Diagram of Reality and Desire

- $\alpha$ is the reality
- $\beta$ is what is desired

- The diagram $RD_\beta(\alpha)$ is the diagram of reality and desire:
Diagram of Reality and Desire
(for better visualization)

- Extended $\alpha$: 0, +3, −2, −1, +4, −5, 6
- Extended $\beta$: 0, +1, +2, +3, +4, +5, 6

Saad Mneimneh
http://engr.smu.edu/~saad/
Definition

Two reality edges on the same cycle converge if they are traversed in the same direction (either clockwise or counterclockwise) on the cycle.

Example:

(+3,+2) and (-1,-4) converge

(0,-3) and (+3,+2) diverge
Reversals and $c(\alpha)$

Let $\rho$ be a reversal defined by two reality edges $e$ and $f$, then:

- If $e$ and $f$ belong to different cycles, $c(\alpha\rho) = c(\alpha) - 1$

- If $e$ and $f$ belong to the same cycle and converge, $c(\alpha\rho) = c(\alpha)$

- If $e$ and $f$ belong to the same cycle and diverge, $c(\alpha\rho) = c(\alpha) + 1$
Another lower bound

- Let \( \rho_1, \ldots, \rho_t \) be such that \( \alpha \rho_1 \ldots \rho_t = \beta \), then:

\[
\begin{align*}
    c(\alpha \rho_1) - c(\alpha) &\leq 1 \\
    c(\alpha \rho_1 \rho_2) - c(\alpha \rho_1) &\leq 1 \\
    \vdots \\
    c(\alpha \rho_1 \ldots \rho_t) - c(\alpha \rho_1 \ldots \rho_{t-1}) &\leq 1 \\
\end{align*}
\]

\[
c(\alpha \rho_1 \ldots \rho_t) - c(\alpha) = n + 1 - c(\alpha) \leq t
\]

- Let \( t = d(\alpha) \), then

\[
d(\alpha) \geq n + 1 - c(\alpha)
\]
Better lower bound

- The lower bound $n + 1 - c(\alpha)$ is better than $b(\alpha)/2$.

- For most signed permutations, it comes very close to the actual reversal distance.

- It does not always work. We cannot always choose two divergent edges (doing so will increase the number of cycles by 1 each time and achieve the exact bound).

- To understand why, we need to define some concepts.
Good / Bad cycles

The easy stuff first....

• A cycle is **good** iff it has two reality edges that diverge

• A cycle is **bad** iff all its reality edges converge

• A cycle is **proper** iff it contains more than two edges (we will only look at these).
Interleaving

- Assume the special way of drawing $RD(\alpha)$
  - either the counterclockwise circle where reality edges are on the circumference and desire edges are chords
  - or reality edges are on the line and desire edges on top
- Two cycles interleave iff a desire edge of one crosses a desire edge of the other
- In general we can define this using intervals without assuming a special way of drawing $RD(\alpha)$
Interleaving

A and E interleave
B and C interleave
C and D interleave
B and D interleave
Components

We have three components:

\{F\}

\{A, E\}

\{B, C, D\}
Good / Bad components

We have three components:

\{F\} : good

\{A, E\} : bad

\{B, C, D\} : good
Classification of bad components

Bad Components
  ├── Non-Hurdles
  │     └── Simple Hurdles
  └── Hurdles
     └── Super Hurdles
Hurdles

A bad component is a hurdle iff it does not separate between any two components
Super Hurdles

$F$ is a super hurdle, it protects $E$

All other hurdles are Simple hurdles
Fortress

A signed permutation $\alpha$ is called a fortress iff $RD(\alpha)$ has an odd number of hurdles and all of them are super hurdles

The smallest fortress, with 3 super hurdles
New lower bound

\[ d(\alpha) \geq n + 1 - c(\alpha) + h(\alpha) \]

Any reversal \( \rho \) acts on two reality edges, and can “destroy” at most two hurdles (by definition, hurdles cannot separate between hurdles), therefore

\[ \Delta h = h(\alpha \rho) - h(\alpha) \geq -2 \]

three cases:

- \( \Delta c = 1 \), then \( \rho \) acts on a good cycle and \( \Delta h = 0 \) (no hurdles destroyed), \( \Delta(c - h) = 1 \)

- \( \Delta c = 0 \), then \( \rho \) acts on a bad cycle and \( \Delta h \geq -1 \) (at most one hurdle destroyed), \( \Delta(c - h) \leq 1 \)

- \( \Delta c = -1 \), then \( \Delta h \geq -2 \) anyway (at most two hurdles destroyed), \( \Delta(c - h) \leq 1 \)
Exact traversal distance

\[ d(\alpha) = n + 1 - c(\alpha) + h(\alpha), \ \alpha \text{ non-fortress} \]

\[ d(\alpha) = n + 1 - c(\alpha) + h(\alpha) + 1, \ \alpha \text{ fortress} \]

where \( h(\alpha) \) is the number of hurdles
Approach

- We will achieve the lower bound \( n + 1 - c(\alpha) + h(\alpha) \) for a non-fortress permutation \( \alpha \).

