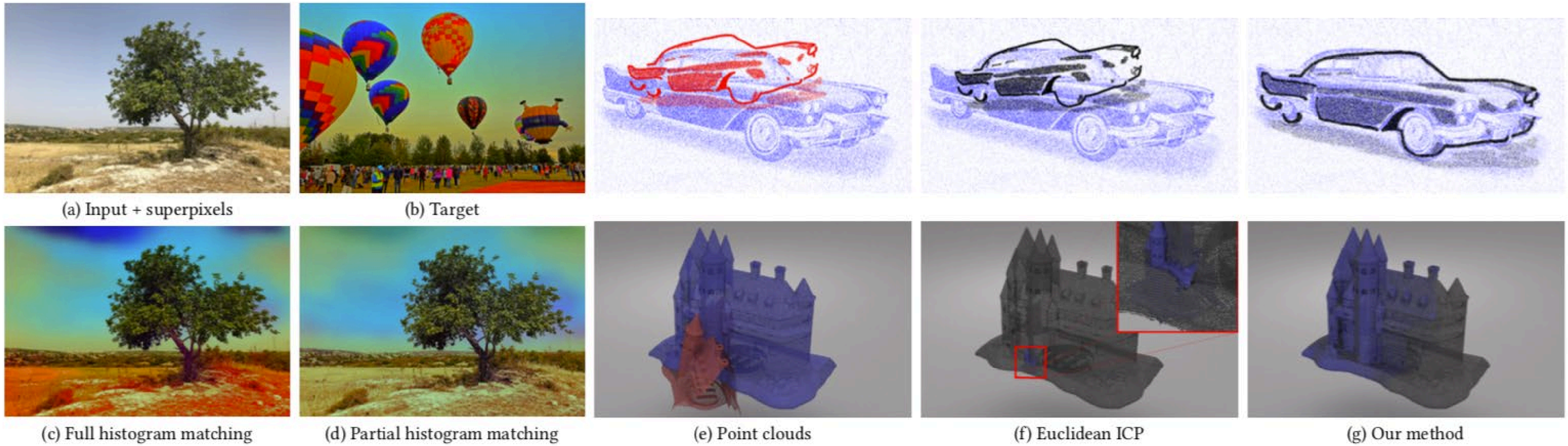
Sliced Partial Optimal Transport

Nicolas Bonneel*, David Coeurjolly*



(a) Input + superpixels (b) Target

(c) Full histogram matching (d) Partial histogram matching (e) Point clouds (f) Euclidean ICP (g) Our method

***CNRS, Univ. Lyon.**

Matching points

- Linear Assignment Problem

$$\min_{\text{T bijective}} \sum_i c(x_i, y_{T(i)})$$

- Optimal transport

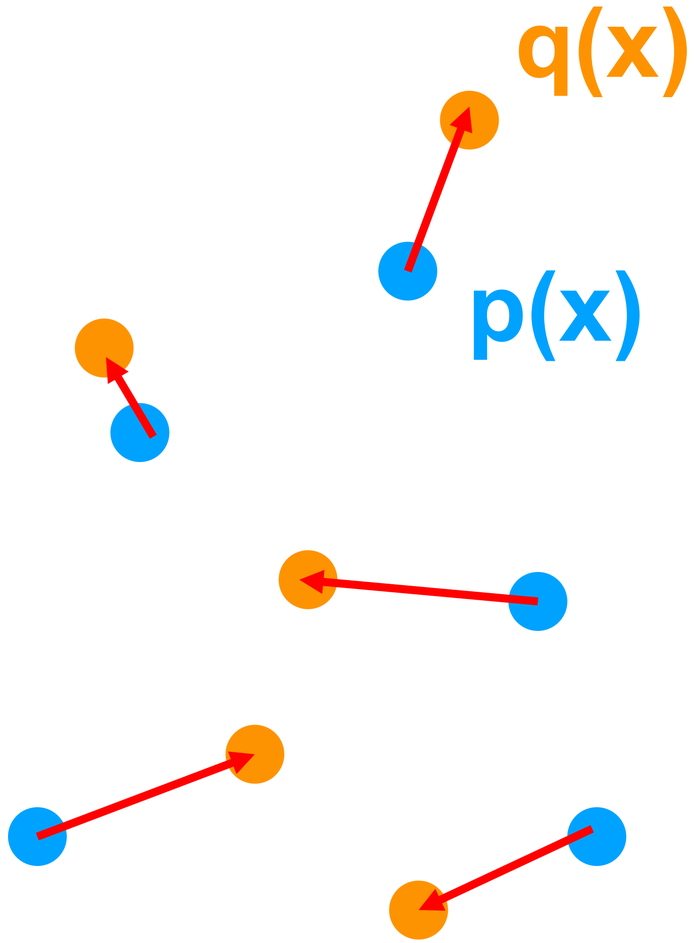
$$W(f,g) = \min \sum_{i,j} c(x_i, y_j)\pi_{i,j}$$

s.t.

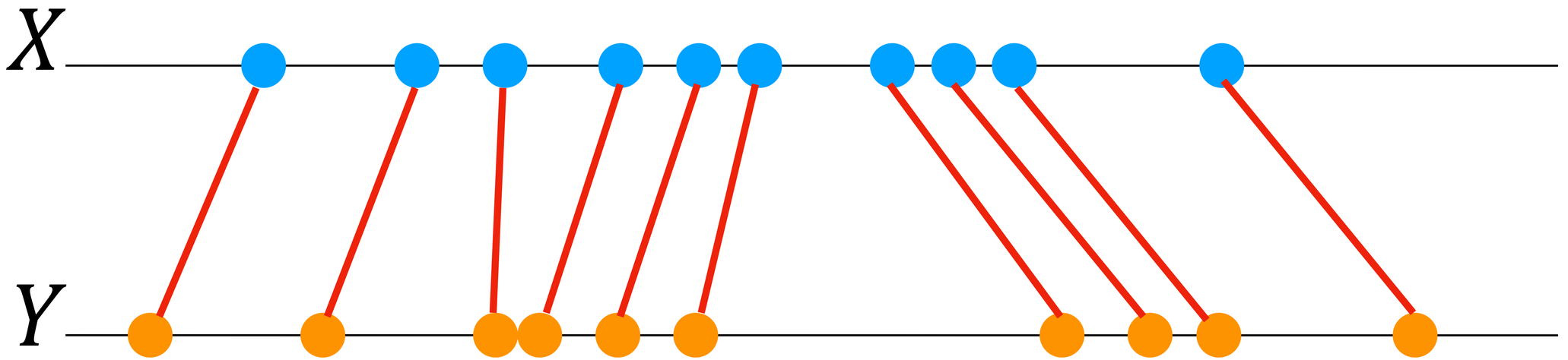
$$\sum_j \pi_{i,j} = 1$$

$$\sum_i \pi_{i,j} = 1$$

$$\pi_{i,j} \geq 0$$

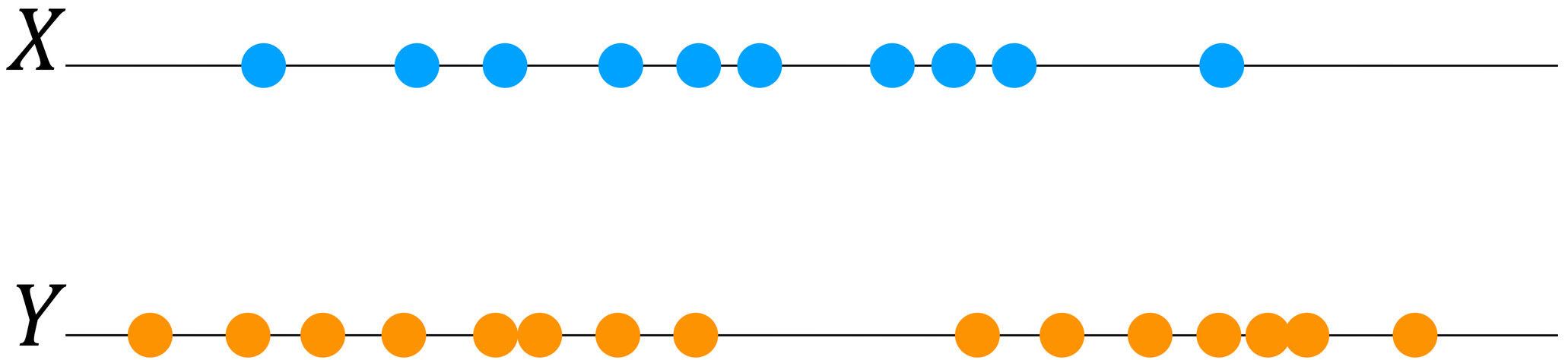


1-d Linear Assignment Problem is trivial*



*assuming the cost c is a convex function of |x-y|

Partial optimal assignment ?



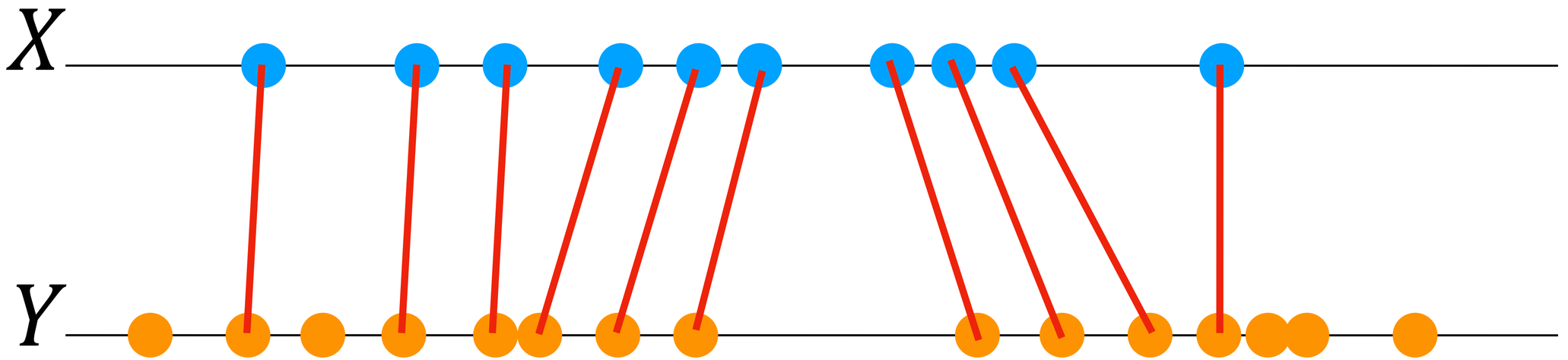
$$W(f,g) = \min \sum_{i,j} c(x_i, y_j)\pi_{i,j} \quad \textbf{s.t.} \quad \sum_j \pi_{i,j} = 1$$

$$\sum_i \pi_{i,j} \leq 1$$

$$\pi_{i,j} \geq 0$$

$$\min_{T \text{ injective}} \sum_i c(x_i, y_{T(i)})$$

Partial optimal assignment ?

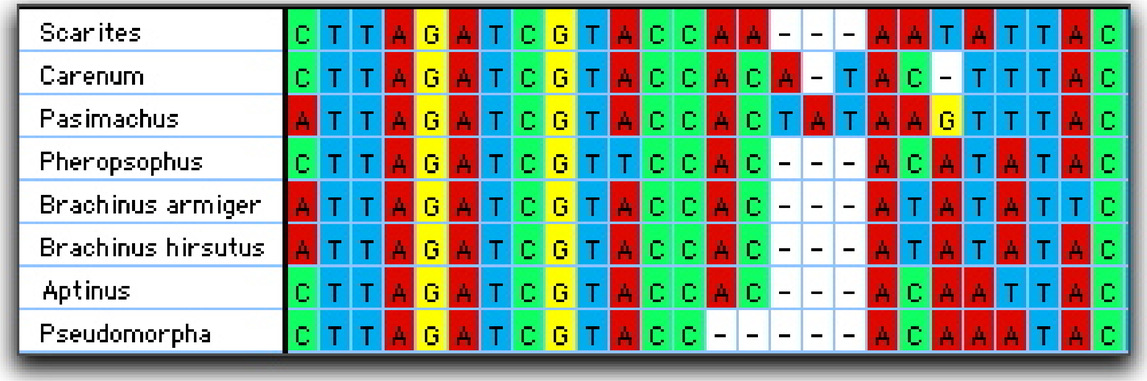


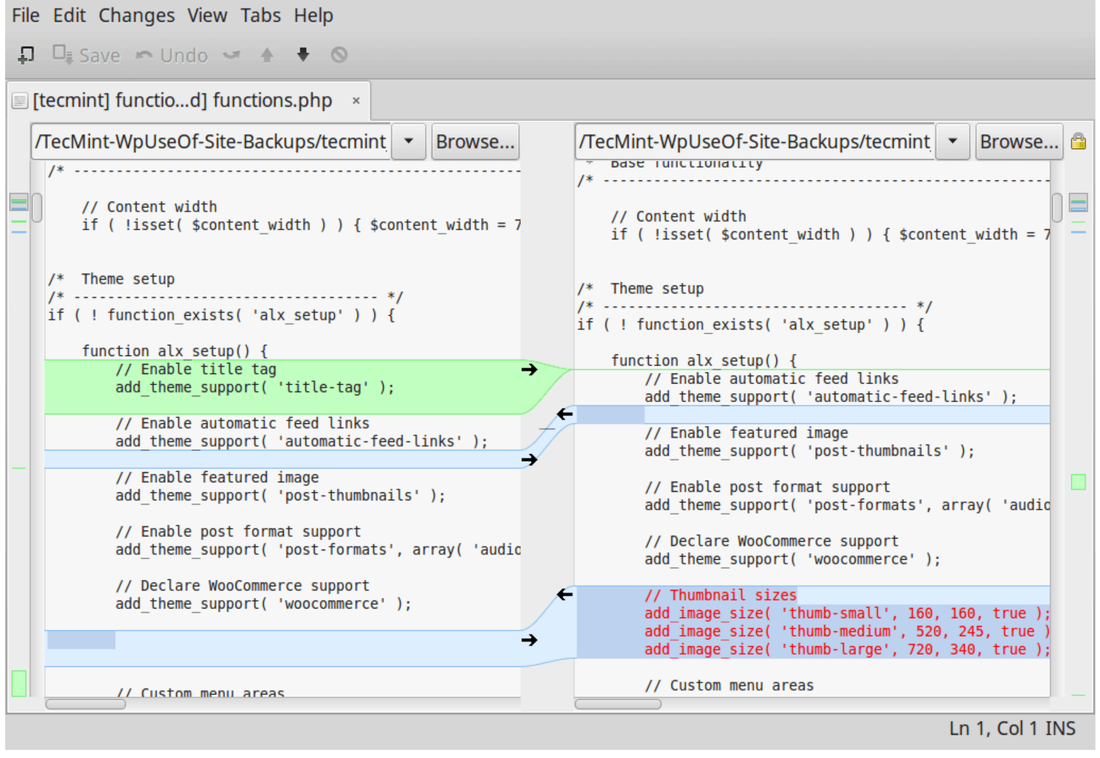
$$W(f,g) = \min \sum_{i,j} c_{i,j}\pi_{i,j} \quad \textbf{s.t.} \quad \sum_j \pi_{i,j} = 1 \qquad \sum_i \pi_{i,j} \leq 1 \qquad \pi_{i,j} \geq 0$$

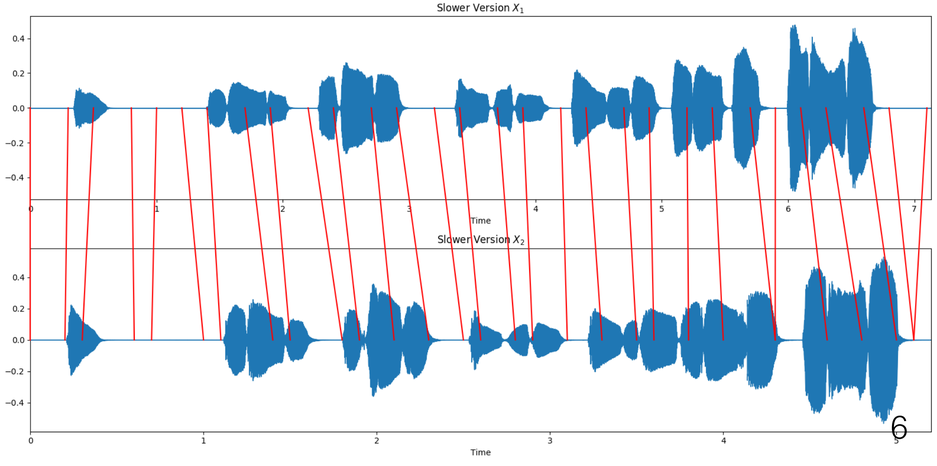
$$\min_{T \text{ injective}} \sum_i c(x_i, y_{T(i)})$$

Similar problems

- DNA sequence alignment
- Text alignment
  - Levenshtein distance
- Music synchronization
- …

