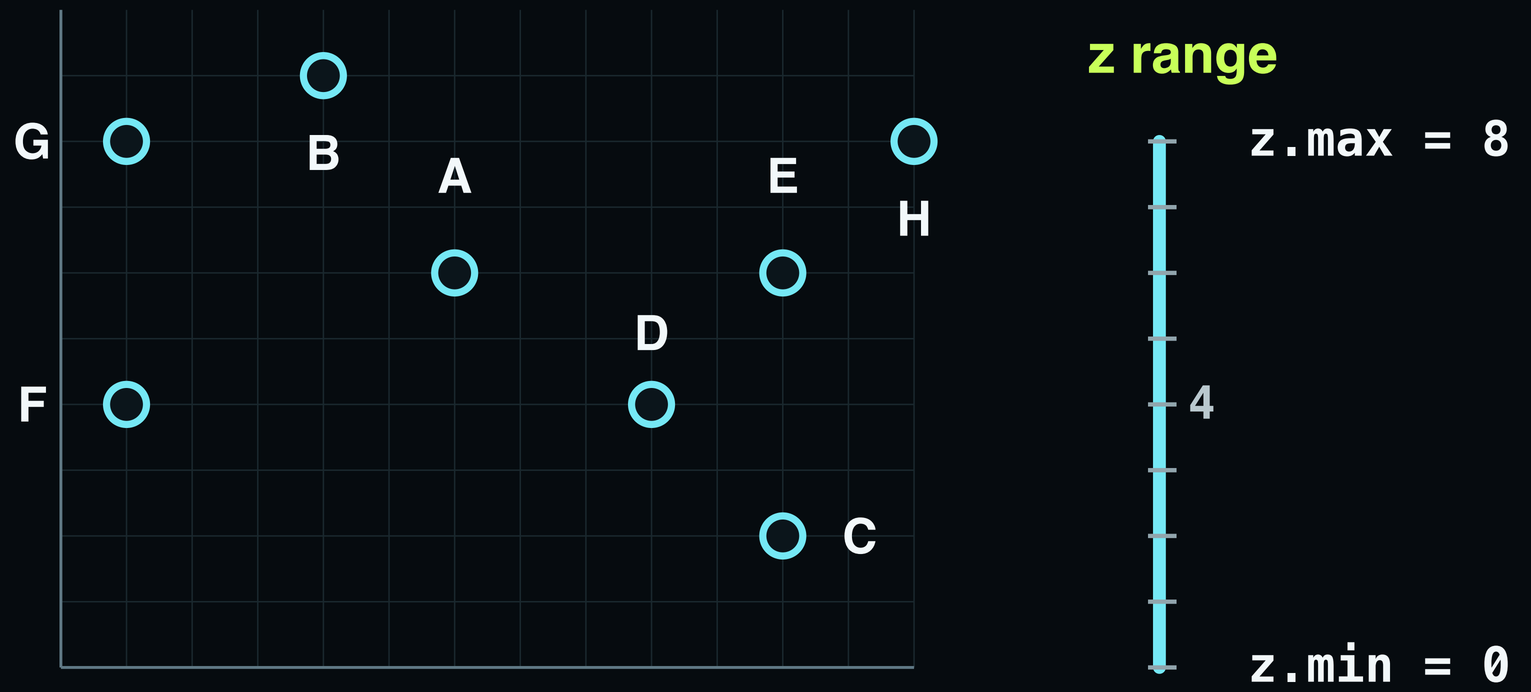
PROPAGATION ALGORITHMS FOR THE MINIMUM-DISTANCE CONSTRAINT

$\mathrm{min\_distance}(D, x, z, R)$

Mikael Z. Lagerkvist · ModRef 2026

CHOOSE FIVE SITES.



Euclidean distance

Maximise the shortest distance among the five selected sites.

