A Streaming Algorithm for Computing an Approximate Minimum Spanning Ellipse

Asish Mukhopadhyay
University of Windsor
&
Eugene Greene
University of Waterloo

Minimum Spanning Ellipse

- Given n points $S = \{s_1, s_2, \ldots, s_n\}$ in the plane
- Find the minimum area ellipse containing S
- Can be solved exactly by
  - $O(n)$ average time randomized algorithm:
    E. Welzl, Smallest enclosing disks (balls and ellipsoids)
  - $O(n)$ time deterministic algorithm:
    Martin Dyer,
    A class of convex programs with applications to computational geometry

Streaming Model

- Only allowed *O(1)* space at any given time
- Only allowed *O(1)* time for each new point

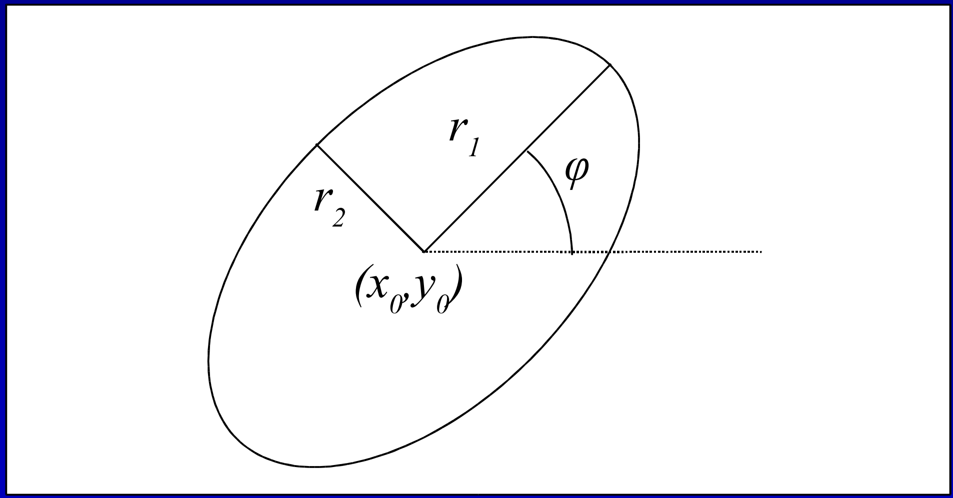
- Algorithm outline:
  - Store the current approximate ellipse E_i
  - Use the next point s_{i+1} and the current ellipse to generate the next approximate ellipse E_{i+1}

Minimum Spanning Ball

- Hamid Zarrabi-Zadeh, Timothy Chan
  A simple streaming algorithm for minimum enclosing balls
  - Each new ball is the smallest one containing the current ball and the new point
  - Requires *O(d)* time and *O(d)* space for each new point in the stream
  - Generates an approximate ball
    - $r_{Approx} \leq 3/2\ r_{Exact}$
- Can we extend this to ellipses?

Ellipses

- Center $p_0 = (x_0, y_0)$
- Semi-axis lengths r_1, r_2
- Angle φ



$$[p - p_0]^T A [p - p_0] = 1$$

$$[p - p_0]^T \begin{bmatrix} a & b \\ b & c \end{bmatrix} [p - p_0] = 1$$

$$a(x - x_0)^2 + 2b(x - x_0)(y - y_0) + c(y - y_0)^2 = 1$$

$$Area = \pi r_1 r_2 = \frac{\pi}{\sqrt{\det(A)}} = \frac{\pi}{\sqrt{ac - b^2}}$$

Algorithm

- $s_1 \rightarrow E_1 = \text{point}(s_1)$
- $s_2 \rightarrow E_2 = \text{segment}(s_1, s_2)$
- $s_3 \rightarrow E_3 = \text{ellipse}(s_1, s_2, s_3)$
- $s_4 \rightarrow E_4 =$ smallest ellipse containing E_3, s_4
- …
- $s_{i+1} \rightarrow E_{i+1} =$ smallest ellipse containing E_i, s_{i+1}