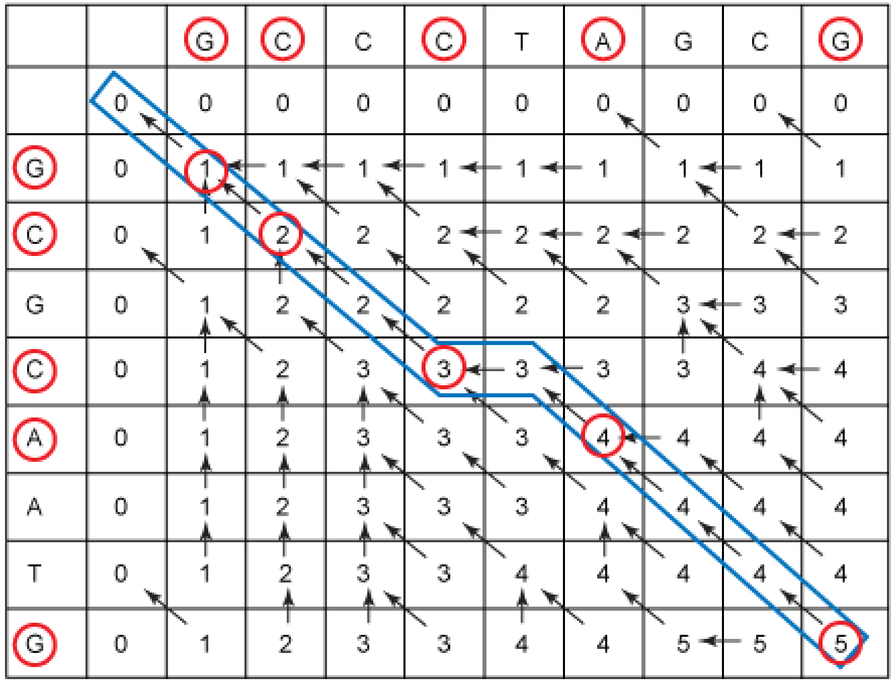






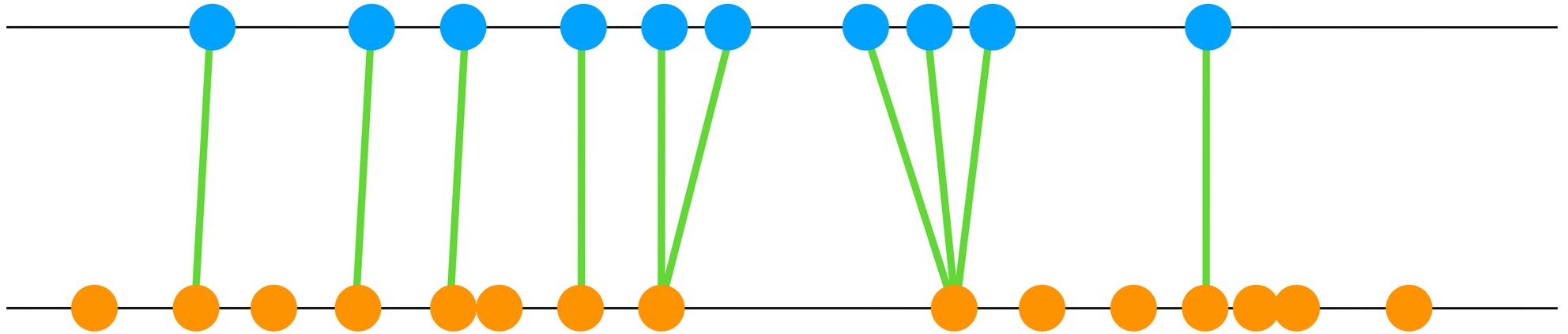
Existing Solutions

- Dynamic Time Warping
  - Solves a dynamic programming problem
  - Smith–Waterman algorithm, Needleman–Wunsch algorithm O(N^2) space and time
  - Hirschberg's algorithm O(N^2) time, O(N) space
- All end up doing variants of
  - $A_{i,j} = \min(A_{i-1,j-1} + cost, A_{i-1,j} + cost', A_{i,j-1} + cost'')$



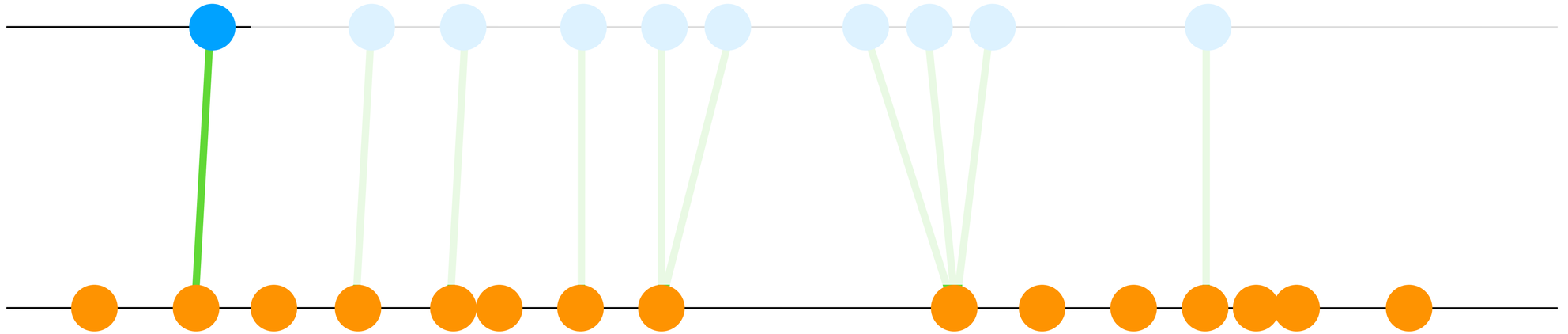
Quadratic time complexity algorithm (linear space)

Quadratic time complexity algorithm (linear space)



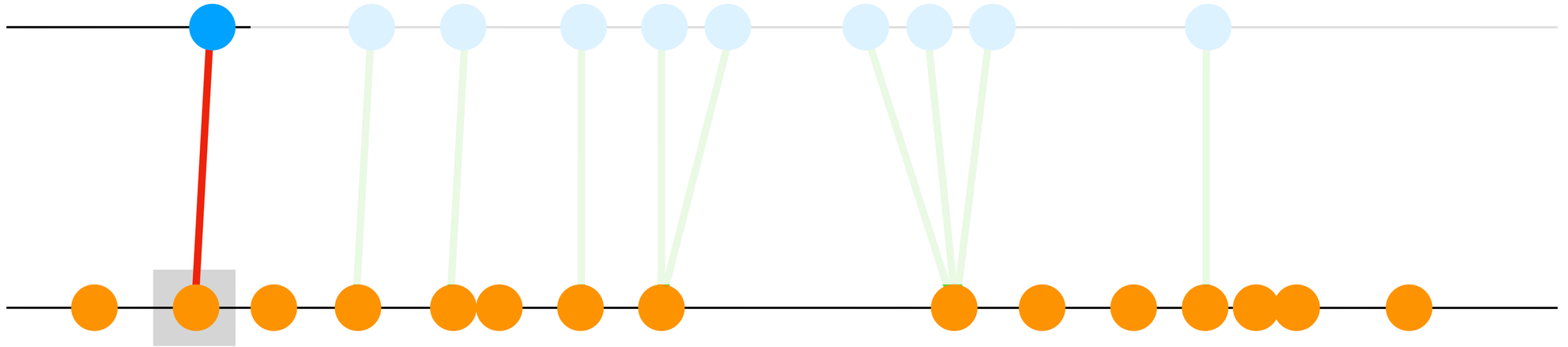
Euclidean Nearest Neighbor assignment

Quadratic time complexity algorithm (linear space)



Euclidean Nearest Neighbor assignment

Quadratic time complexity algorithm (linear space)