MODELLING MIN DISTANCE PROBLEMS

DIRECT DECOMPOSITION

$$y_{ij} = D[x_i, x_j] \qquad z = \min_{i<j}(y_{ij})$$

- one distance variable per selected pair
- quadratic count of y variables
- quadratic number of propagators
- (potentially) quadratic domain sizes

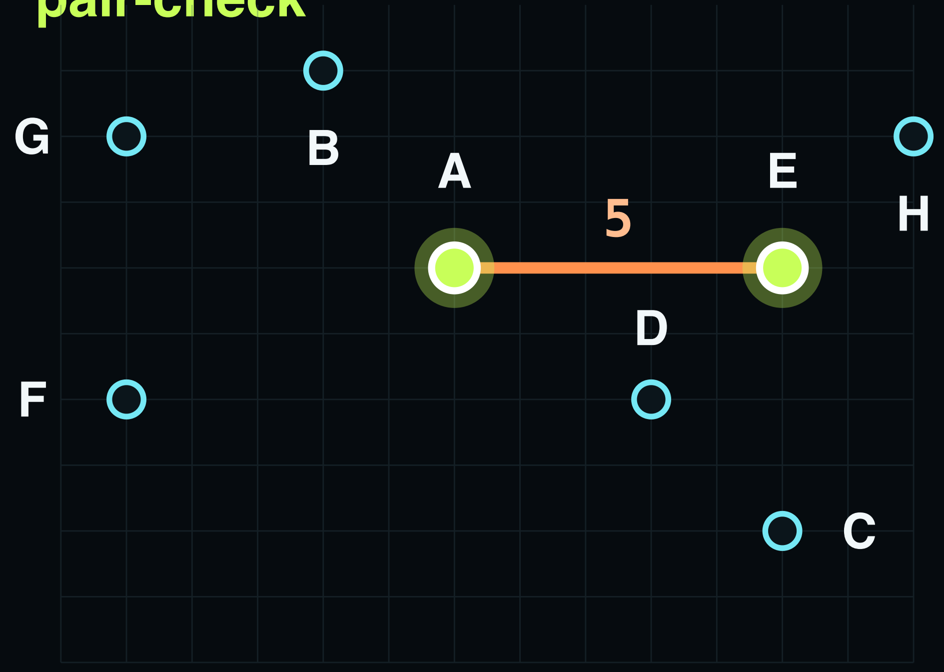
DEDICATED CONSTRAINT

$$min\_distance(D, x, z)$$

- single constraint
- global view of problem
- smarter propagation choices

PAIR-CHECK CHECKS ON ASSIGNMENT

pair-check



filled = selected · hollow = possible

z bounds

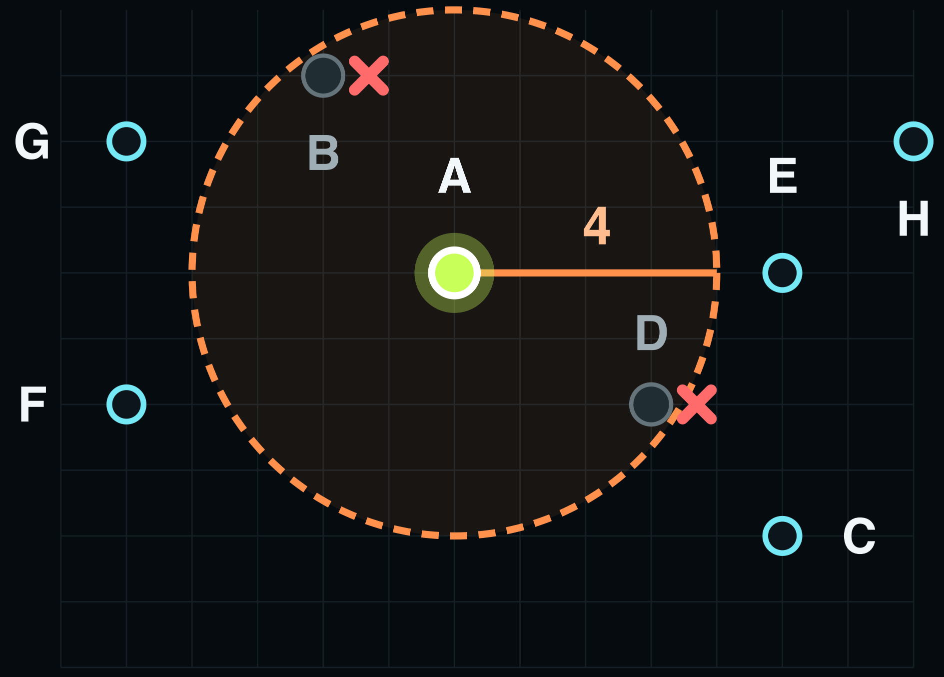
same scale

previous z.max = 7

```
z.max ← 5
z.min = 4
```

A–E is 5, so $z \le 5$. Another selected pair may still lower z.

FORWARD-BOUND SCANS ON ASSIGNMENT



z bounds

same scale

```
z.max = 5
z.min = 4
```