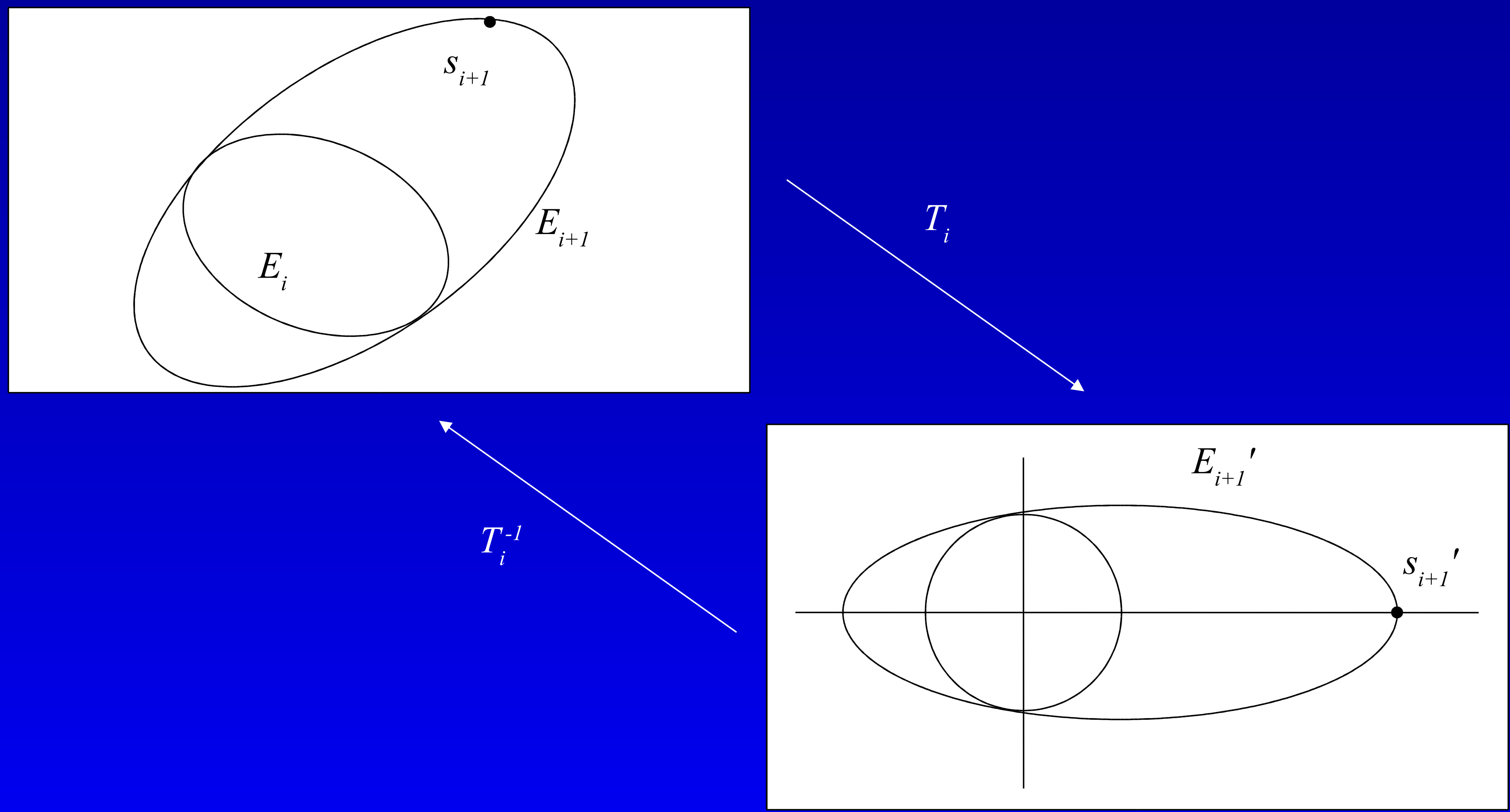
Smallest Ellipse of Point & Ellipse

- E_{i+1} = smallest ellipse containing E_i & s_{i+1}
- Find a transformation T_i mapping E_i to unit circle
- T_i must satisfy:

  1) $T_i(E)$ is an ellipse, $T_i^{-1}(E)$ is an ellipse

  2) $area(E_1) \leq area(E_2) \Leftrightarrow area(T_i(E_1)) \leq area(T_i(E_2))$
- $s_{i+1}' := T_i(s_{i+1})$
- $E_{i+1}' :=$ smallest ellipse containing s_{i+1}', unit circle
- Then $E_{i+1} = T_i^{-1}(E_{i+1}')$

Smallest Ellipse of Point & Ellipse



Transformation

- A given translation, rotation, scaling
  - Maps ellipses to ellipses
  - Multiplies ellipse areas by constant factor

