}\big((-1,0),3\big) \subset \mathbb{R}^2$
  - If more than 2 supporting hyperplanes generated, SHDQP uses only the last two hyperplanes.
  - Fast convergence: 10 iterations to converge numerically

Performance of the algorithm



- Observations from projecting $(0, 10)$ onto $\mathbb{B}\big((1,0),3\big) \cap \mathbb{B}\big((-1,0),3\big) \subset \mathbb{R}^2$
  - If more than 2 supporting hyperplanes generated, SHDQP uses only the last two hyperplanes.
  - Fast convergence: 10 iterations to converge numerically

Performance of the algorithm



- Observations from projecting $(0, 10)$ onto $\mathbb{B}\big((1,0),3\big) \cap \mathbb{B}\big((-1,0),3\big) \subset \mathbb{R}^2$
  - If more than 2 supporting hyperplanes generated, SHDQP uses only the last two hyperplanes.
  - Fast convergence: 10 iterations to converge numerically

Numerical Performance for that small example

Iteration	x'	$\|x - (0, \sqrt{8})\|$
1	[+0.000000 +10.000000]	7.171573e+00
2	[+0.701489 +2.985112]	7.187744e-01
3	[-0.047520 +2.910211]	9.458681e-02
4	[+0.058764 +2.849630]	6.247192e-02
5	[+0.000483 +2.828652]	5.328667e-04
6	[+0.000558 +2.828624]	5.918623e-04
7	[-0.000000 +2.828427]	4.692090e-08
8	[+0.000000 +2.828427]	5.617550e-08
9	[+0.000000 +2.828427]	4.787408e-16
10	[+0.000000 +2.828427]	4.787408e-16

- Superlinear convergence to $(0, \sqrt{8})$.
  - Better than the linear convergence associated with projection-only algorithms, even for simple problems

Testing algorithms for the SIP

- The set of symmetric matrices in $\mathbb{R}^{n\times n}$ is an $\bar{n}$-dimensional space, where $\bar{n} = \frac{1}{2}n(n+1)$.
- Identify $\mathcal{S}^n$ with $\mathbb{R}^{\bar{n}}$.
- Consider the DNN cone $\mathcal{D}_n := \mathcal{S}^n_+ \cap \mathbb{R}^{\bar{n}}_+$.
  - In other words, $x \in \mathcal{D}_n$ iff x is symmetric positive definite with nonnegative entries
- We test the SHDQP algorithms for the SIP on $\mathcal{D}_n$.
  - Random starting matrix generated by randn()

```
*** Test 1:  SHDQP(SIP) with parameter 1 ***
Number of iterations:          5
Min value/ eigenvalue of output:  +1.1765e-03 +7.65679e-02
*** Test 2:  SHDQP(SIP) with parameter 0 ***
Number of iterations:          23
Min value/ eigenvalue of output:  +0.0000e+00 +2.94207e-16
*** Test 3:  Alternating reflections     ***
Number of iterations:          5
Min value/ eigenvalue of output:  +1.1765e-03 +7.65679e-02
*** Test 4:  Alternating Projections     ***
Number of iterations:          111
Min value/ eigenvalue of output:  +0.0000e+00 +1.44823e-17
*** Test 5:  Douglas-Rachford            ***
Number of iterations:          23
Min value/ eigenvalue of output:  +0.0000e+00 +6.90296e-14
```

- SHDQP(SIP) with parameter 1 and Alternating reflections same iterates, and perform best.
- Easy to construct examples where SHDQP outperforms alternating reflections.