- Each time we will find a reversal \( \rho \) such that \( c(\alpha \rho) - h(\alpha \rho) = c(\alpha) - h(\alpha) + 1 \).

- Call such a reversal a safe reversal.

- \( c(\alpha) - h(\alpha) \) can be at most \( n+1 \) (when \( \alpha = \beta \)); therefore, we will perform only \( n+1 - c(\alpha) + h(\alpha) \) reversals.
Safe reversal Kind I

- A reversal $\rho$ defined by two divergent reality edges of a good cycle that does not lead to the creation of bad components.

- Safe: increases $c$ by 1, keeps $h$ unchanged; therefore, $c(\alpha \rho) - h(\alpha \rho) = c(\alpha) - h(\alpha) + 1$.

- Fact: if we have a good component, then there exists a safe reversal of kind I (we are not going to prove this).
Safe reversal Kind II
(hurdle merging)

- Define two opposite hurdles $A$ and $B$ such that the number of hurdles between $A$ and $B$ is the same on either sides of the circle.

- A reversal $\rho$ defined by two reality edges of two opposite hurdles (i.e. the number of hurdles must be even)

- $\rho$ destroys two hurdles which become a good component with any non-hurdle that separates them.

- $\rho$ does not create new hurdles.

- Safe: decreases $c$ by 1, decreases $h$ by 2; therefore, $c(\alpha \rho) - h(\alpha \rho) = c(\alpha) - h(\alpha) + 1$
Illustration

This becomes good component

This cannot become a hurdle

Saad Mneimneh
http://engr.smu.edu/~saad/
Safe reversal Kind III (hurdle cutting)

• A reversal $\rho$ defined by two convergent reality edges of a bad cycle in a simple hurdle when number of hurdles is odd.

• $\rho$ destroys the hurdle (makes it good component), and does not create new hurdles (otherwise it is a super hurdle), and results in even number of hurdles.

• Safe: keeps $c$ unchanged, decreases $h$ by 1; therefore, $c(\alpha\rho) - h(\alpha\rho) = c(\alpha) - h(\alpha) + 1$
Non-fortress

• If $\alpha$ is not a fortress, then we can always find a safe reversal

• Proof:
  – If there is a good component, then there is a kind I safe reversal.

  – If there are no good components and the number of hurdles is even, then there is a Kind II safe reversal (hurdle merging)

  – If there are no good components and the number of hurdles is odd, then there is a simple hurdle (otherwise $\alpha$ is a fortress), and there is a Kind III safe reversal (hurdle cutting). After that point the number of hurdles is always even.
Algorithm

given distinct $\alpha$ and $\beta$

repeat
  if there is a good component in $RD_\beta(\alpha)$
  then pick a Kind I reversal

  else if $h(\alpha)$ is even
  then pick a Kind II reversal (merging two opposite hurdles)

  else if $h(\alpha)$ is odd
  then pick a Kind III reversal (cutting simple hurdle)

until $\alpha = \beta$
What if $\alpha$ is a fortress

- Then we have an odd number of hurdles non of which is simple
  - We cannot cut a super hurdle since this will create a new hurdle
  - We can merge two super hurdles (not opposite), but there is a danger of creating a new hurdle!

- FACT: merging two super hurdles in a fortress with $h \neq 3$ super hurdles results in a fortress with $h - 2$ super hurdles.

- Therefore, we create a new hurdle only when we have a forest with 3 super hurdles resulting is a distance of $n + 1 - c(\alpha) + h(\alpha) + 1$ for a fortress permutation.

- This is optimal for a fortress because any fortress has to have one unsafe reversal, but we will not prove it here.
Algorithm

given distinct $\alpha$ and $\beta$

repeat
  if there is a good component in $RD_\beta(\alpha)$
    then pick a Kind I reversal
  else if $h(\alpha)$ is even
    then pick a Kind II reversal (merging two opposite hurdles)
  else if $h(\alpha)$ is odd and there is a simple hurdle
    then pick a Kind III reversal (cutting simple hurdle)
  else //fortress
    merge any two hurdles (this will result in a fortress if $h(\alpha) \neq 3$)

until $\alpha = \beta$
Running time
(initialization of structure)

- Constructing $RD(\alpha)$ takes $O(n)$ time (determine reality and desire edges)
- Finding cycles can be done in $O(n)$ time
- For each cycle we have to determine whether it is good or bad, this takes $O(n)$ time for each cycle for a total of $O(n^2)$ time
- Interleaving of cycles can be done in $O(n^2)$ time by examining every pair of desire edges
- Determining good and bad components, non-hurdles, simple hurdles, and super hurdles can be done in $O(n)$ time
- So far $O(n^2)$ time
Running time (cont.)

• The most time consuming part is identifying whether a safe reversal of Kind I exists

• Since a reversal is defined by two edges, we have $O(n^2)$ reversals to try

• For each reversal we have to see whether a bad component will be created (this can be done in $O(n^2)$ time as discussed above by computing $RD(\alpha \rho)$)

• Therefore, we spend $O(n^4)$ time

• This is performed at most $n$ times yielding an $O(n^5)$ time algorithm