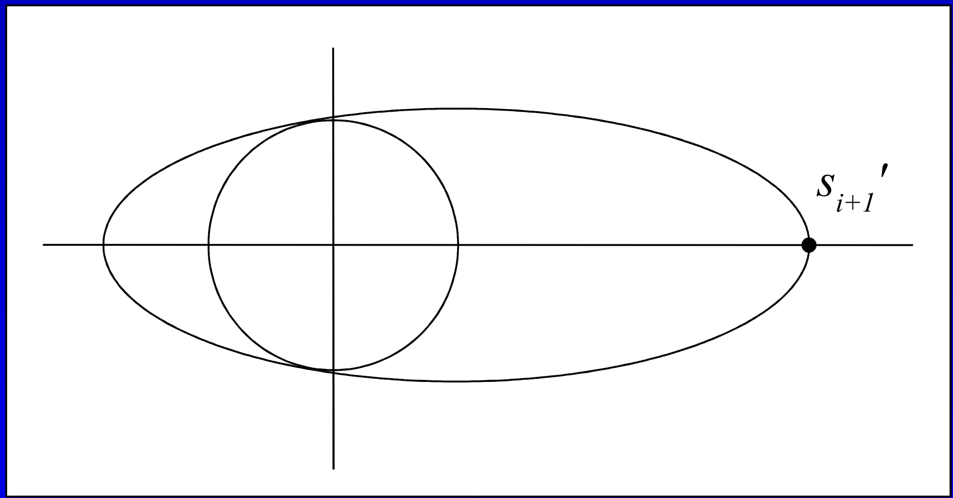
- $T_i(p) = S^y(1/r_{2i})\ S^x(1/r_{1i})\ R(-\varphi_i)\ (p-p_0)$

- Add another rotation so $s_{i+1}' = T_i(s_{i+1})$ is on positive x-axis

Smallest Ellipse of Point & Unit Circle

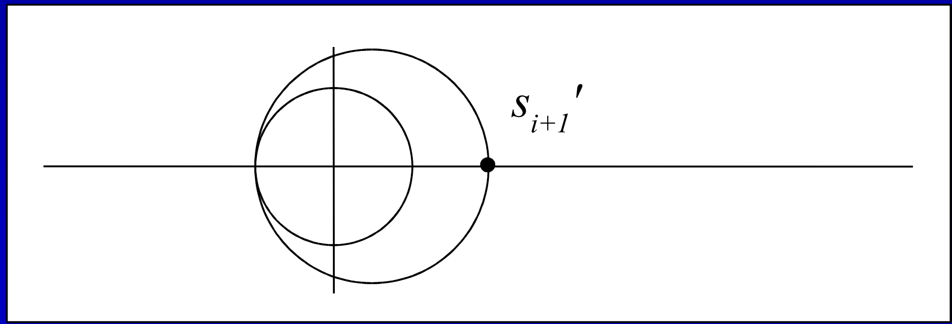
- E_{i+1}' =smallest ellipse containing unit circle and point *(d,0)* on positive *x*-axis

- E_{i+1}' must
  - have an axis on *x*-axis
  - contain the unit circle
  - be tangent to the unit circle
  - pass through s_{i+1}'