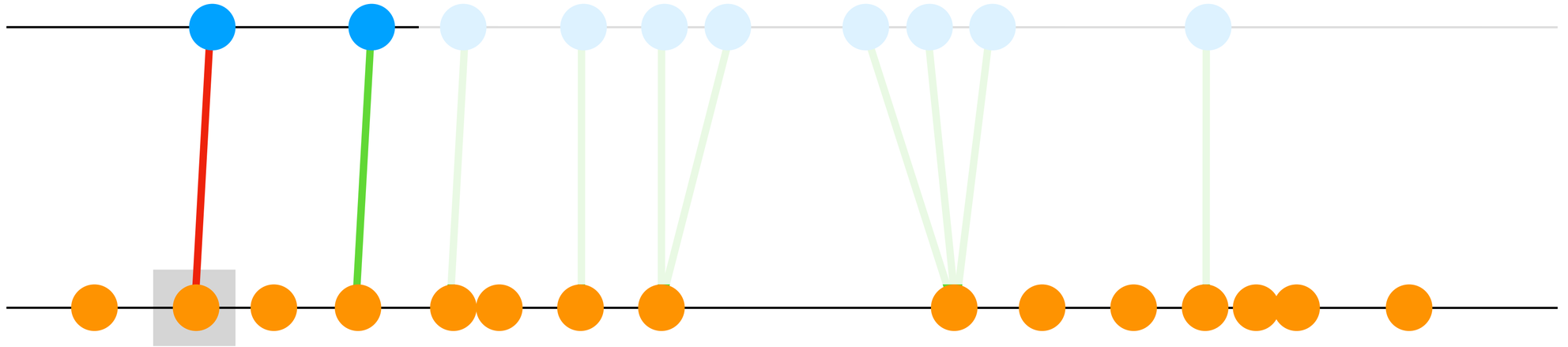


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

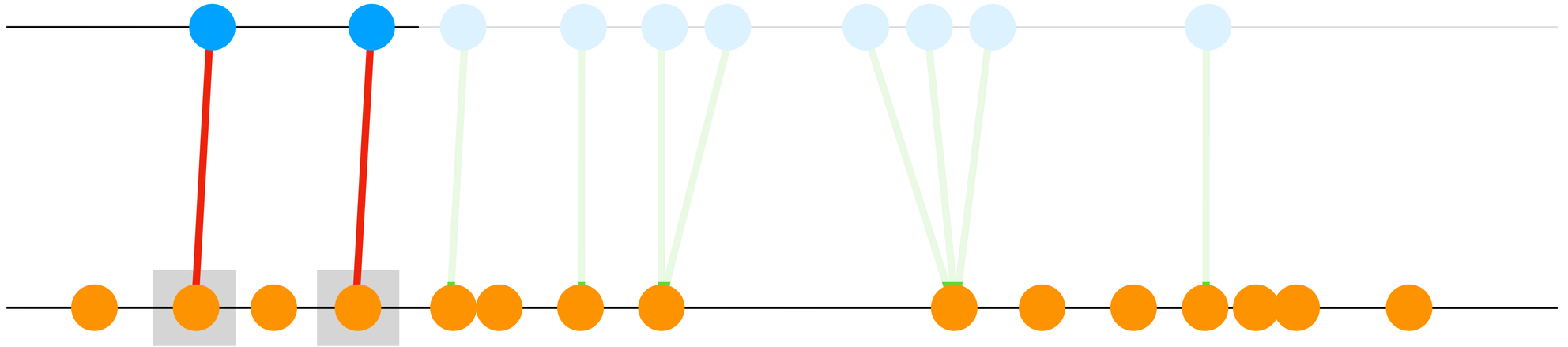


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

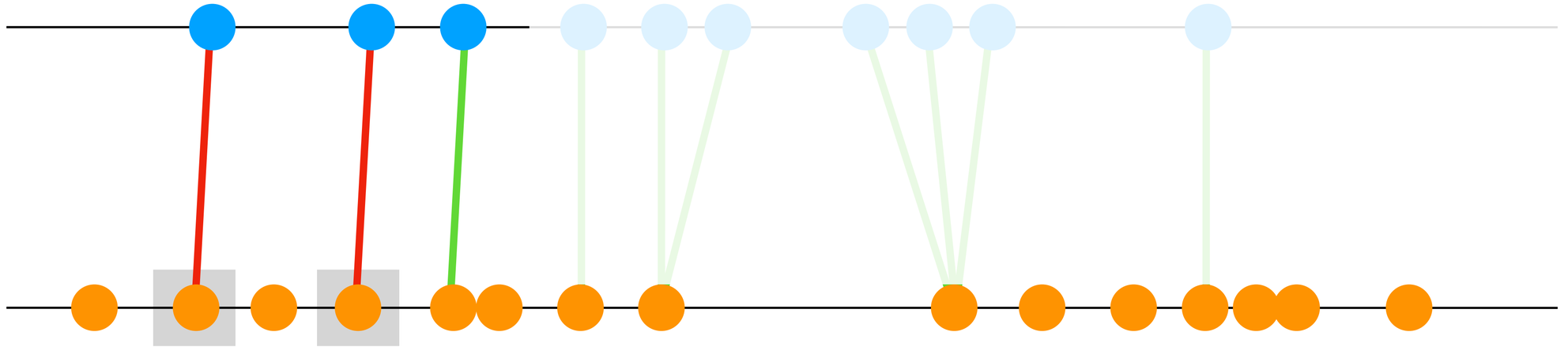


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

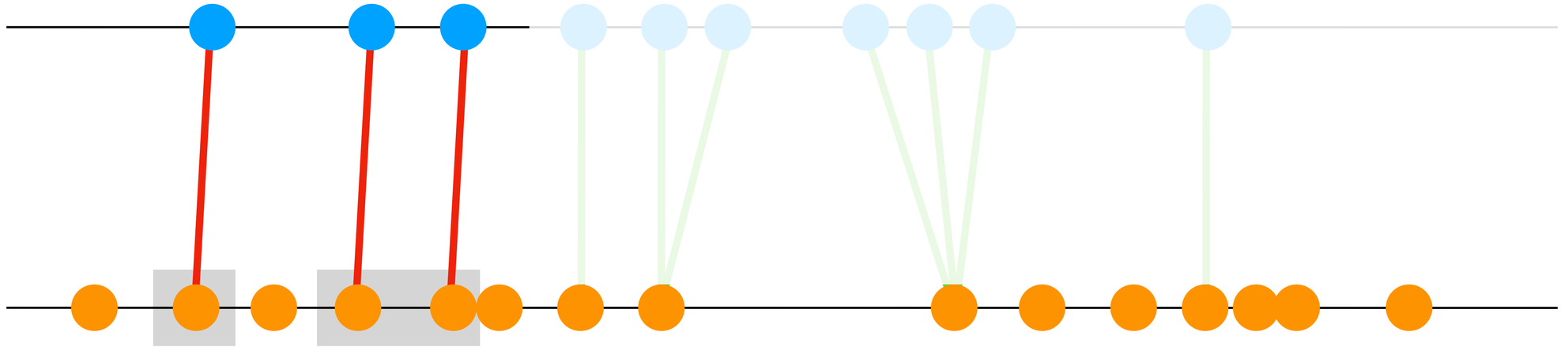


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

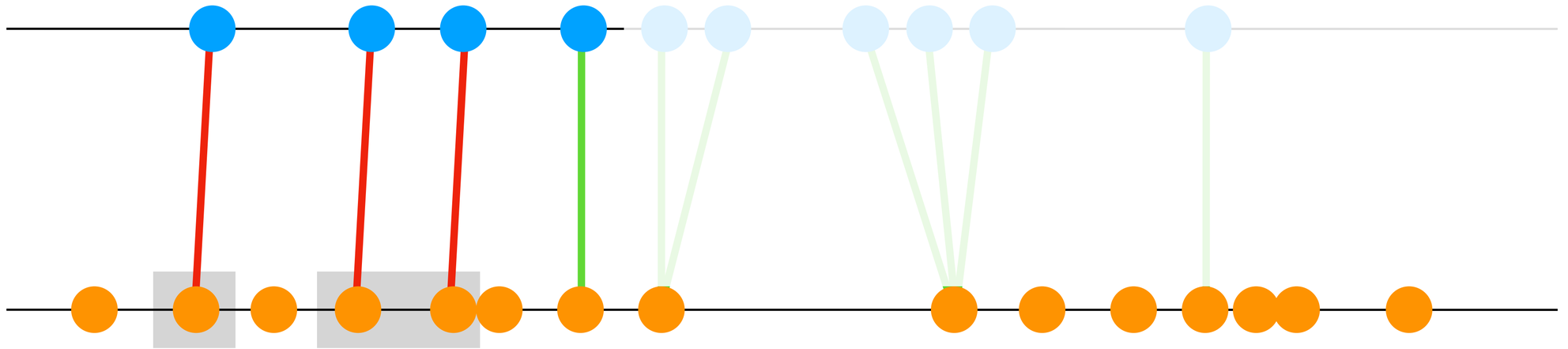


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

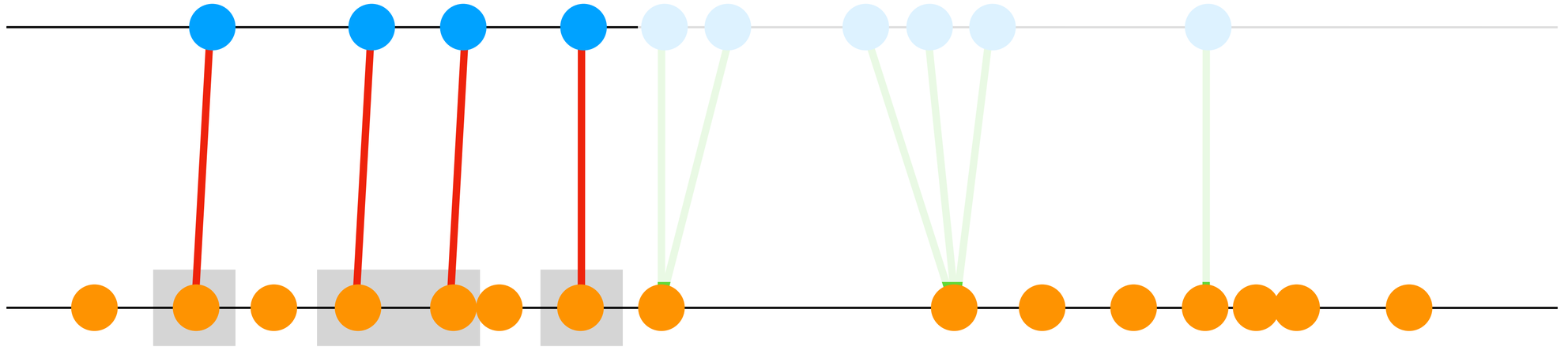


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

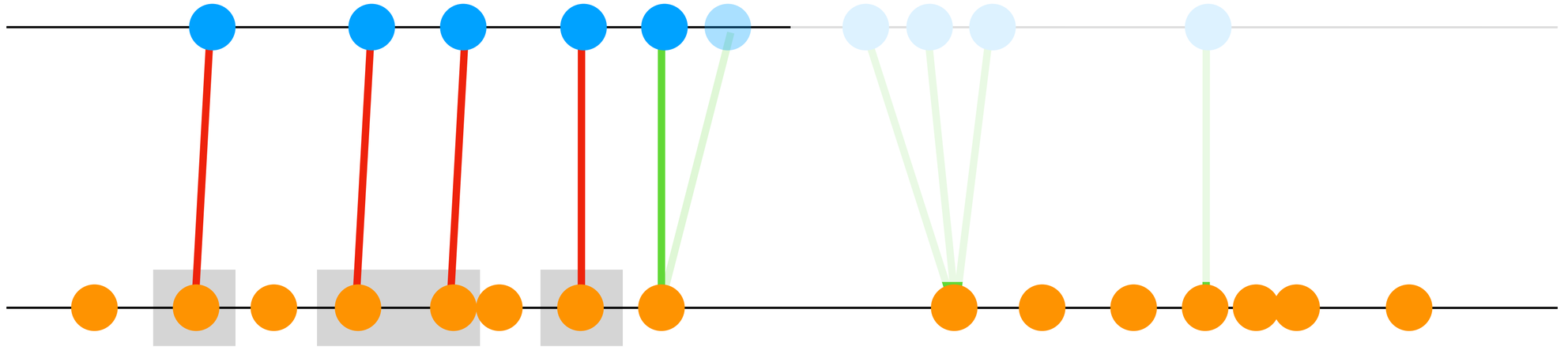


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

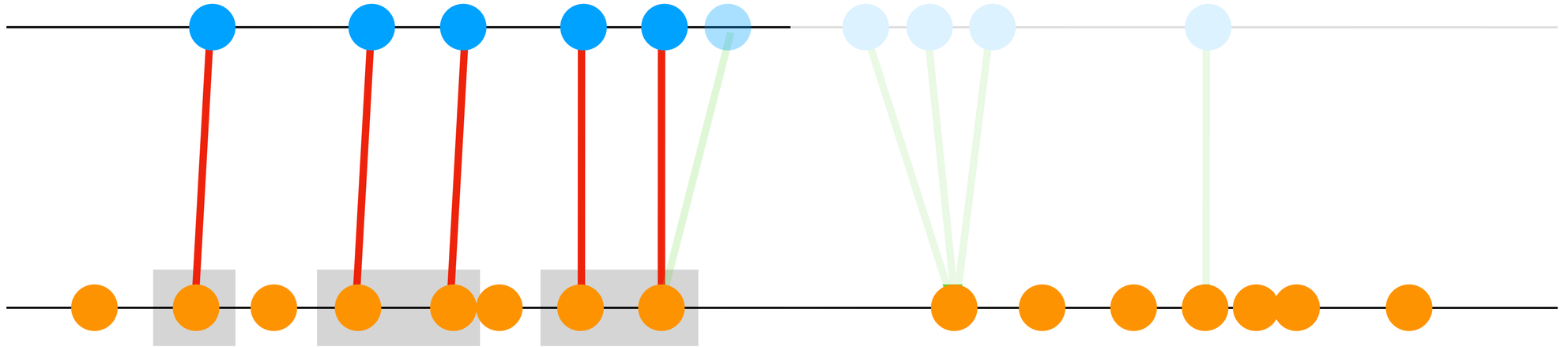


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

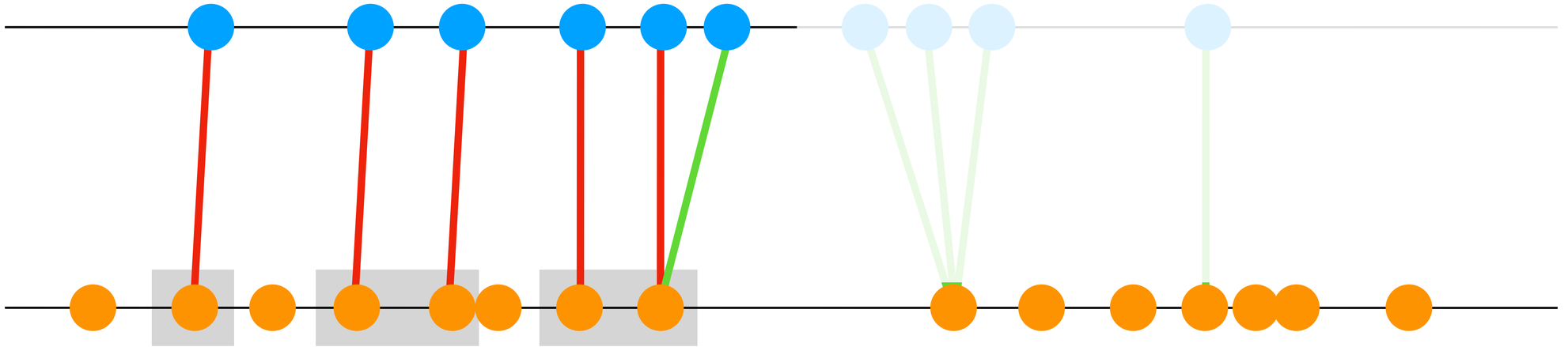


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

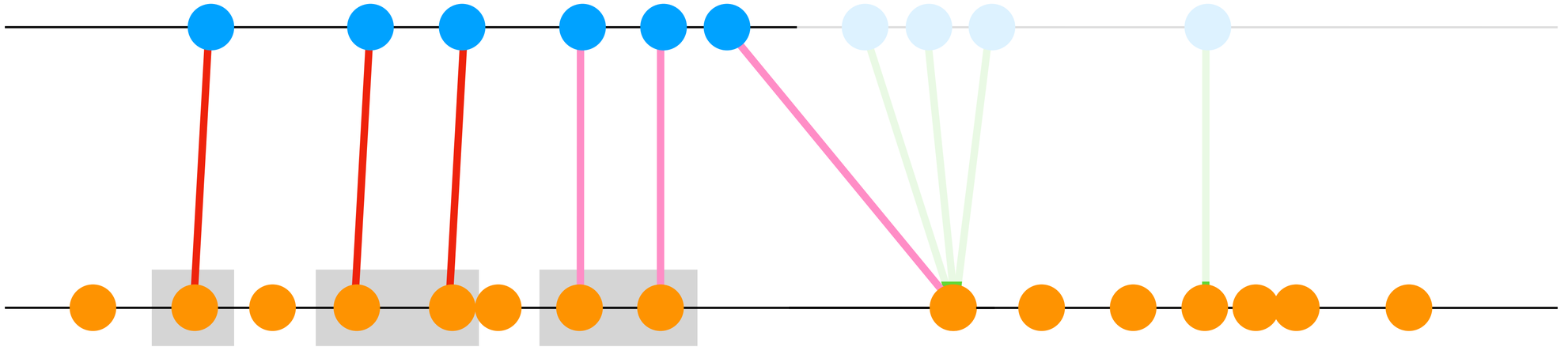


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

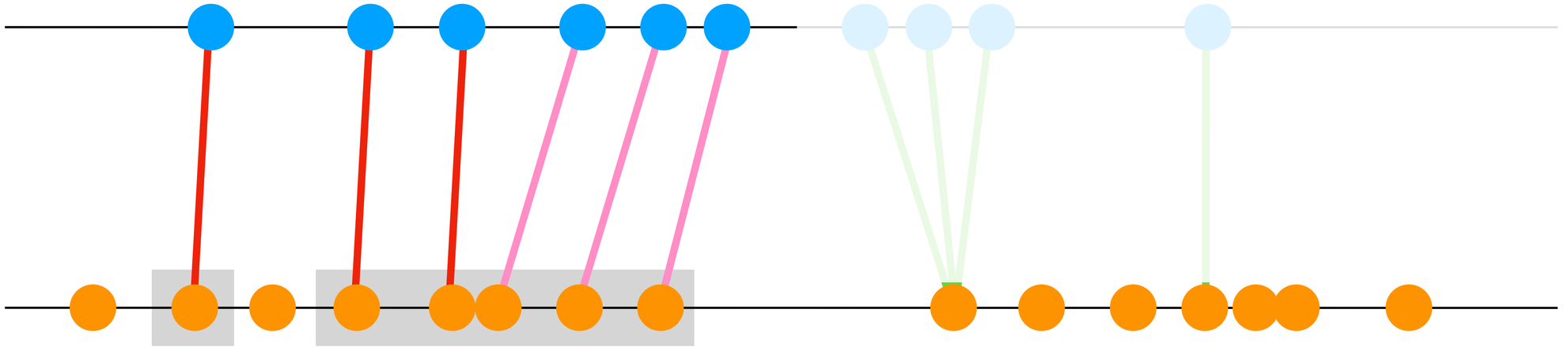


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

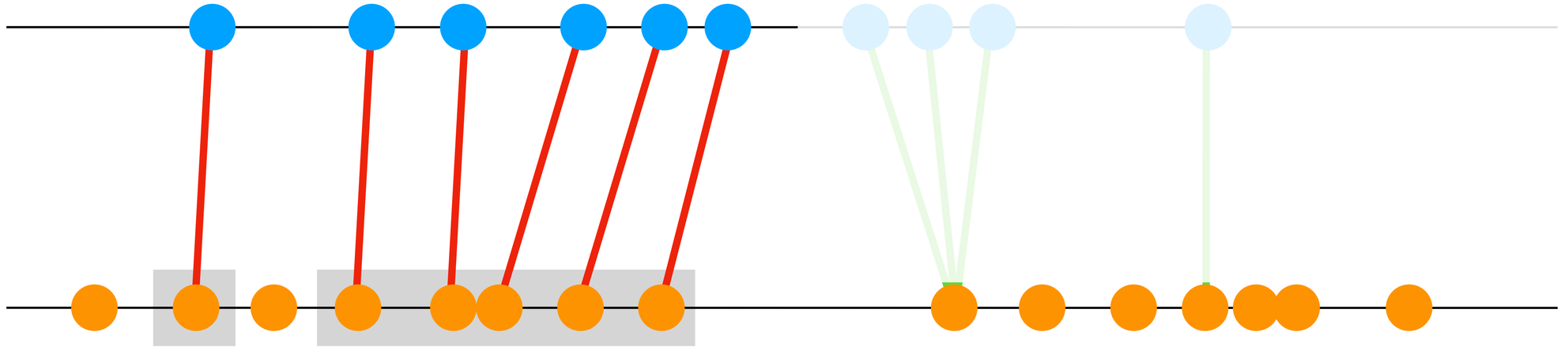


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

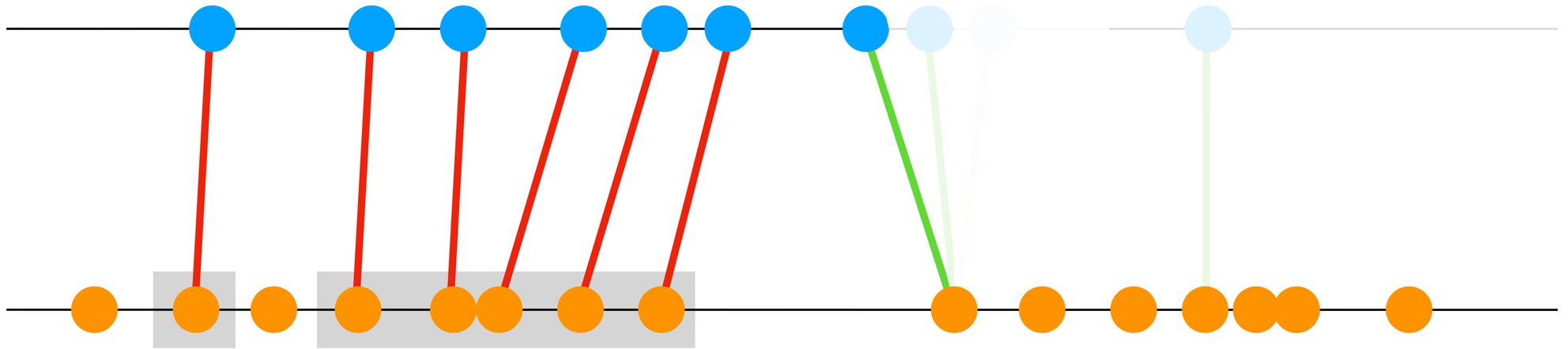