Projecting onto the DNN Cone

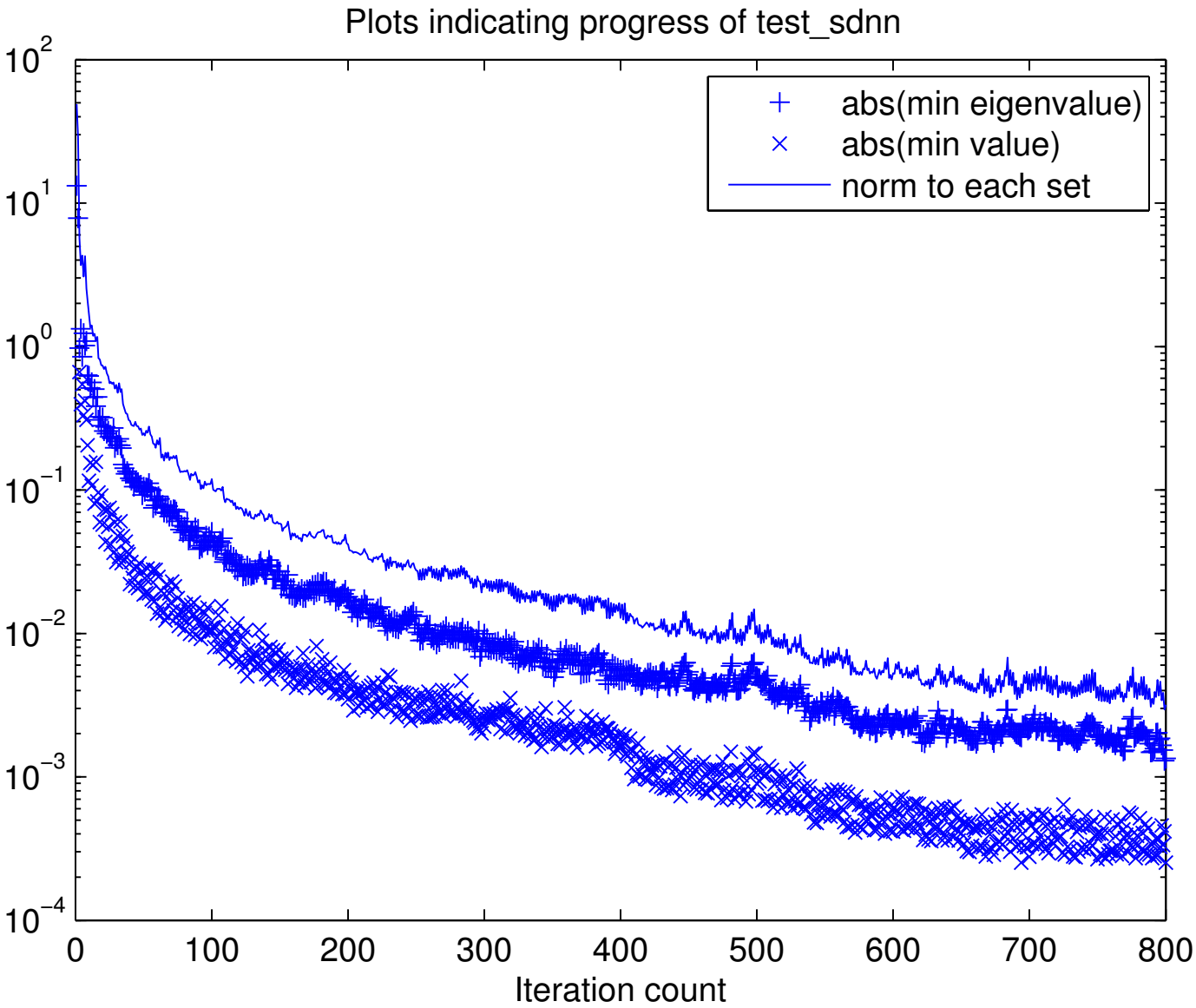
Next, consider the problem of finding $P_{\mathcal{D}_n}(y)$.

- $\mathcal{D}_n = \mathcal{S}_+^n \cap \mathbb{R}_+^{\bar{n}}$
- Naive projection: Treat $\mathbb{R}_+^{\bar{n}}$ as just another set
  - This strategy was used in SIP experiments.
  - Effective when few projections onto $\mathbb{R}_+^{\bar{n}}$ needed
- Constraint collection: Treat $\mathbb{R}_+^{\bar{n}}$ as $\bar{n}$ inequalities $(x_i \geq 0)$
  - May be better when many projections onto $\mathbb{R}_+^{\bar{n}}$ needed
    - Normals generated take on only $\bar{n}$ possibilities, and these possibilities are the elementary vectors
    - More in line with spirit of dual QP