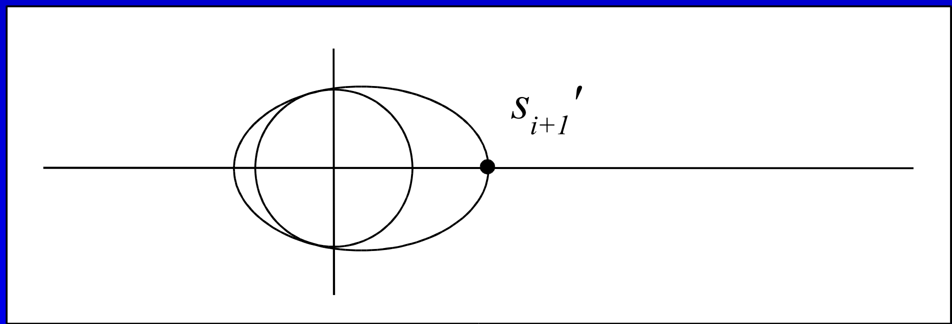


Smallest Ellipse of Point & Unit Circle

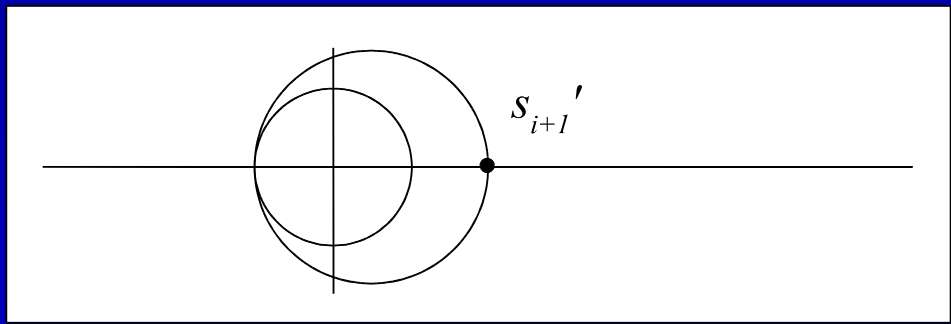
- C_{i+1} := smallest circle containing unit circle and s_{i+1}'

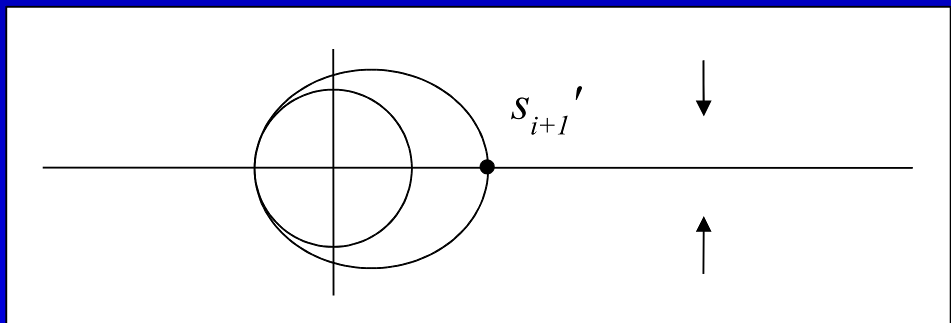


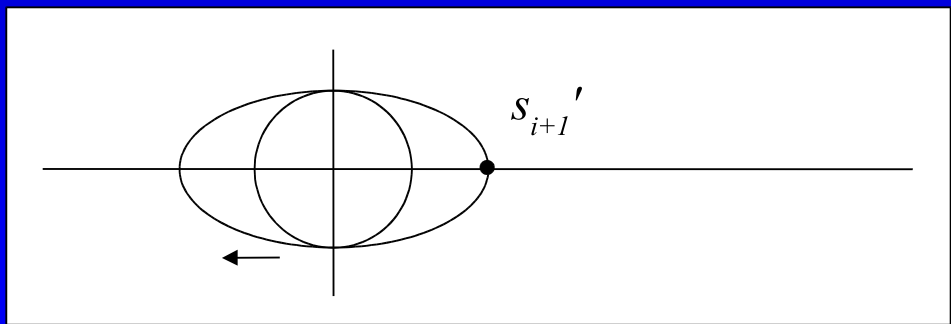
- Scale C_{i+1} into C_{i+1}', maintaining previous properties