filled = selected · hollow = possible · × = removed

At T = 4, B and D are strictly too close to A. Equality stays feasible.

PAIR SUPPORT STARTS ON THE SITE GRAPH

pair support at T = 4

two variable domains

◆ variable x_i

dom(x_i)
= {A, E}

■ variable x_j

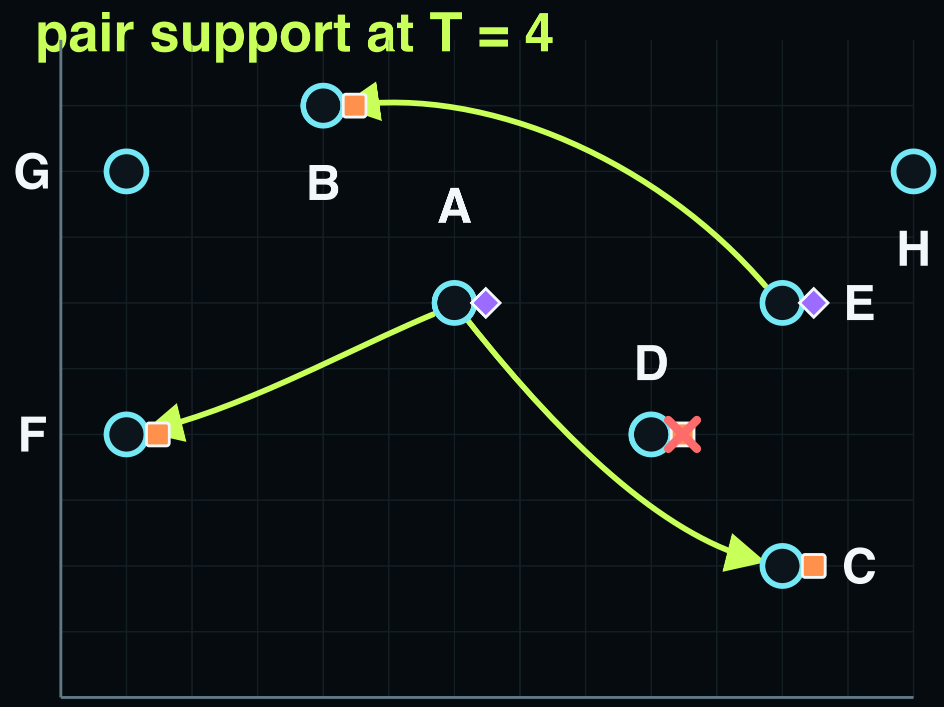
dom(x_j)
= {B, C, D, F}

support requires
distance ≥ 4

violet diamond = value of x_i · orange square = value of x_j · no sites selected

Can every value of x_j find a partner at distance at least 4?

PAIR SUPPORT REMOVES D



lime = support · red × = remove x_j = D · site D remains

supported values

◆ variable x_i

$\mathrm{dom}(x_i)$
= {A, E}

■ variable x_j

$\mathrm{dom}(x_j)$
= {B, C, ~~D~~, F}

D has no support
remove x_j = D

Thin curves witness B, C, and F; remove the unsupported x_j = D.