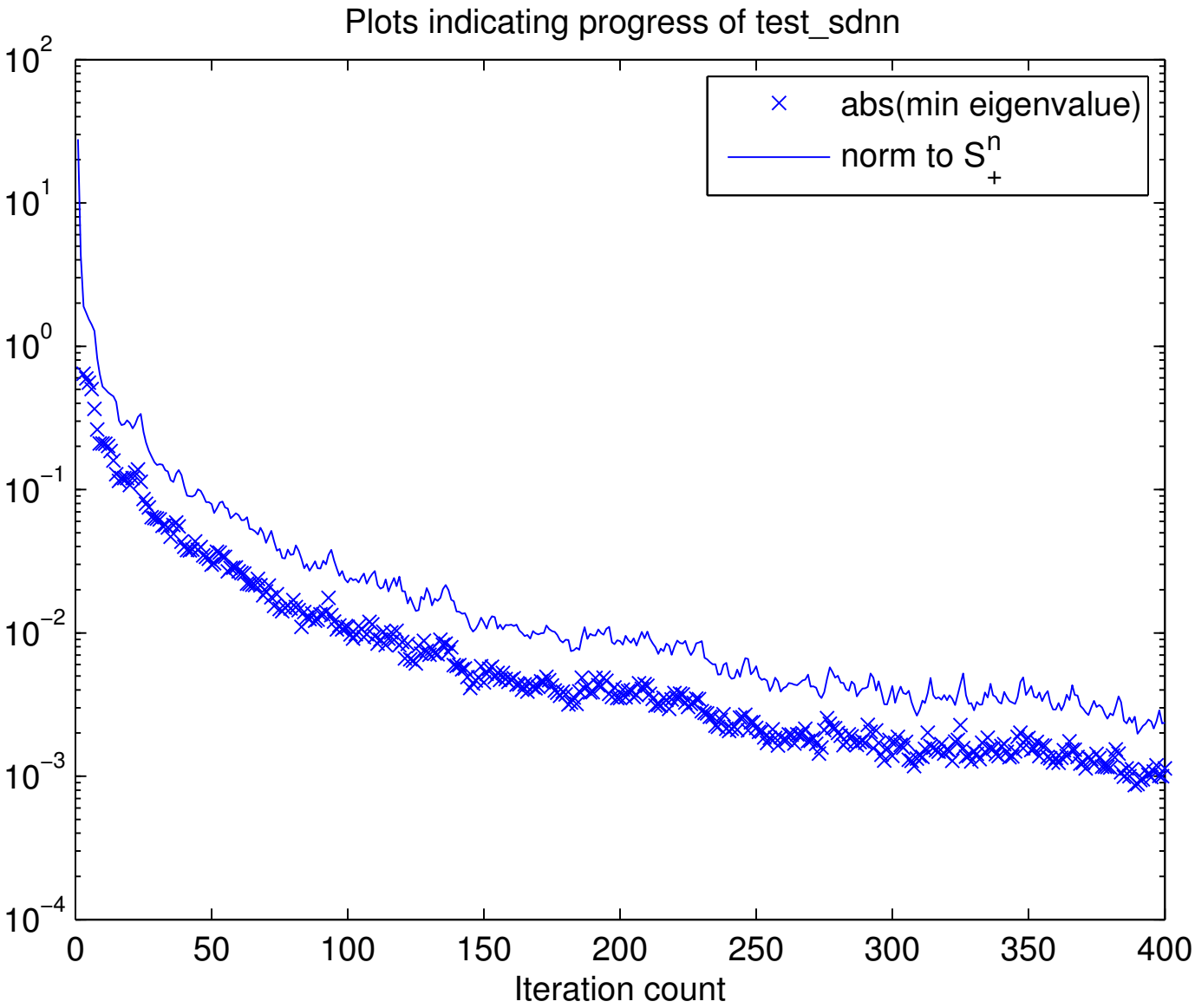
The convergence from y to $P_{\mathcal{D}_n}(y)$ is poor,

- Possible reason: $\mathcal{D}_n$ too nonsmooth at $P_{\mathcal{D}_n}(y)$.
- Consider problem of projecting a point onto a polyhedron.
  - Suppose m constraints tight at that optimum
    - If one constraint removed, then get different solution
  - Have to identify most constraints to get close to $P_{\mathcal{D}_n}(y)$.
- Contrast to fast convergence for projection onto $\mathbb{B}\big((1,0),3\big) \cap \mathbb{B}\big((-1,0),3\big) \subset \mathbb{R}^2$ earlier
- Apparently, the extra calculations in constraint collection strategy do not bring about much improvement.