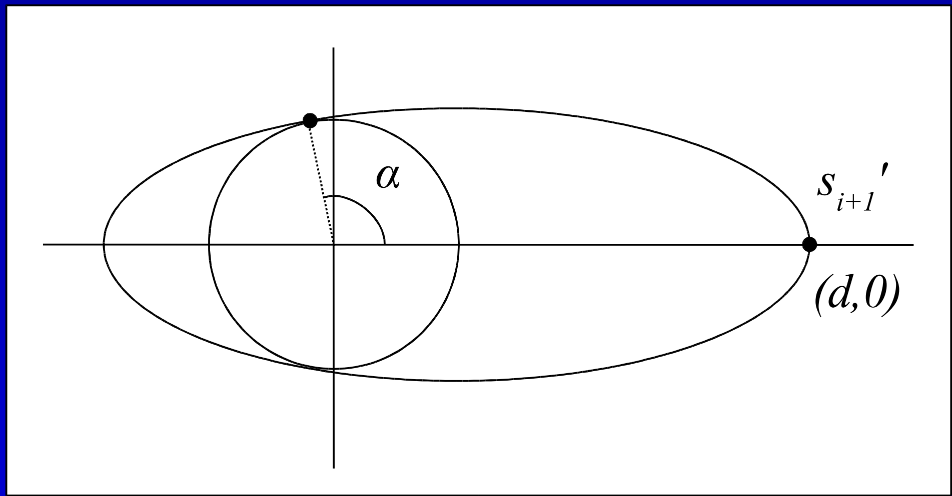
Smallest Ellipse of Point & Unit Circle

- $C_{i+1}'(t)$
  - $t = t_0$ 

    $d+1)/2)^2$
  - $t = t_1$ 

    $(d+1)/2)^{3/2}$
  - $t = t_2$ 

Smallest Ellipse of Point & Unit Circle



- *min Area(α)* → $\beta = \cos\alpha = \dfrac{d - \sqrt{d^2 + 8}}{4}$

Smallest Ellipse of Point & Unit Circle

$$x_0 = \frac{d^2 - 1}{2d - \beta - \beta^{-1}} \qquad a = \frac{1}{(\beta - x_0)(\beta^{-1} - x_0)} \qquad c = \frac{1}{1 - \beta x_0}$$

- $E_{i+1}' : \quad a(x - x_0)^2 + cy^2 = 1$
- $E_{i+1} = T_i^{-1}(E_{i+1}')$

Approximation Ratio

- Assume $ApproxEllipse \leq k\ ApproxDisk$

$$ApproxEllipse \leq k \frac{9}{4} MinDisk$$

$$\leq k \frac{9}{4} MinEllipse \frac{1}{\sqrt{1-e^2}}$$

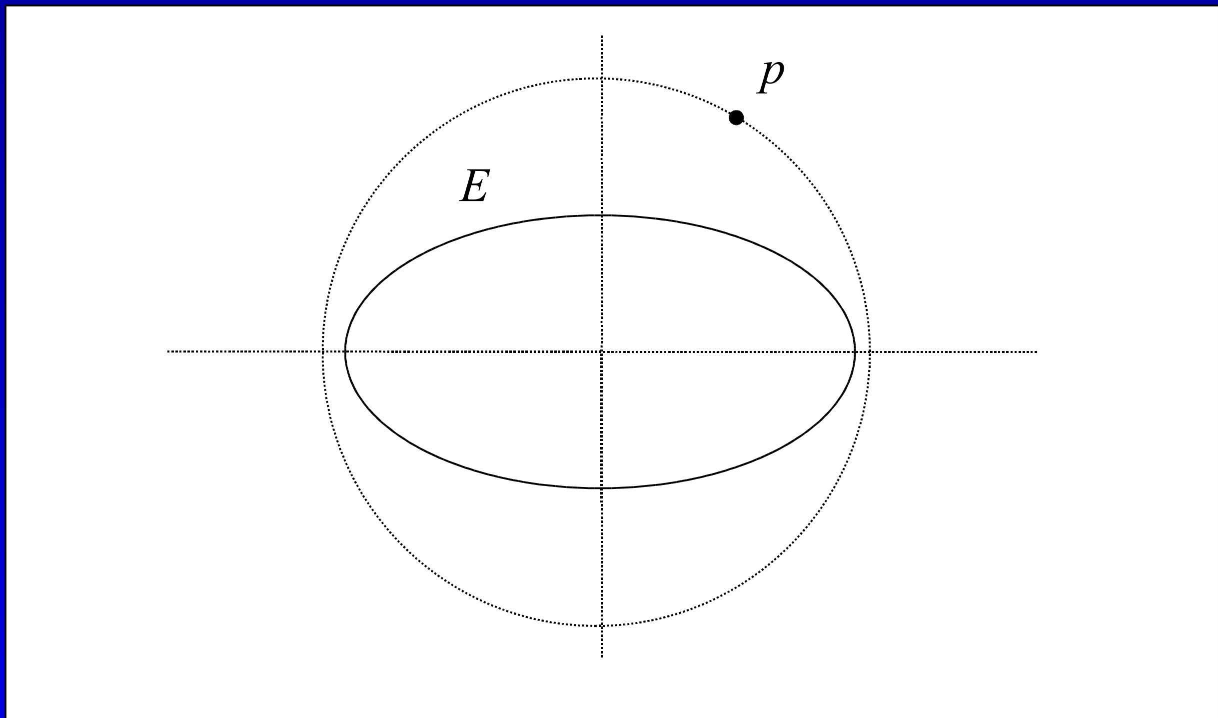
$$ApproxEllipse \leq \frac{9k}{4\sqrt{1-e^2}} MinEllipse$$