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

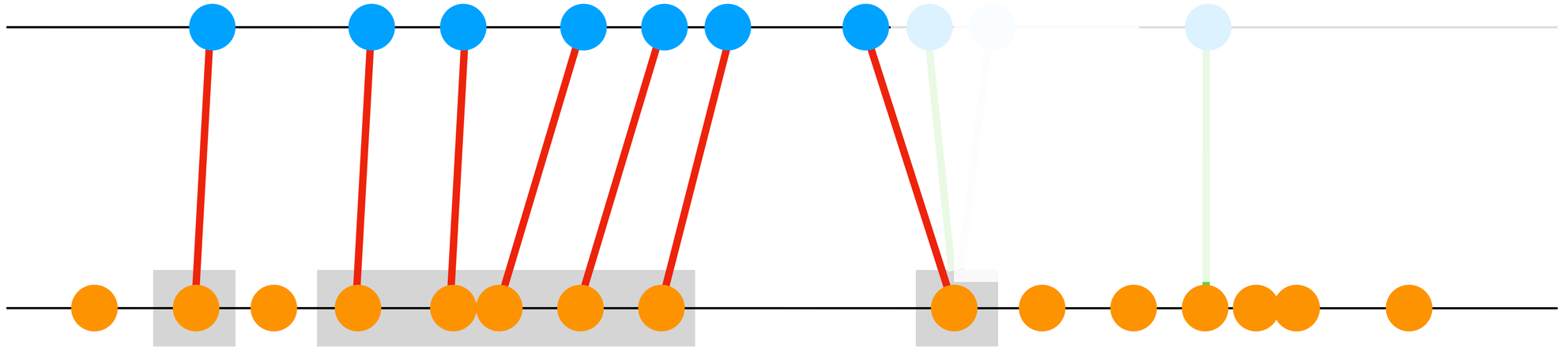


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

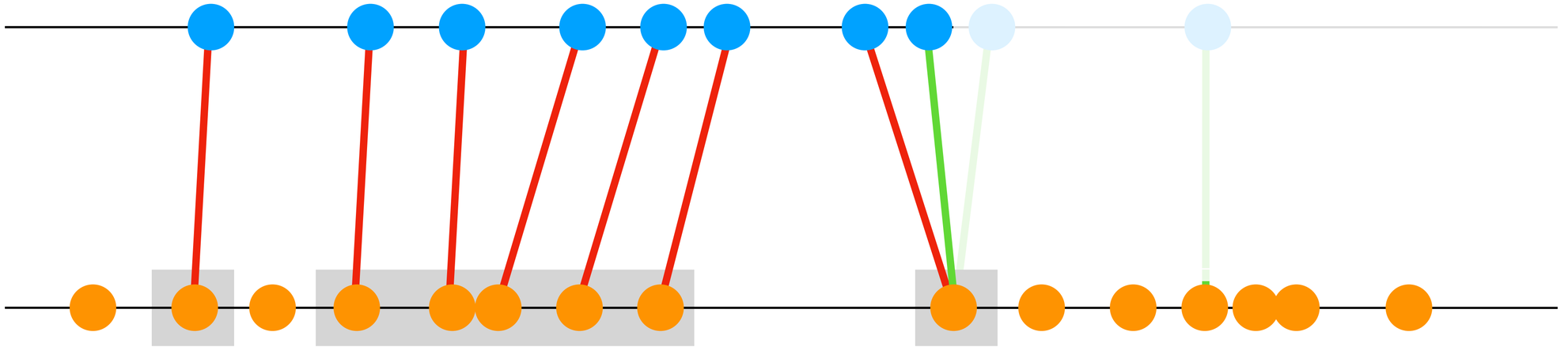


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

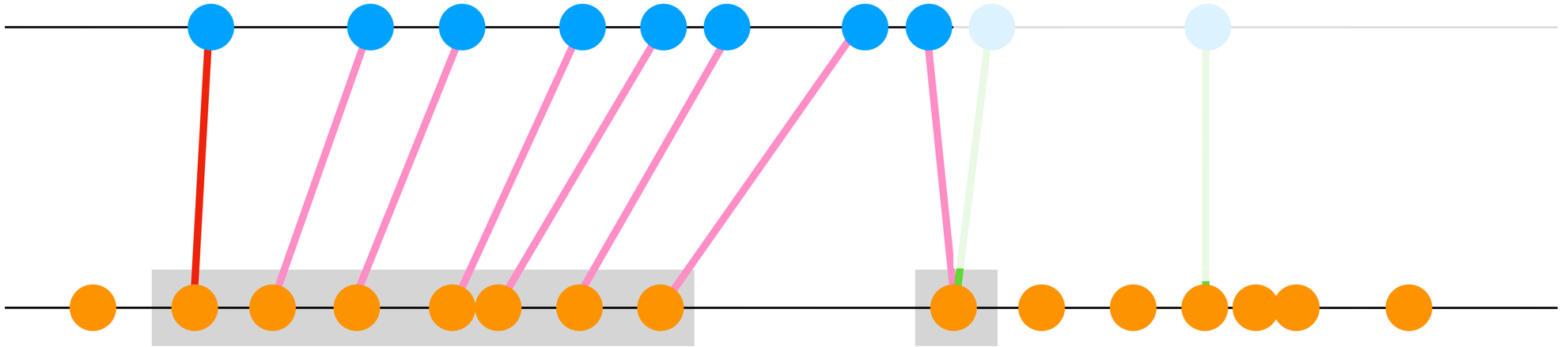


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

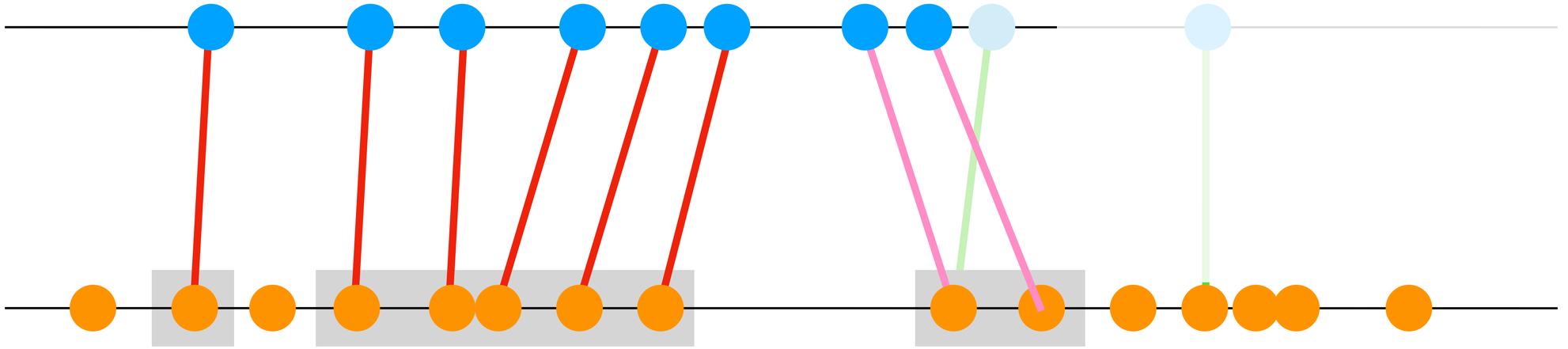


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

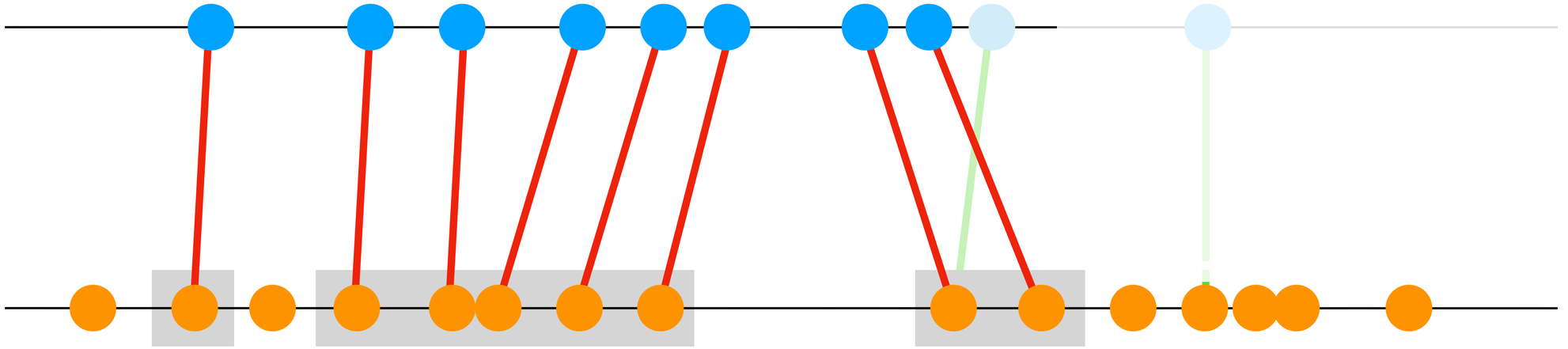


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

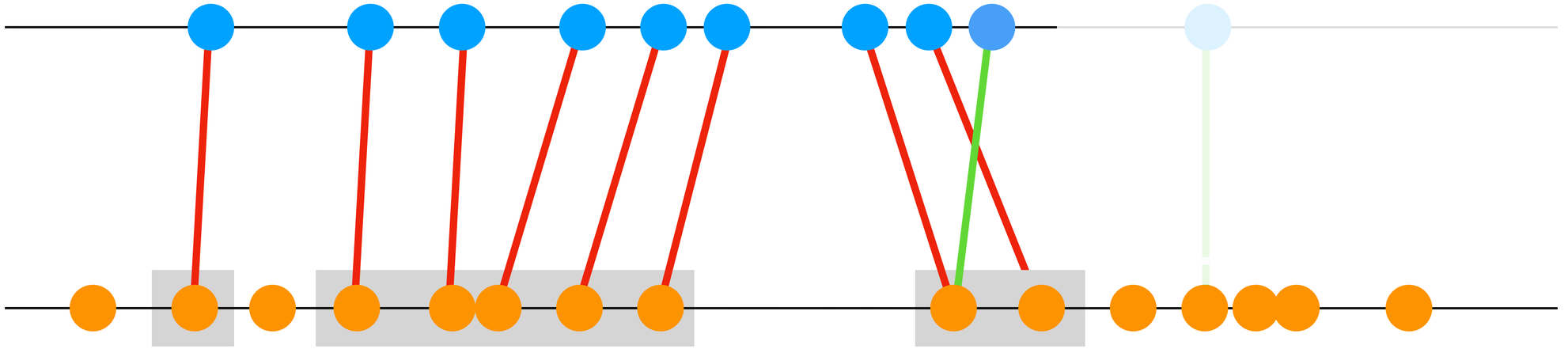


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

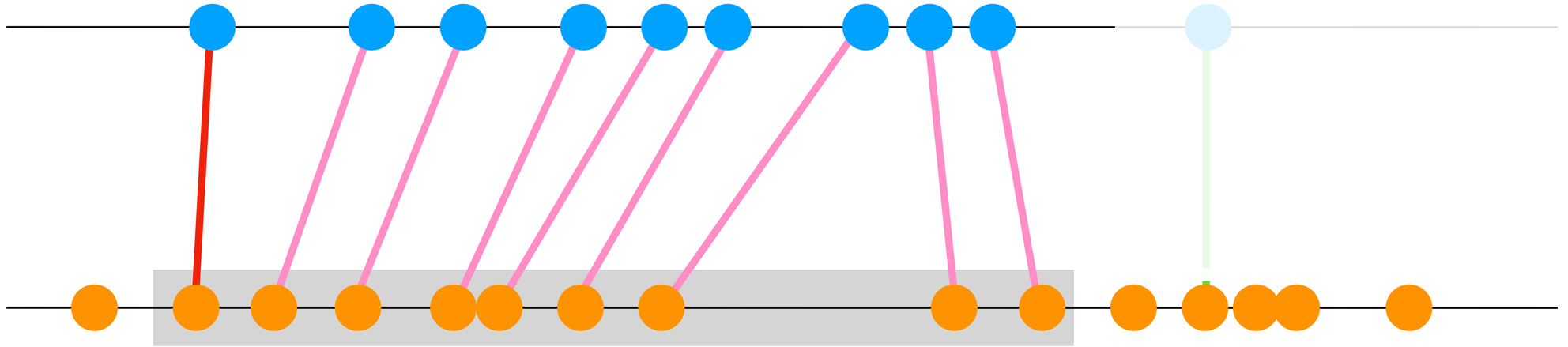


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

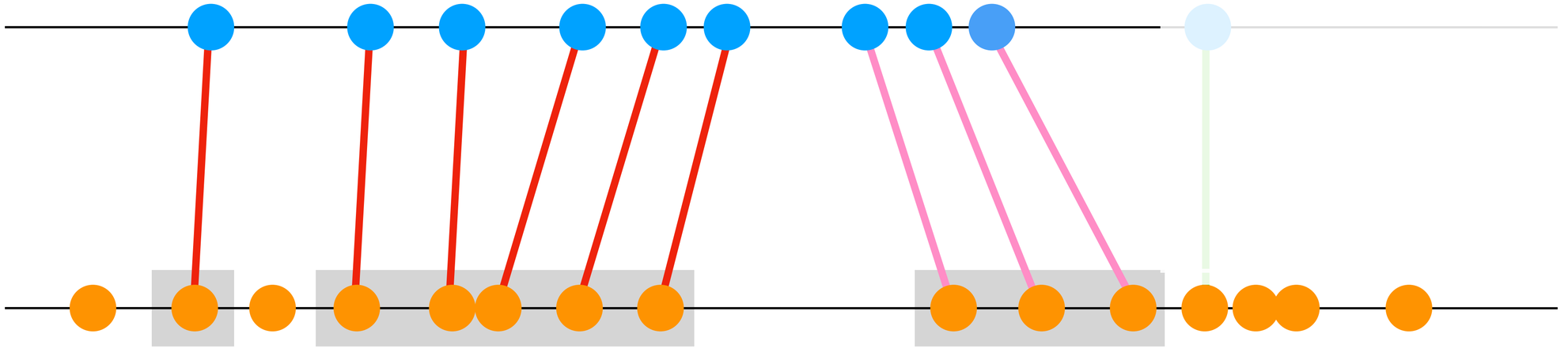


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

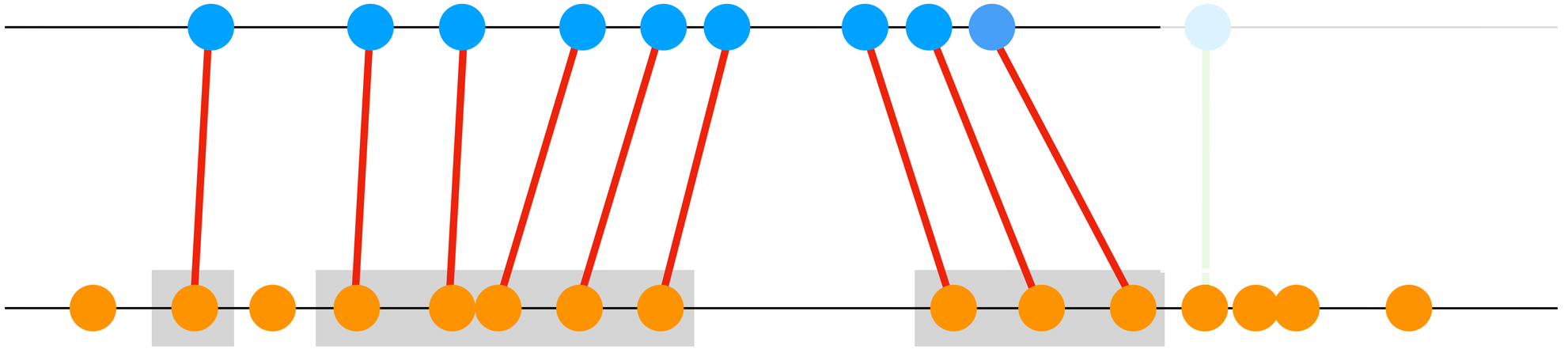


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

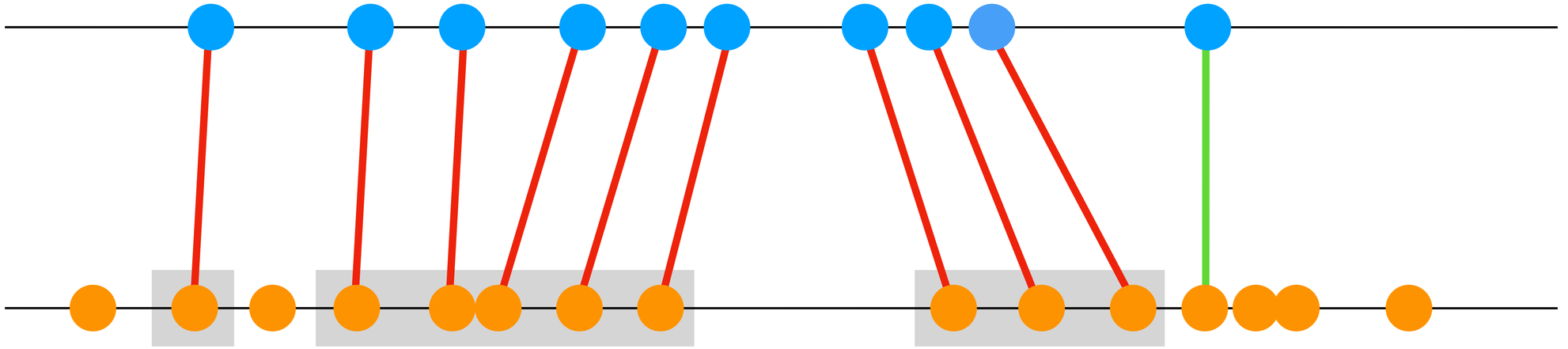


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)

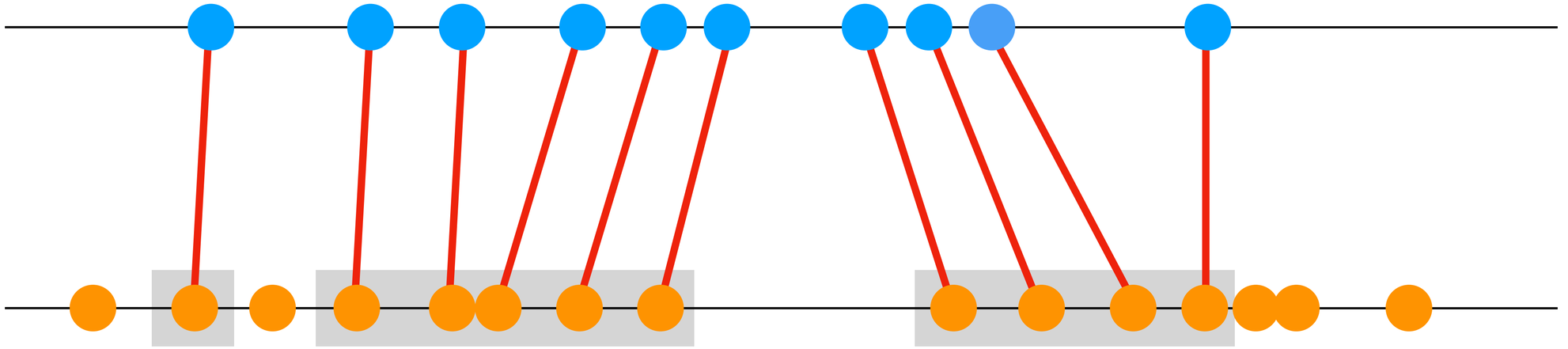


Euclidean Nearest Neighbor assignment

Optimal Transport assignment

Intervals of bijective assignments

Quadratic time complexity algorithm (linear space)