DETAIL: ONE RELATION, FOUR KERNELS

KERNEL	WHEN IT CAN ACT	WHAT IT DOES
Pair-check	both endpoints fixed	measures the selected pair
Forward-bound	one endpoint fixed	prunes inside the active radius
Full pair support	before either is fixed	finds a partner for every value
Advisors	after relevant domain events	revisits affected pairs and witnesses

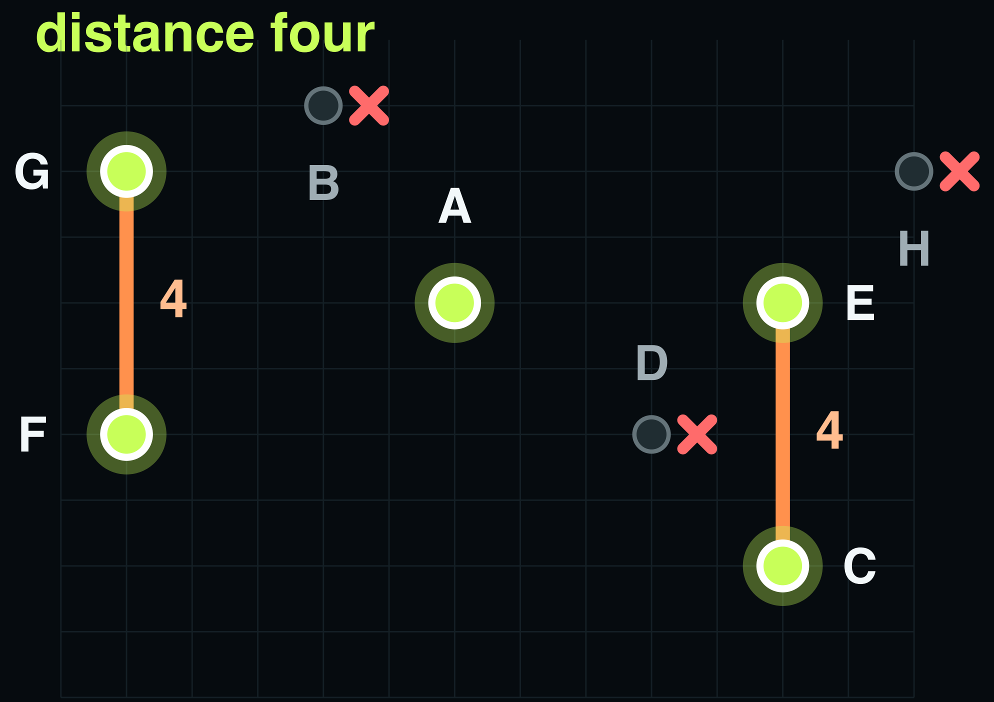
Pairwise and global variants package these kernels differently. Matching is a separate upper-bound pass.

DETAIL: ONE RELATION, THREE MODEL VIEWS

MODEL VIEW	DECISION VARIABLES	WHAT THE PROPAGATOR SEES
Indexed	p site-valued variables	one current site domain per position
Site-boolean	n selection booleans	selected and possible sites
Assignment-boolean	p × n assignment booleans	a Boolean encoding of each position domain

The pictures use the indexed view. The matching graph uses the union of its current position domains.