Approximation Ratio

- Given some E_{exact}, assume it is the exact minimum spanning ellipse
- What sequence of points results in largest E_{approx}?
- If such a sequence exists, then E_{approx} / E_{exact} is an approximation ratio for the algorithm
- We can use E_{exact} = unit circle

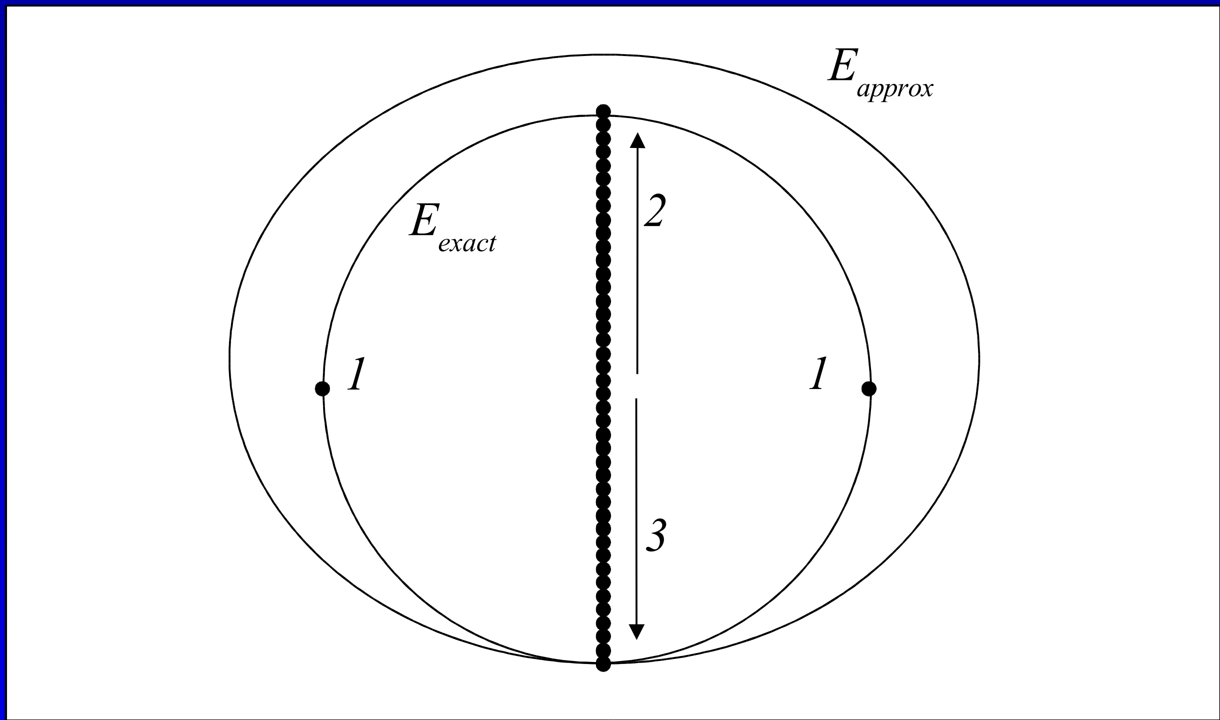
Approximation Ratio

- Consider some ellipse E, and some fixed distance d



- To maximize E' – smallest ellipse containing E, p
  - p must be on supporting line of minor axis of E