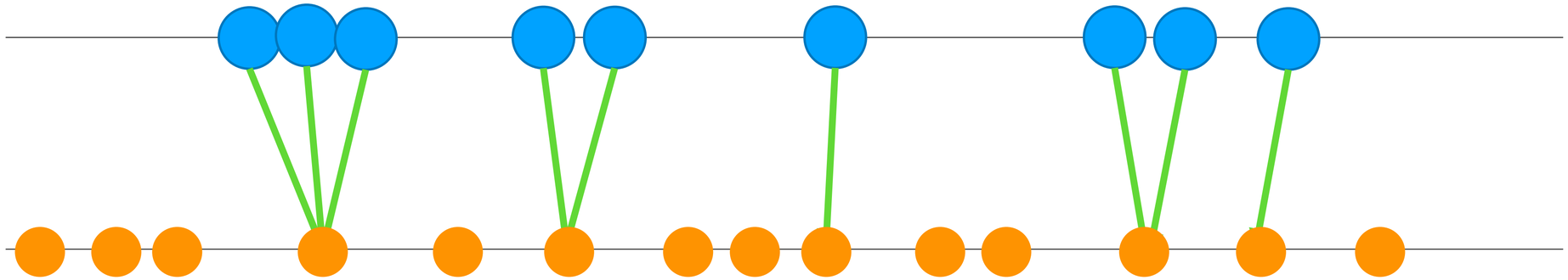
Euclidean Nearest Neighbor assignment

Optimal Transport assignment

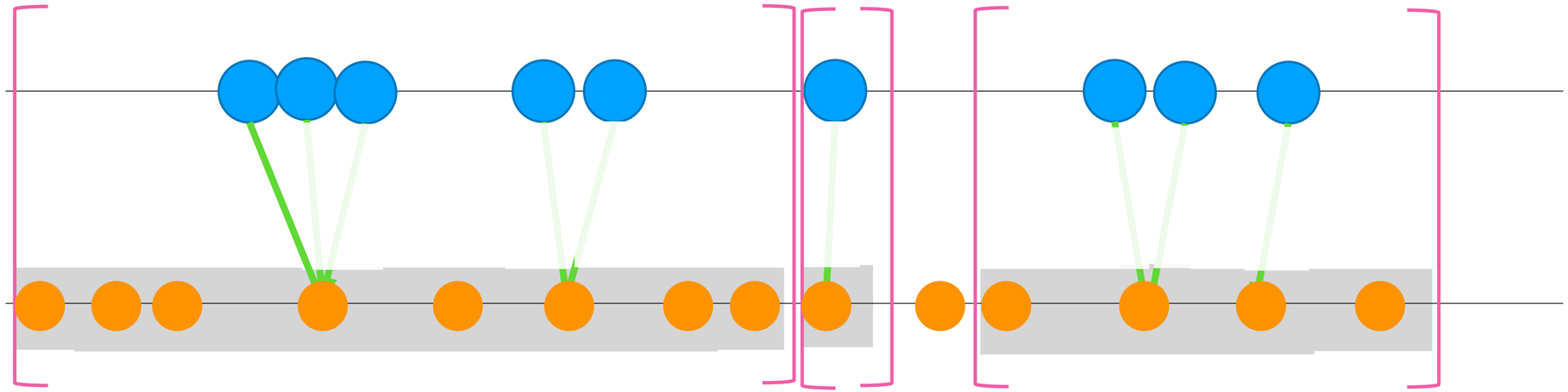
Intervals of bijective assignments

Linear time problem decomposition

Problem decomposition



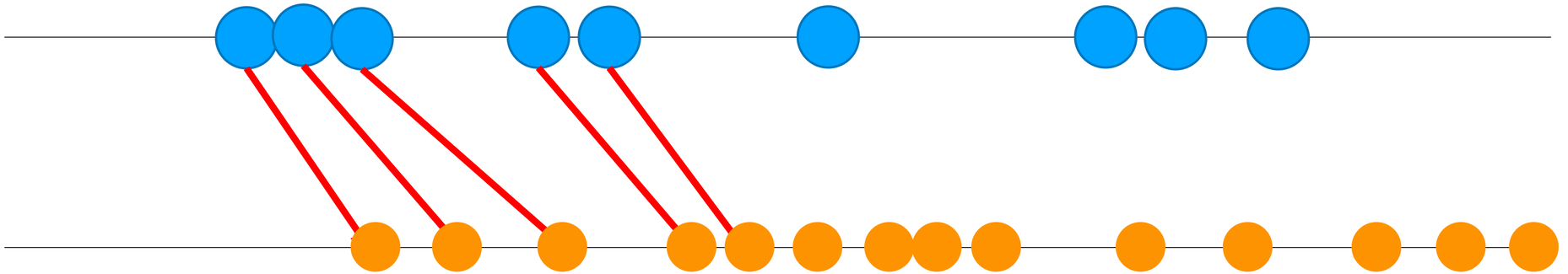
Problem decomposition



Simplifications

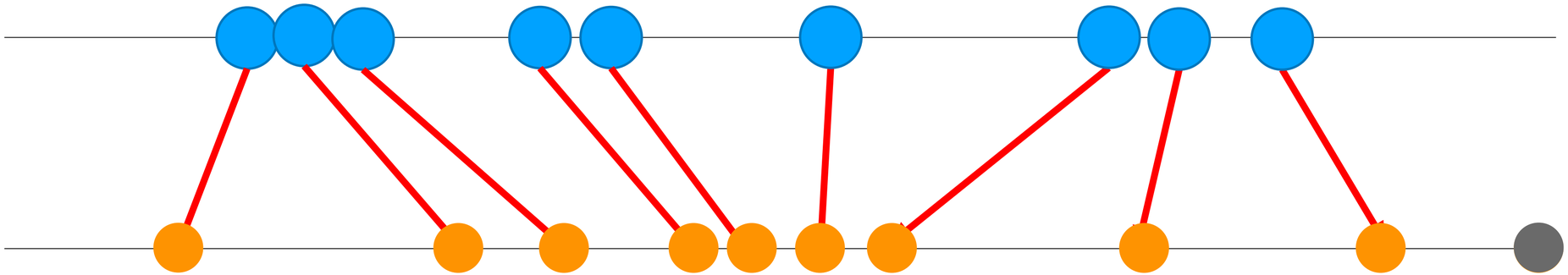
Simplification rules

- Matching N among N, or 1 among N: trivial
- Nearest neighbors injective: trivial
- Snap to the left (similarly to the right)



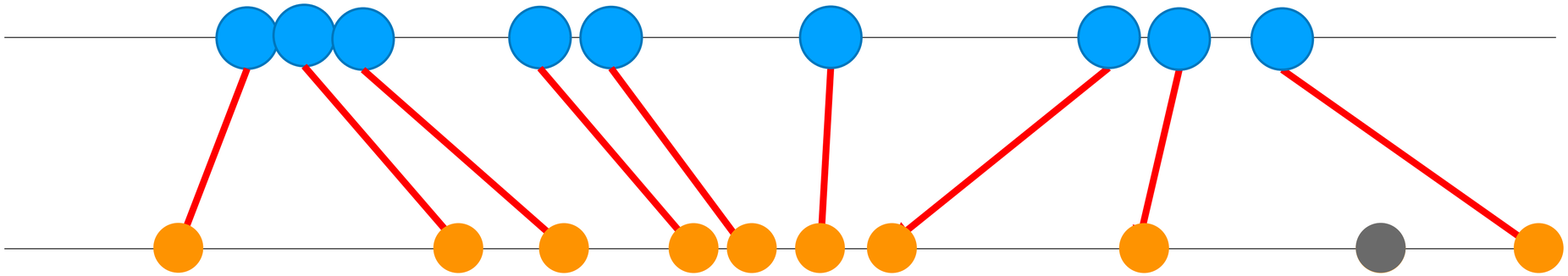
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



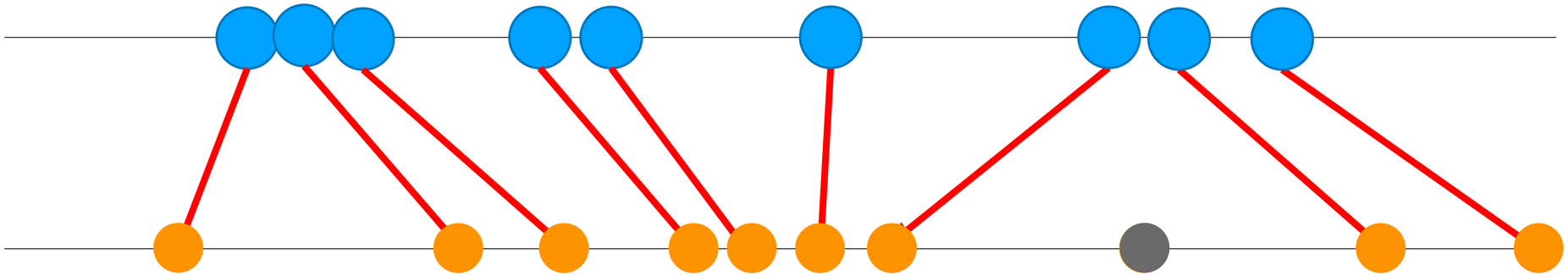
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



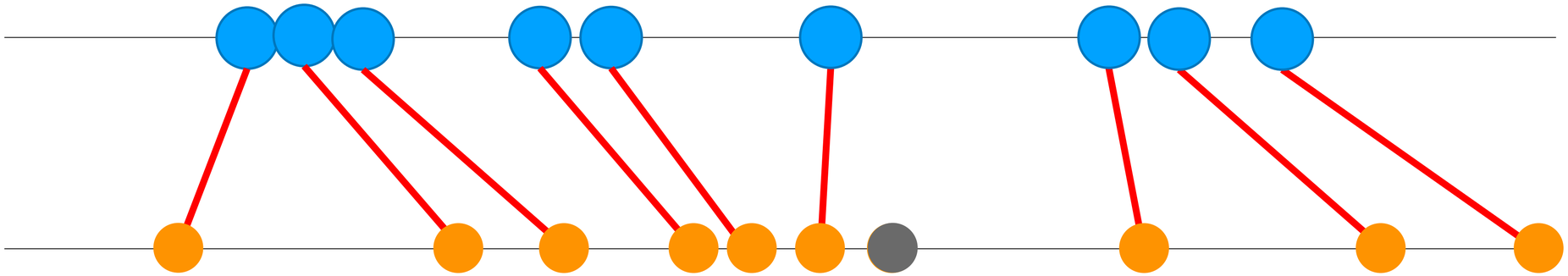
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



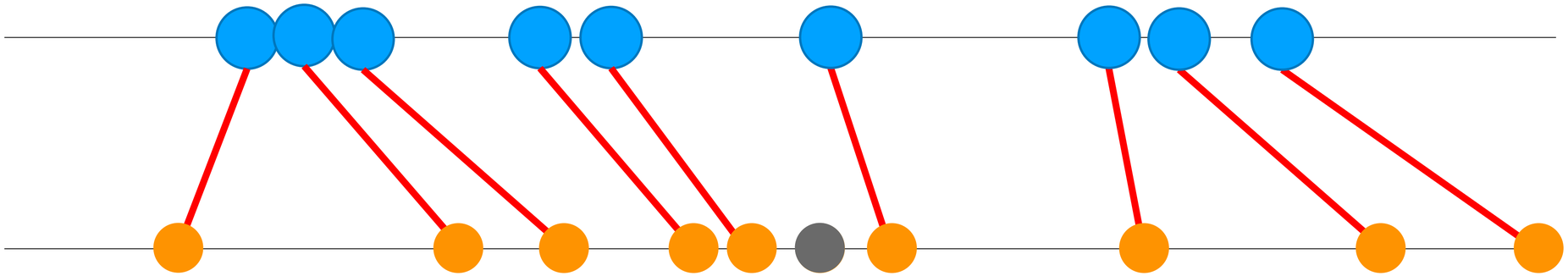
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



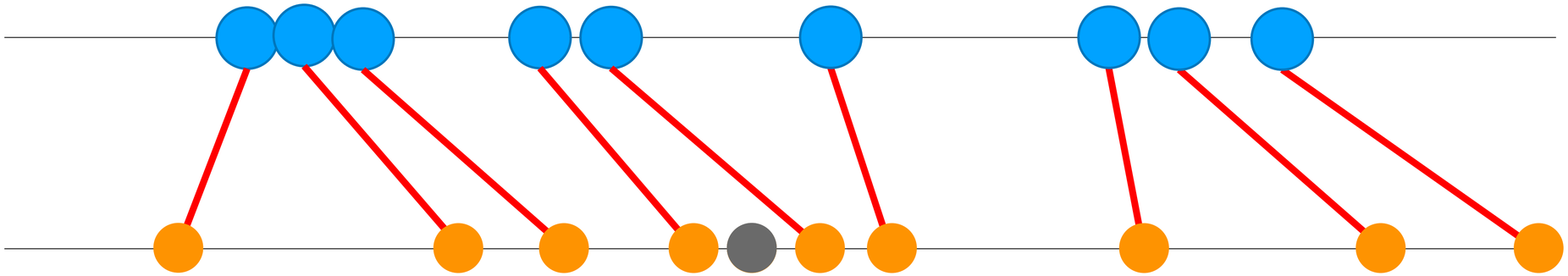
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



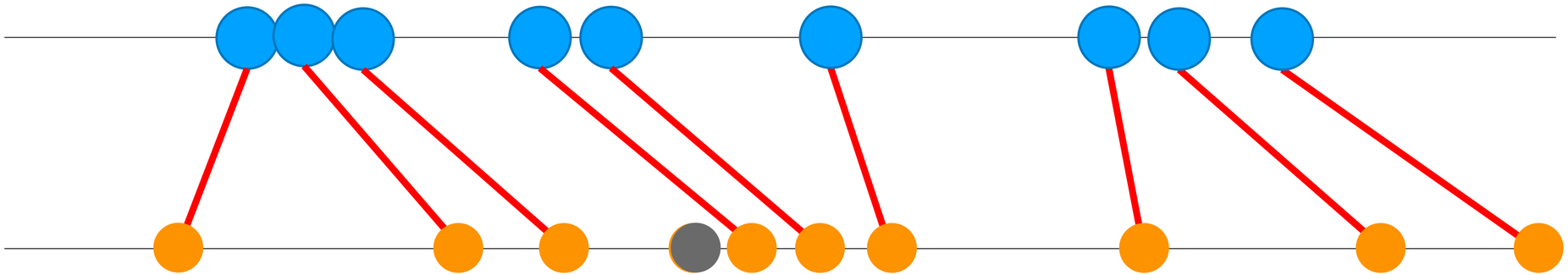
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



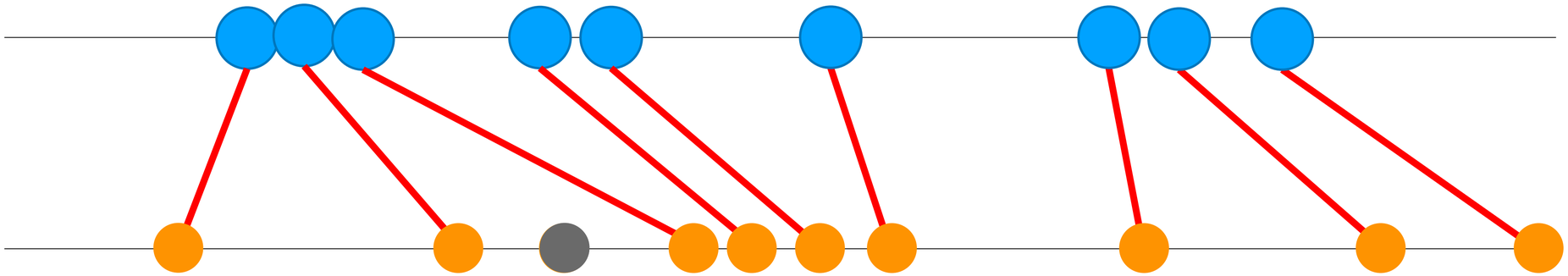
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



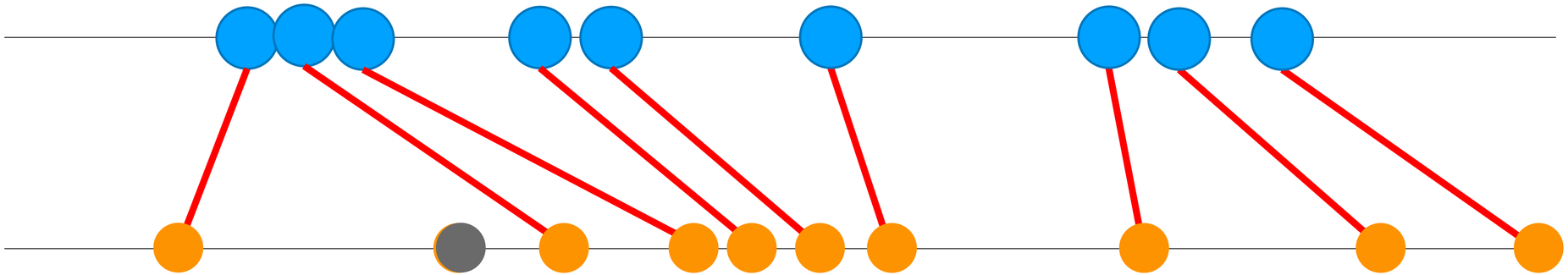
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



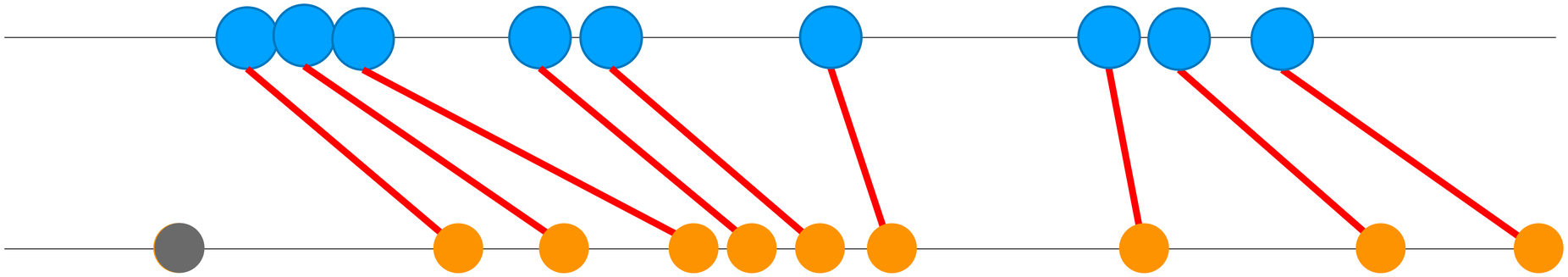
Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts