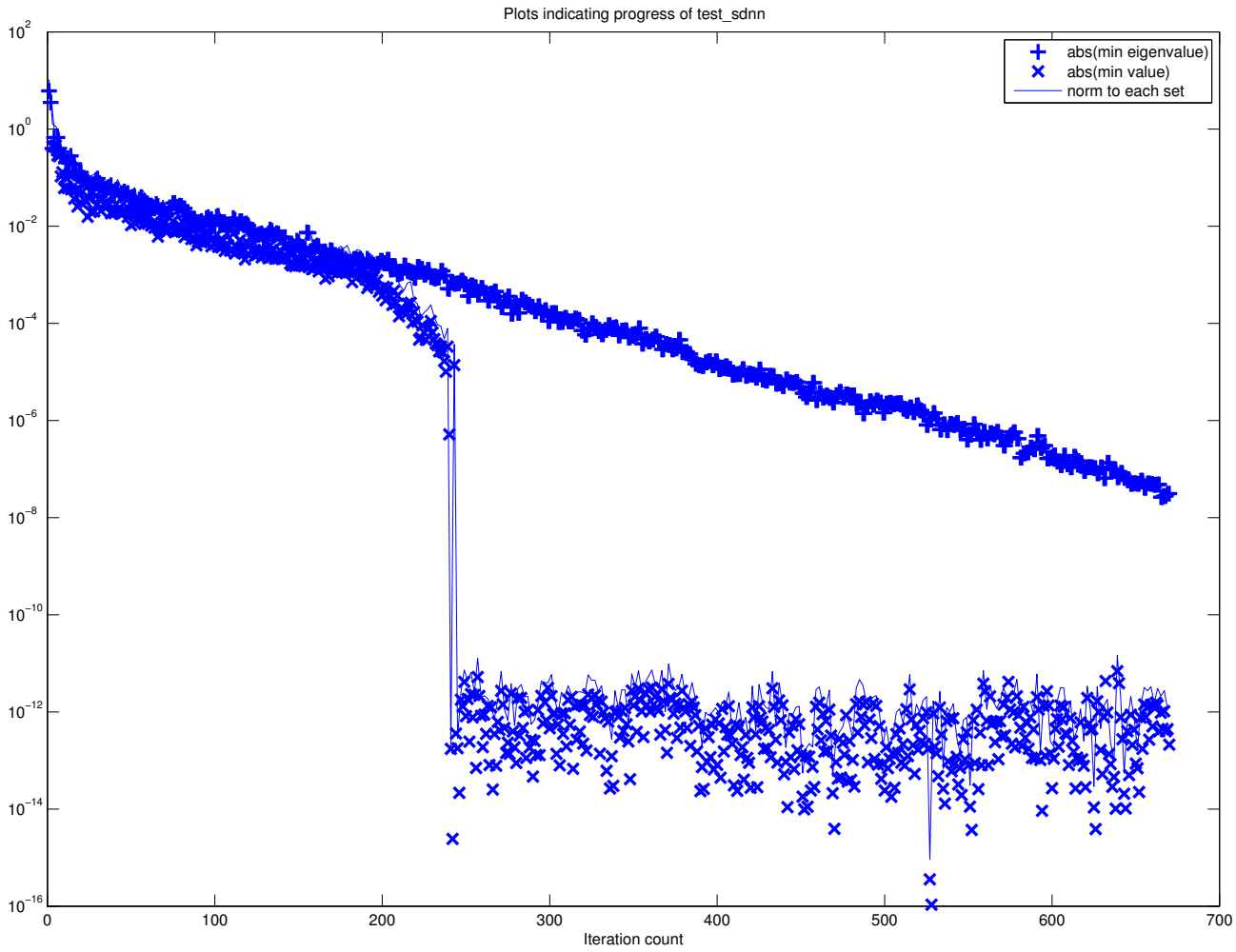
Naive projection method ($n = 100$)



Constraint collection strategy ($n = 100$)



Naive projection strategy ($n = 20$)



Projecting onto $\mathcal{S}_+^n$ harder than projecting onto $\mathbb{R}_+^{\bar{n}}$

Other topics:

- Overcoming ill-conditioning arising from algorithm implementation

Conclusions:

- For SIP (Finding a point $x \in K := \cap_{l=1}^{r} K_l$) and
  BAP (Find $P_K(y) = \arg\min_{x \in K} \|x - y\|^2$),
  projections/ reflections not the only trick.
  - Solve a QP partially by using inner GI steps from Goldfarb-Idnani's ('83) dual QP algorithm.
  - Better optimality and infeasibility certificates.
- Implemented a Matlab function that takes in projection functions and starting iterate to solve SIP and BAP.
- Reflection operations onto polyhedron generated by projection process seems very effective for the SIP.