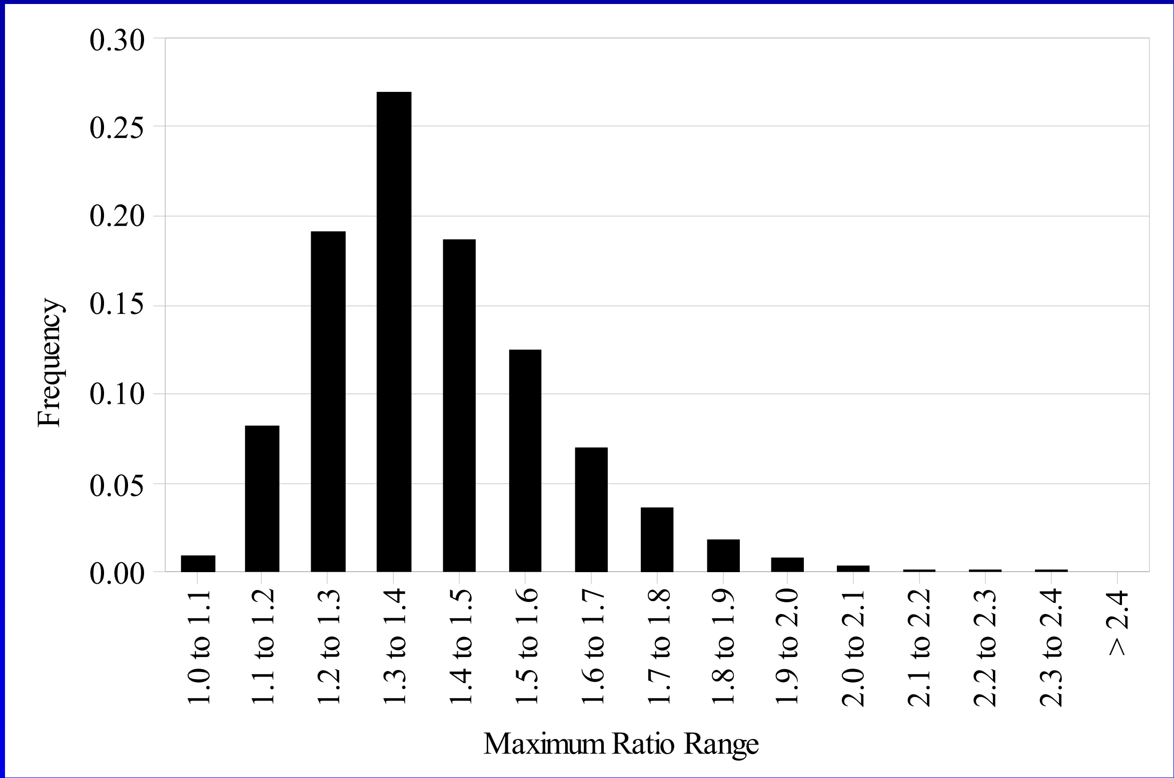
An Undesirable Point Sequence



An Undesirable Point Sequence

# Points	Ratio
2×10^2	3.082697164225789
2×10^3	4.418103486895557
2×10^4	5.14471378862369
2×10^5	5.267371718841295
2×10^6	5.280568308868795
2×10^7	5.281897415706886
2×10^8	5.2820305804653405
2×10^9	5.28204328082953

Experimental Results



Higher Dimensions

- First D points will result in exact, degenerate, ellipsoids
- $(D+1)$th ellipsoid will be exact, non-degenerate
- D-dimensional ellipse E_i can be transformed to D-dimensional unit ball, and s_{i+1} rotated onto x_1-axis
- E_{i+1}' must be symmetric with respect to x_1-axis

Higher Dimensions

- E_{i+1}':

$$\begin{bmatrix} x_1 - x_0 & x_2 & \cdots & x_D \end{bmatrix} \begin{bmatrix} a & 0 & \cdots & 0 \\ 0 & c & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & c \end{bmatrix} \begin{bmatrix} x_1 - x_0 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} = 1$$

- Algorithm: $\left[p - p_{0i}\right]^T A_i \left[p - p_{0i}\right] = 1$

$$A_i = H_i^T H_i$$

Higher Dimensions

1. $s_{i+1}'' \leftarrow H_i\,[s_{i+1} - p_{0i}]$

2. $R_{i+1} \leftarrow$ matrix that rotates s_{i+1}'' onto positive x_1-axis

3. Find d, x_0, a, c

$$p_{0i+1}' \leftarrow (x_0, 0, \ldots, 0);\; S_{i+1} \leftarrow \begin{bmatrix} \sqrt{a} & 0 & \cdots & 0 \\ 0 & \sqrt{c} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sqrt{c} \end{bmatrix}$$

4. Solve for p_{0i+1} in $R_{i+1}H_i[p_{0i+1} - p_{0i}] = p_{0i+1}'$

5. $H_{i+1} \leftarrow S_{i+1}R_{i+1}H_i$

Conclusions & Further Work

- Approximate planar minimum spanning ellipse
  - $O(1)$ space
  - $O(1)$ time for each new point
- Works nicely for random point sequences
- Extends to higher dimensions
- Approximation ratio?