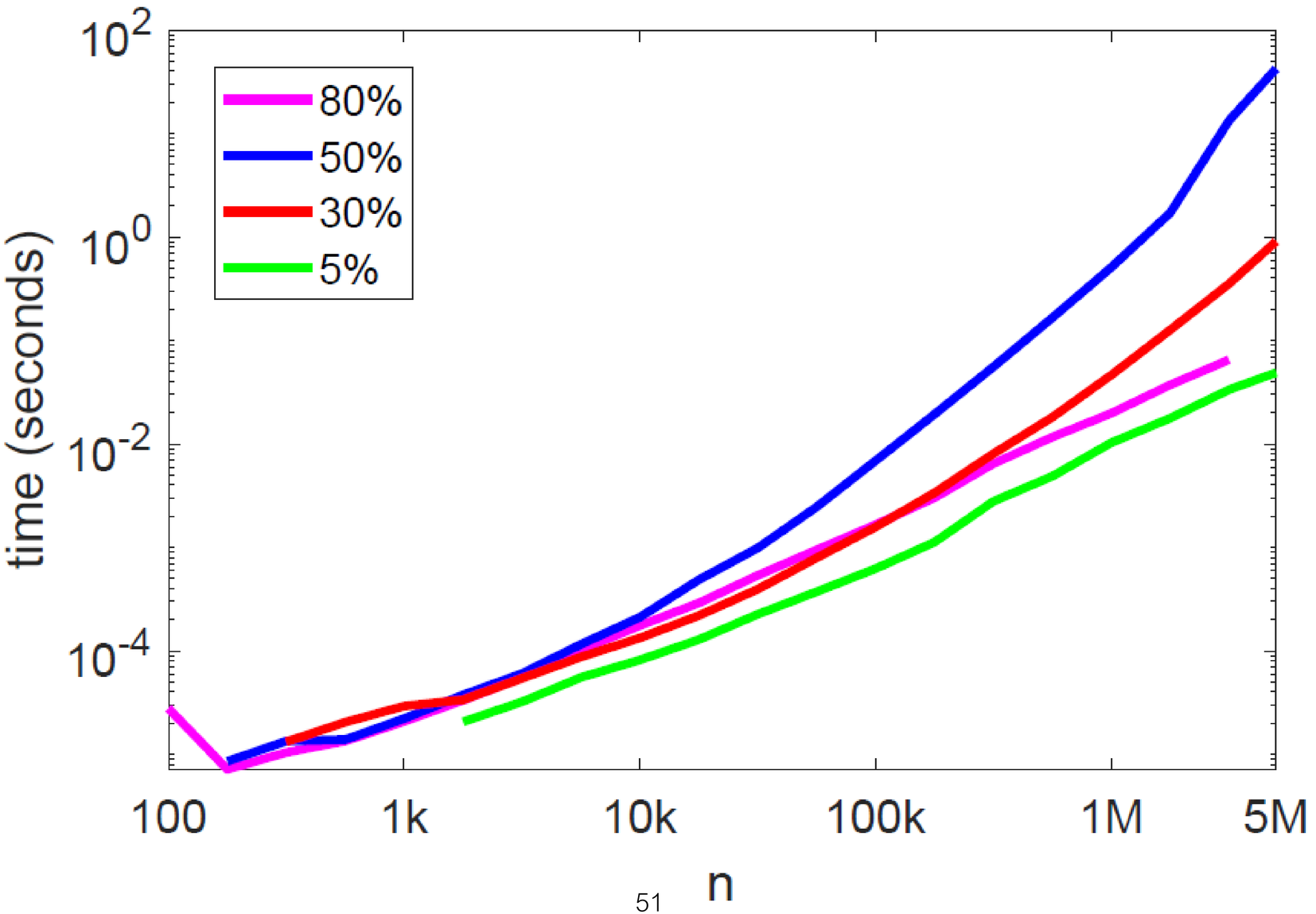


Simplification rules

- Matching N-1 among N
  - Idea from Hirschberg's algorithm
  - Minimum of matching together the K first + matching together N-K-1 lasts



Speed

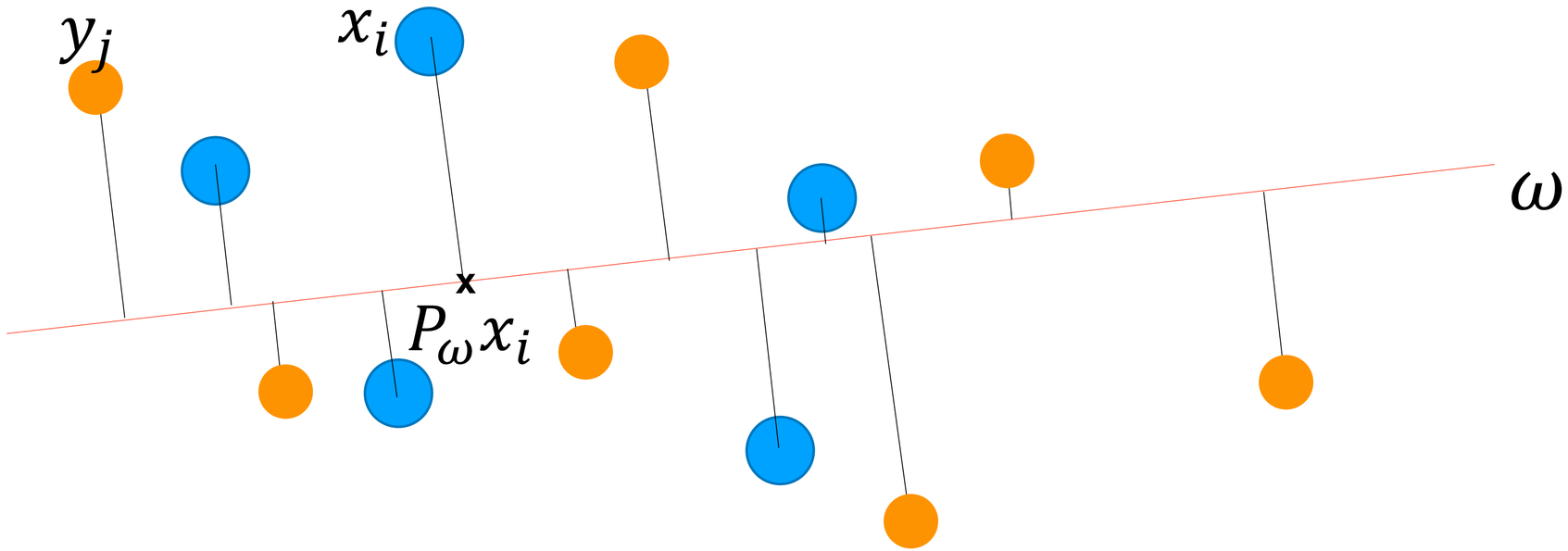


Sliced Partial Optimal Transport (SPOT)

Extension to d dimensions

- Sliced optimal transport cost :

$$E(\{x_i\}_i, \{y_i\}_i) = \int_{\mathbb{S}^{d-1}} \min_{T_\omega} \sum_i \left(P_\omega x_i - P_\omega y_{T_\omega(i)}\right)^2 d\omega$$

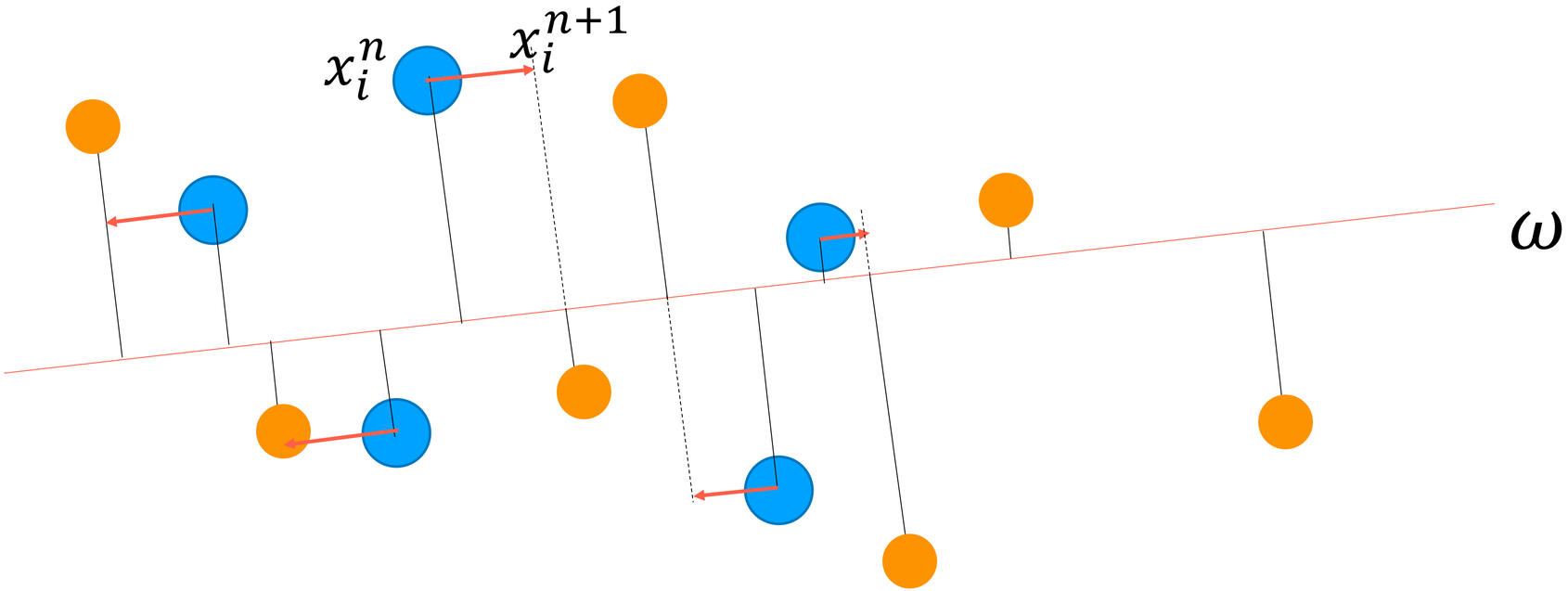


Extension to d dimensions

- Sliced optimal transport gradient flow :

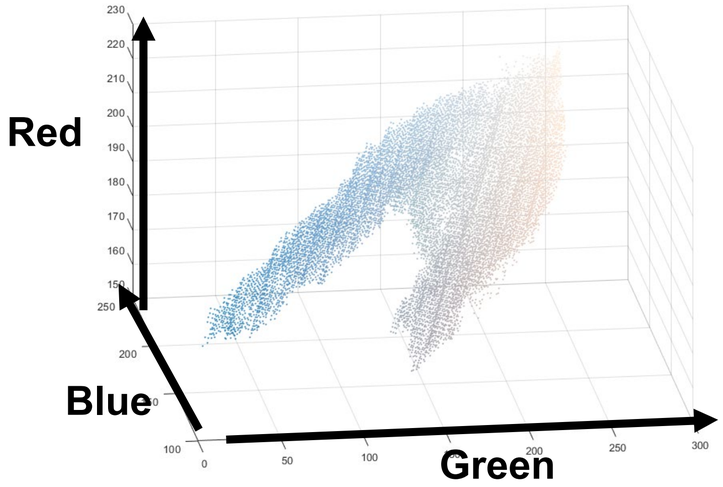
$$x_i^{n+1} = x_i^n - \alpha \nabla_x E$$

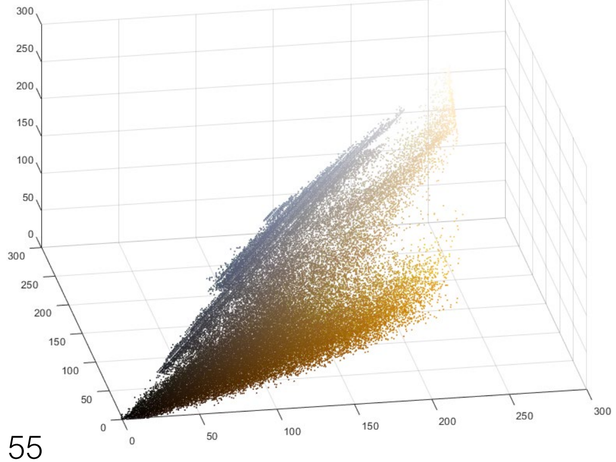
$$\Rightarrow x_i^{n+1} = x_i^n + \alpha \int_{\mathbb{S}^{d-1}} \left(P_\omega y_{T_\omega(i)} - P_\omega x_i \right) \omega \, d\omega$$



Color Transfer Application

- Images seen as point clouds in RGB space





Color Transfer Application

Full Transfer*

Target 20% larger

Target 40% larger

* [Bonneel et al. 2015] Sliced and Radon Wasserstein Barycenters of Measures.

Color Transfer Application

Full Transfer **Target 20% larger** **Target 40% larger**

Color Transfer Application

Full Transfer

Target 20% larger

Target 40% larger

1 superpixel | 2 superpixels | 6 superpixels | 13 superpixels | target

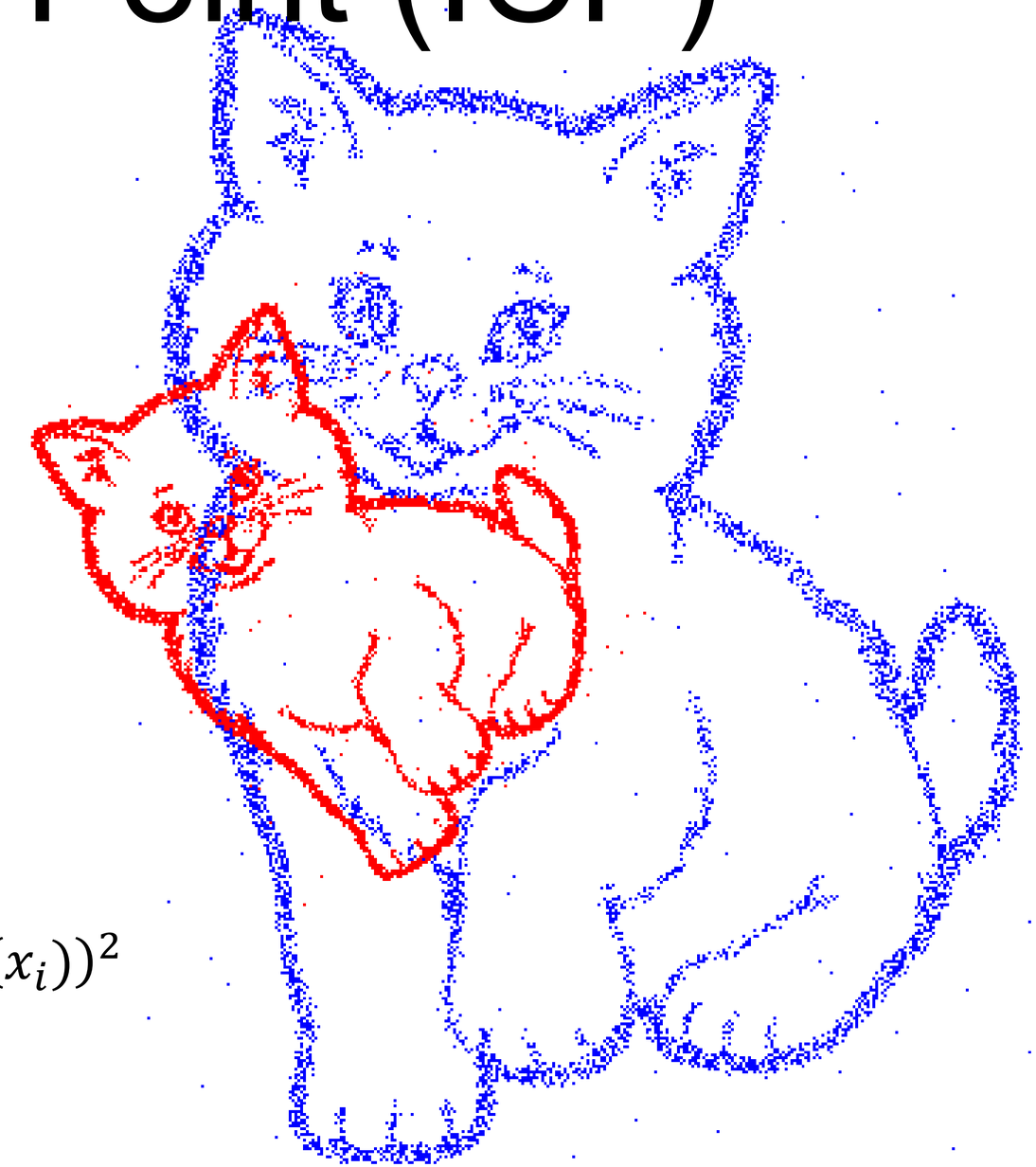
Fast Iterative Sliced Transport

Iterative Closest Point (ICP)

- Registering point sets
  - Class of transformation known
  - Rigid, affine, similarity…
- ICP algorithm
  - Match points to their nearest neighbor
  - Compute transformation
    - Orthogonal Procrustes Problem
- Locally minimizes

$$\min_{\mathrm{M}} \sum_{i} (y_{nn(M(x_i))} - M(x_i))^2$$

$nn(M(x_i))$ Nearest neighbor of $M(x_i)$



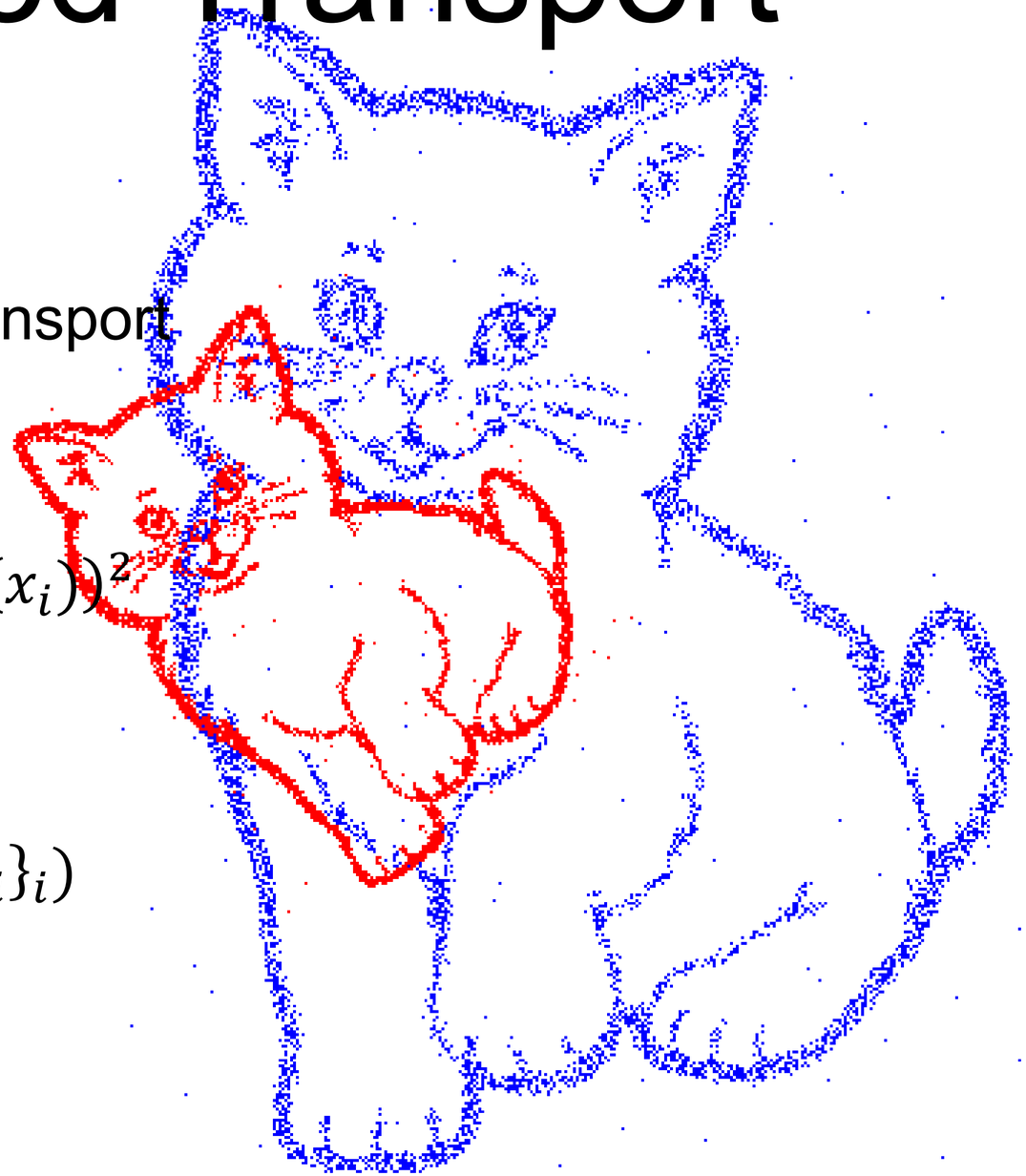
Fast Iterative Sliced Transport

- Replaces nearest neighbor with optimal transport
- Ideally, minimizes

$$\min_{\mathrm{M}} \sum_{i} (y_{OT(M(x_i))} - M(x_i))^2$$

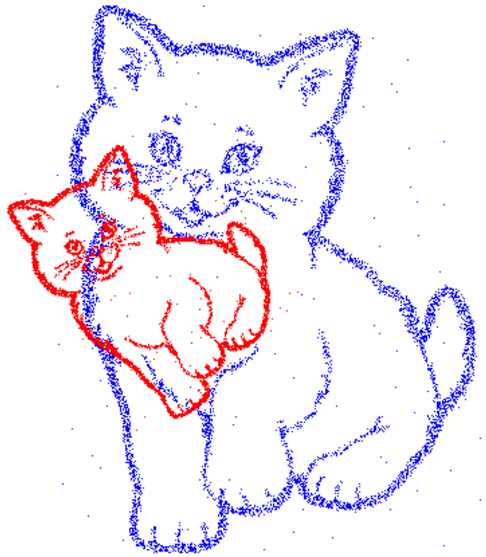
In practice:

$$\min_{\mathrm{M}} E(\{M(x_i)\}_i, \{y_i\}_i)$$



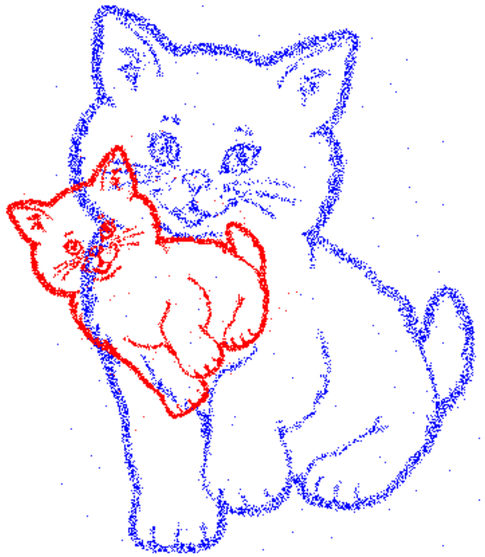
Source: 10k samples
Target: 10k samples

0



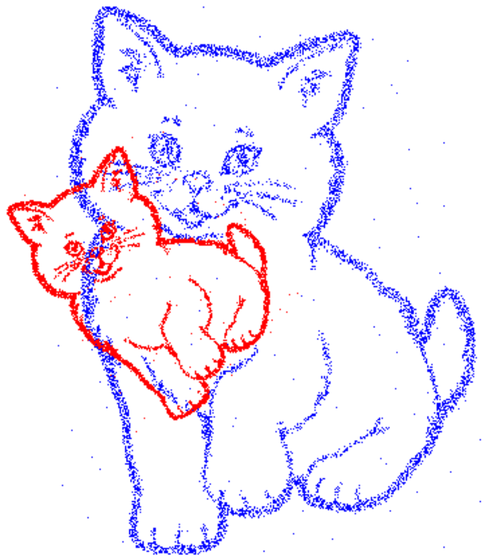
ICP
(0.006 s / iteration)

0



Iterative Transport
with network simplex
(155 s / iteration)

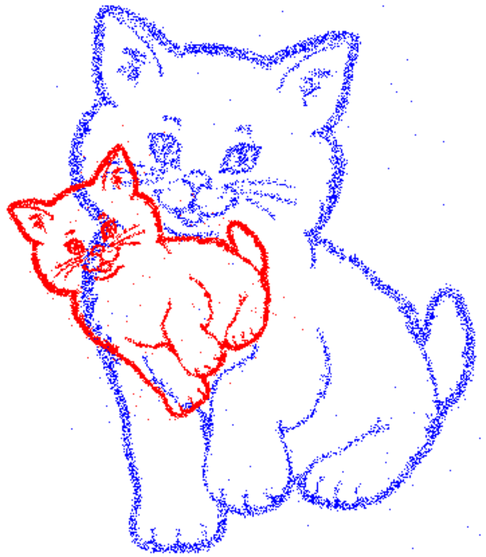
0



Our FIST algorithm
(0.04 s / iteration)

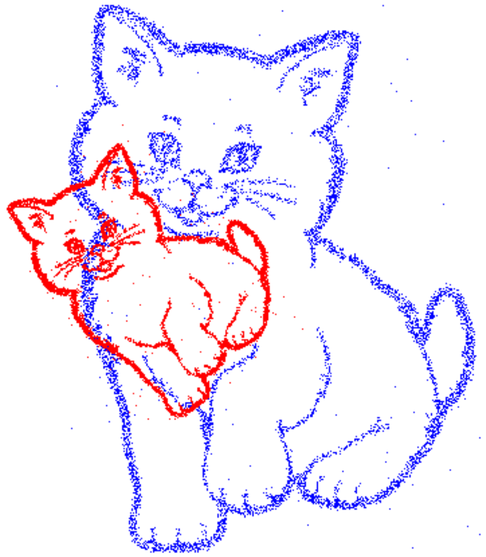
Source: 8k samples
Target: 10k samples

0



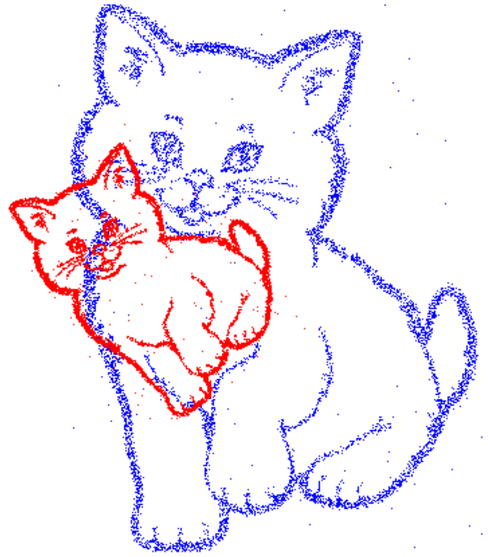
ICP
(0.005 s / iteration)

0



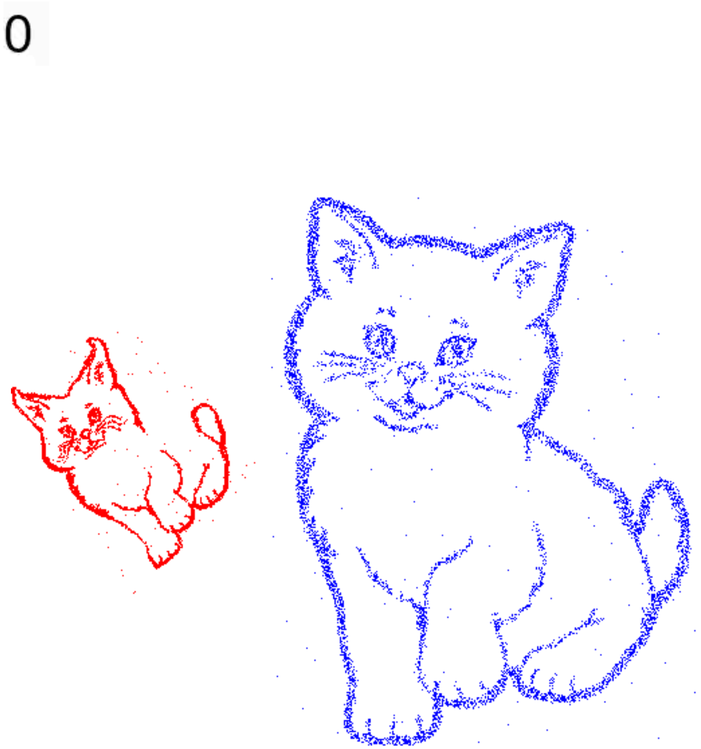
Iterative Transport
with network simplex
(40 s / iteration)

0

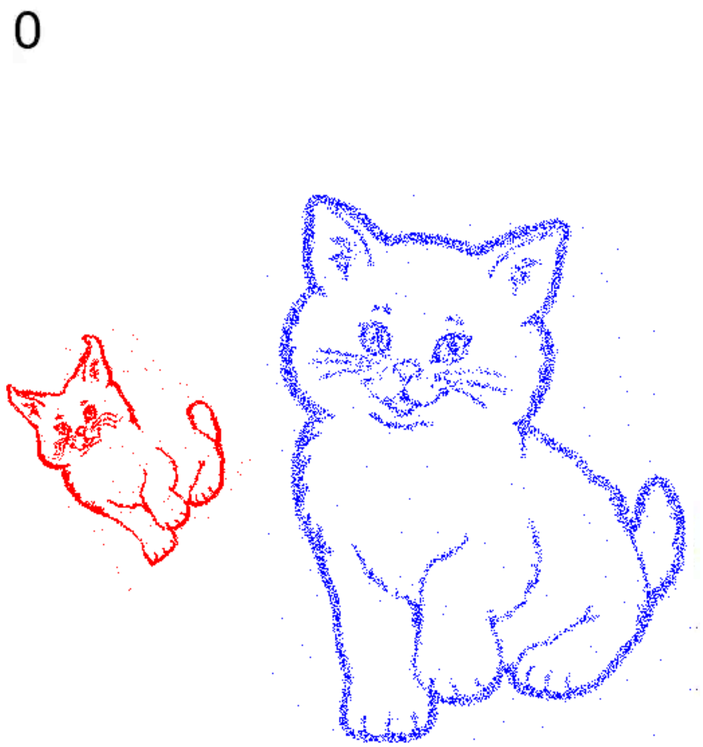


Our FIST algorithm
(0.04 s / iteration)

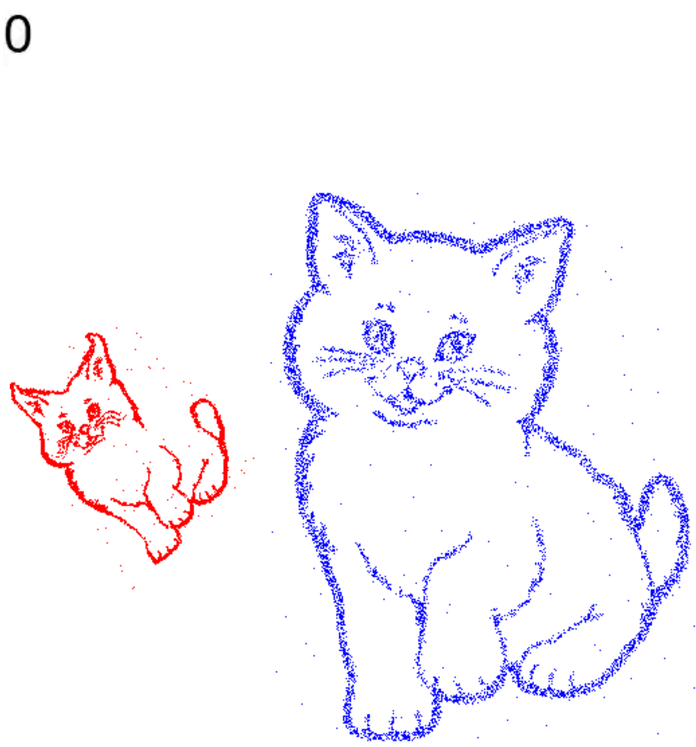
Source: 8k samples
Target: 10k samples



ICP
(0.005 s / iteration)



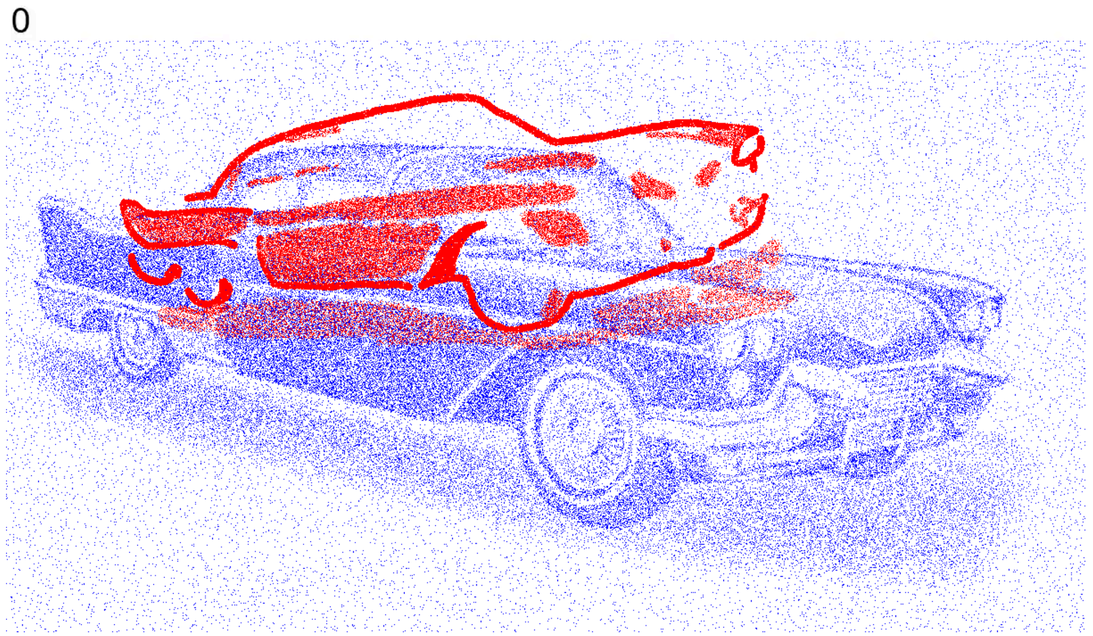
Iterative Transport
with network simplex
(40 s / iteration)



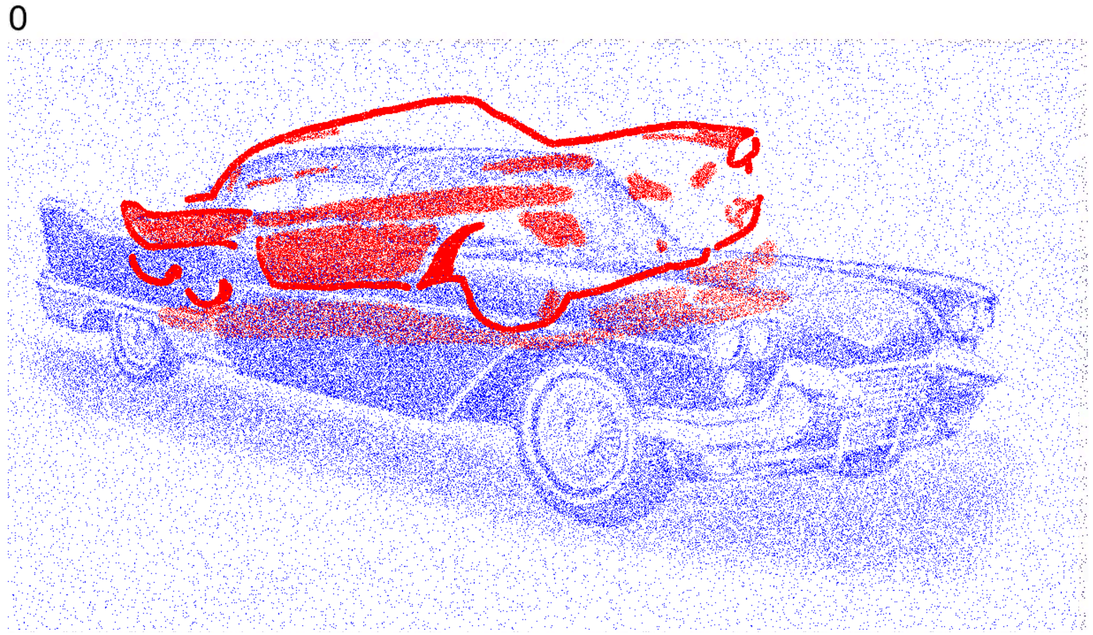
Our FIST algorithm
(0.04 s / iteration)

Source: 90k samples
Target: 100k samples

(input too large for iterative transport with network simplex)



ICP
(0.05 s / iteration)



Our FIST algorithm
(0.66 s / iteration)

Source: 120k samples
Target: 200k samples

(input too large for iterative transport with network simplex)

ICP
(0.07 s / iteration)

Our FIST algorithm
(1.72 s / iteration)

Source: 90k samples
Target: 100k samples

(input too large for iterative transport with network simplex)

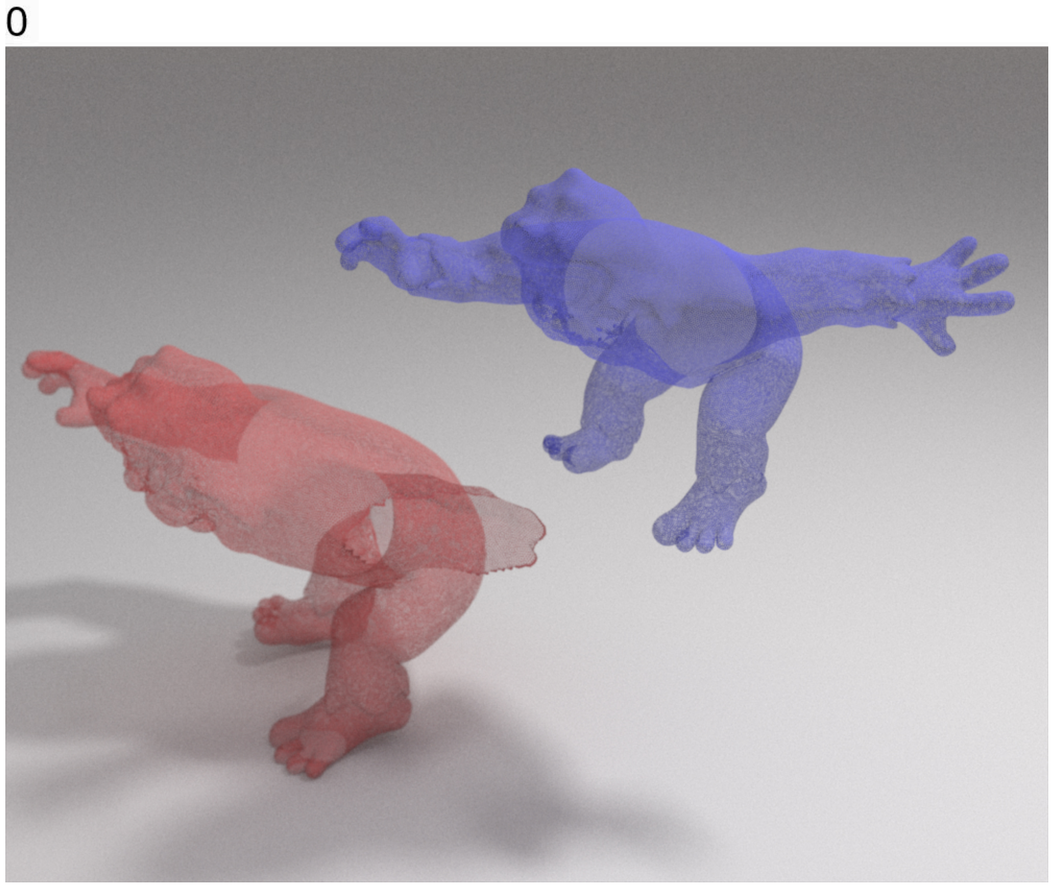
ICP
(0.05 s / iteration)

Our FIST algorithm
(0.69 s / iteration)

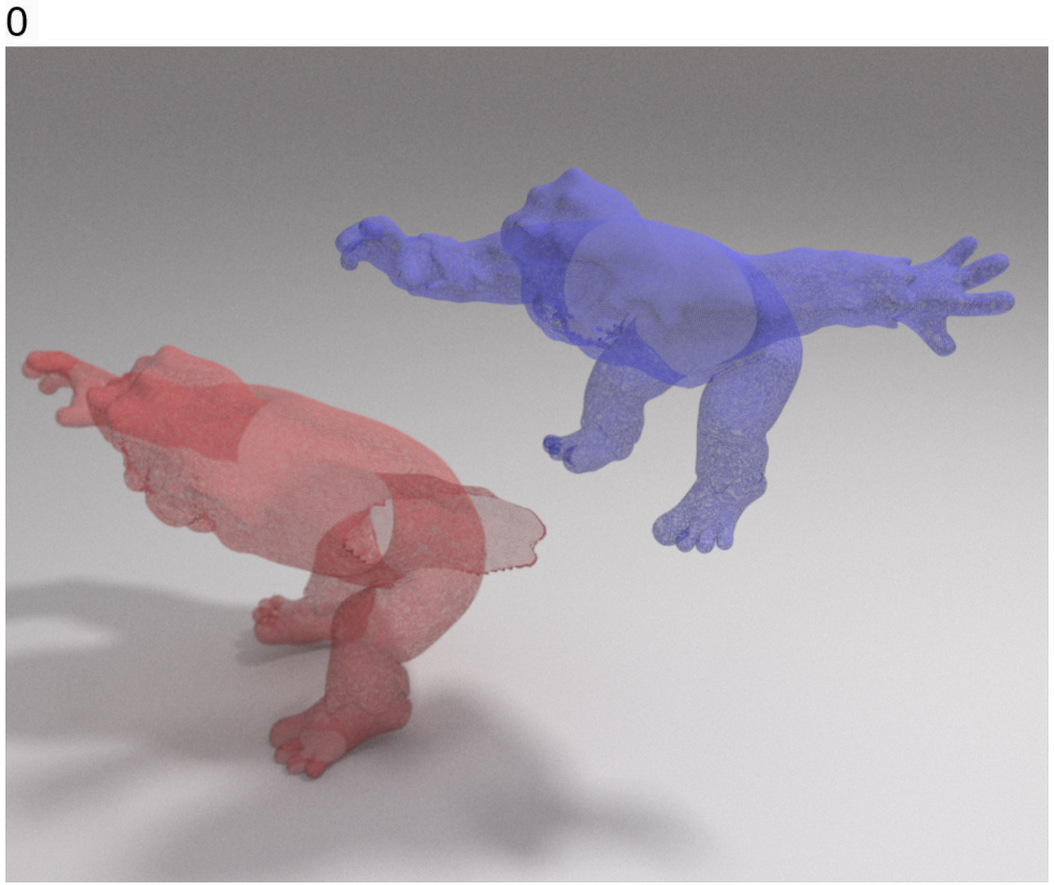
$$\min_{\mathrm{M}} \sum_{i} (y_{nn(\boldsymbol{M}(\boldsymbol{x_i}))} - M(x_i))^2 \;=\; 0\,!$$

Source: 80k samples
Target: 100k samples

(input too large for iterative t transport with network simplex)



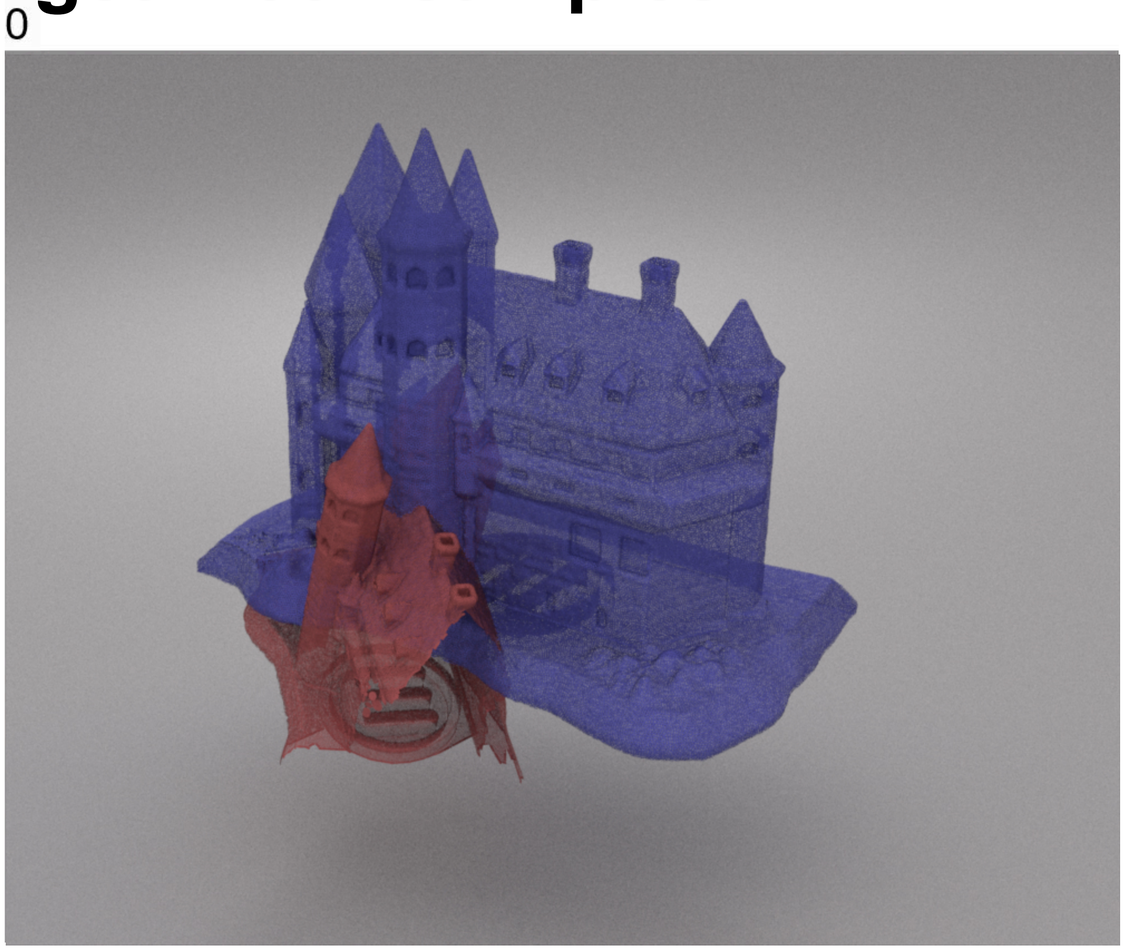
ICP
(0.05 s / iteration)



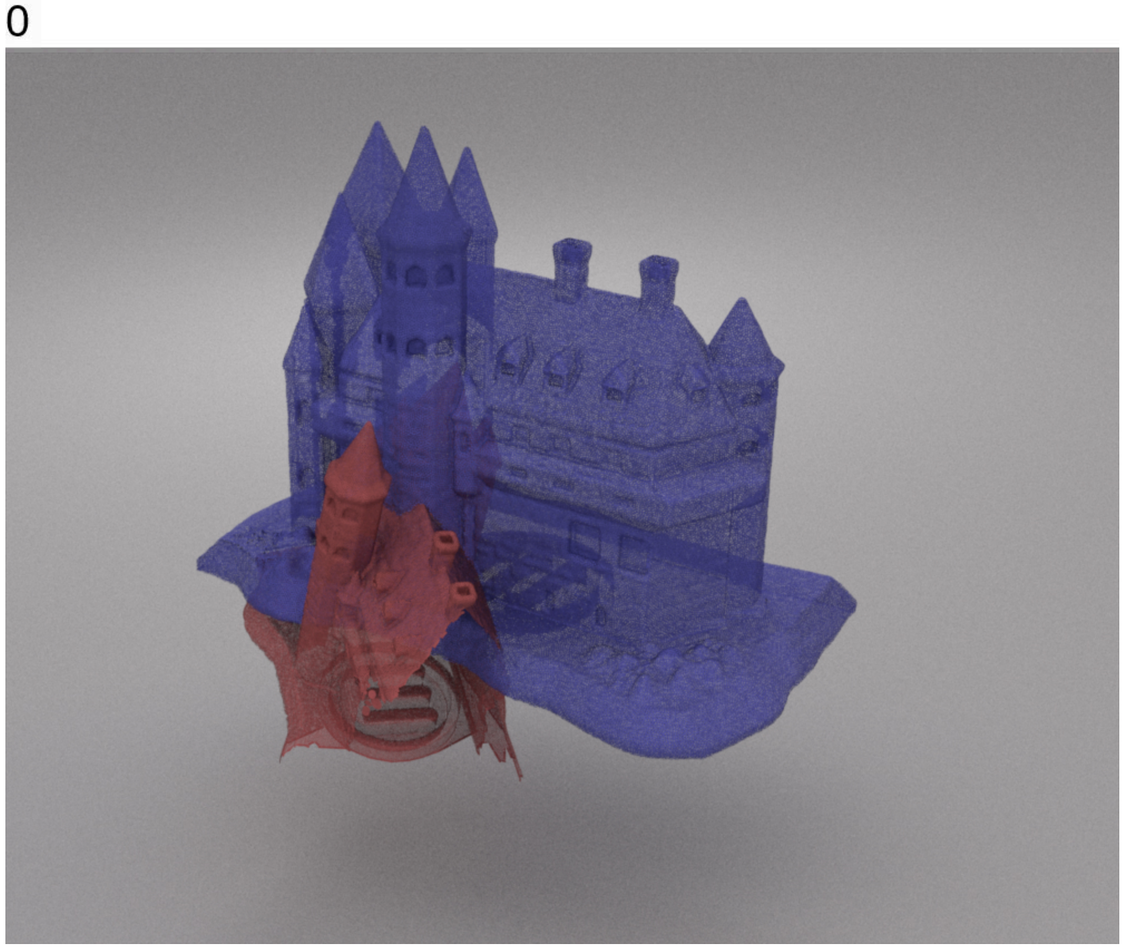
Our FIST algorithm
(2.29 s / iteration)

Source: 150k samples
Target: 200k samples

(input too large for iterative transport with network simplex)

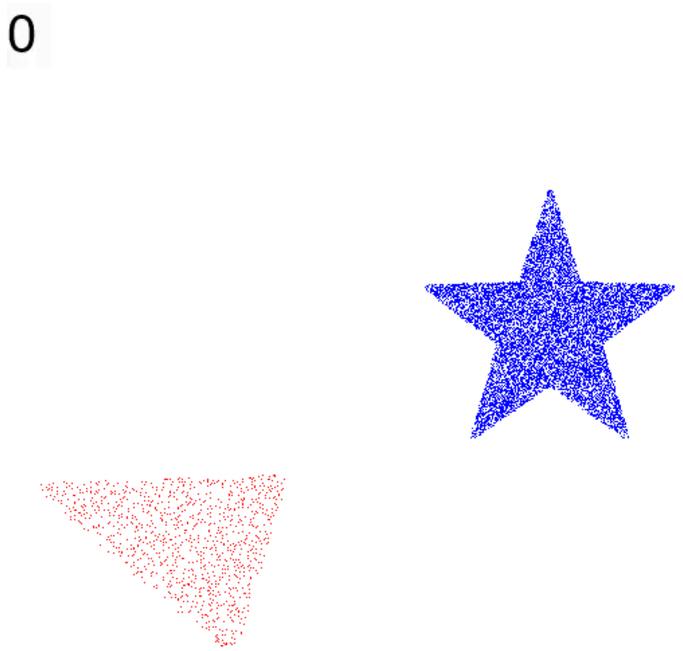


ICP
(0.09 s / iteration)

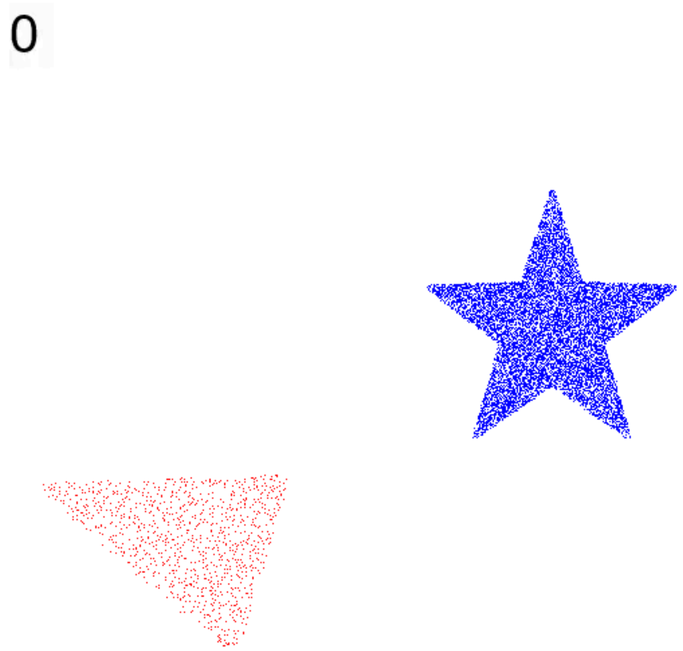


Our FIST algorithm
(2.18 s / iteration)

Failure case: the transport is optimal only on projections



Iterative Transport with Network Simplex



Our FIST algorithm

Conclusions

- Fast partial optimal transport in 1d
  - Quadratic-time algorithm (worst case)
  - Quasi-linear time decomposition

- Sliced Partial Optimal Transport
  - Same implementation for any dimension
  - Convergence in high dimension ?

- Fast Iterative Sliced Transport : variant of ICP

- Applications: point cloud registration, color matching
  - …. Machine learning ?