DISTANCE FOUR SOLUTION



z bounds
same scale

`z.max = 5`
`z.min = 4`

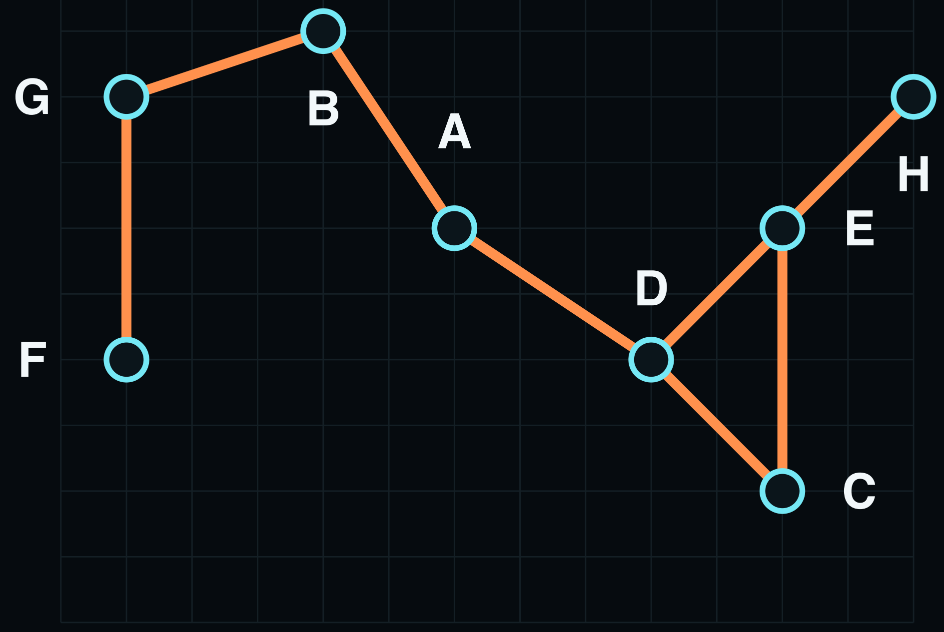
attains 4
can z reach 5?

filled = selected · muted × = no longer possible

Sites A, C, E, F, G reach distance 4. The next distance level is 5.

BUILD THE CONFLICTS FOR Z = 5

conflict graph for z = 5



edges that interfere

distance < 5

both endpoints cannot be selected together

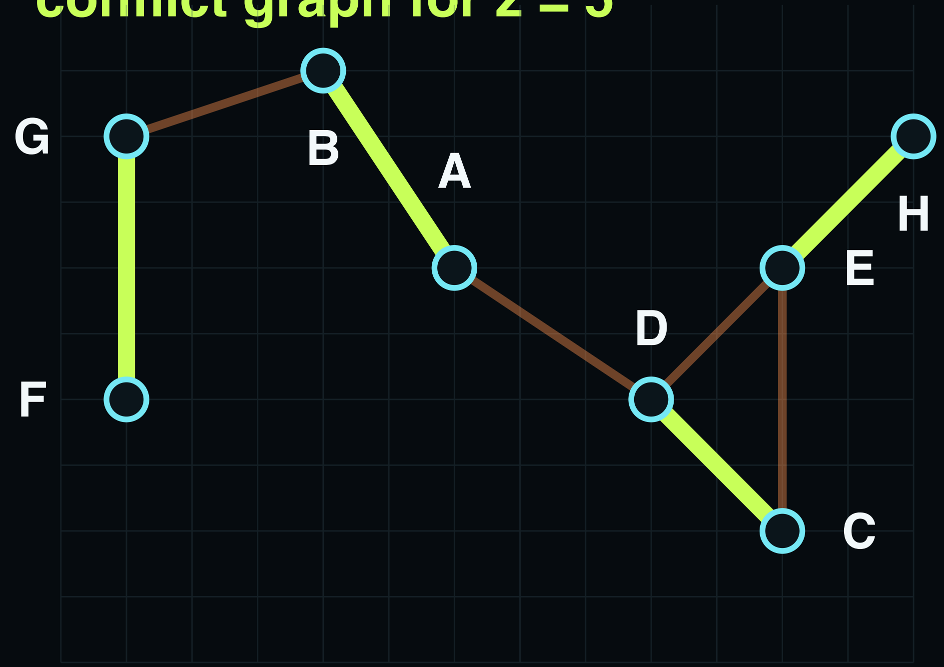
in a z = 5 solution

orange = cannot coexist if z = 5 · no matching marked yet

Join two sites when selecting both would interfere with $z = 5$.

A MATCHING PROVES Z < 5

conflict graph for z = 5



four disjoint conflicts

$|M| = 4$

omit at least one site
from each matched edge

$8 - 4 = 4 < p = 5$

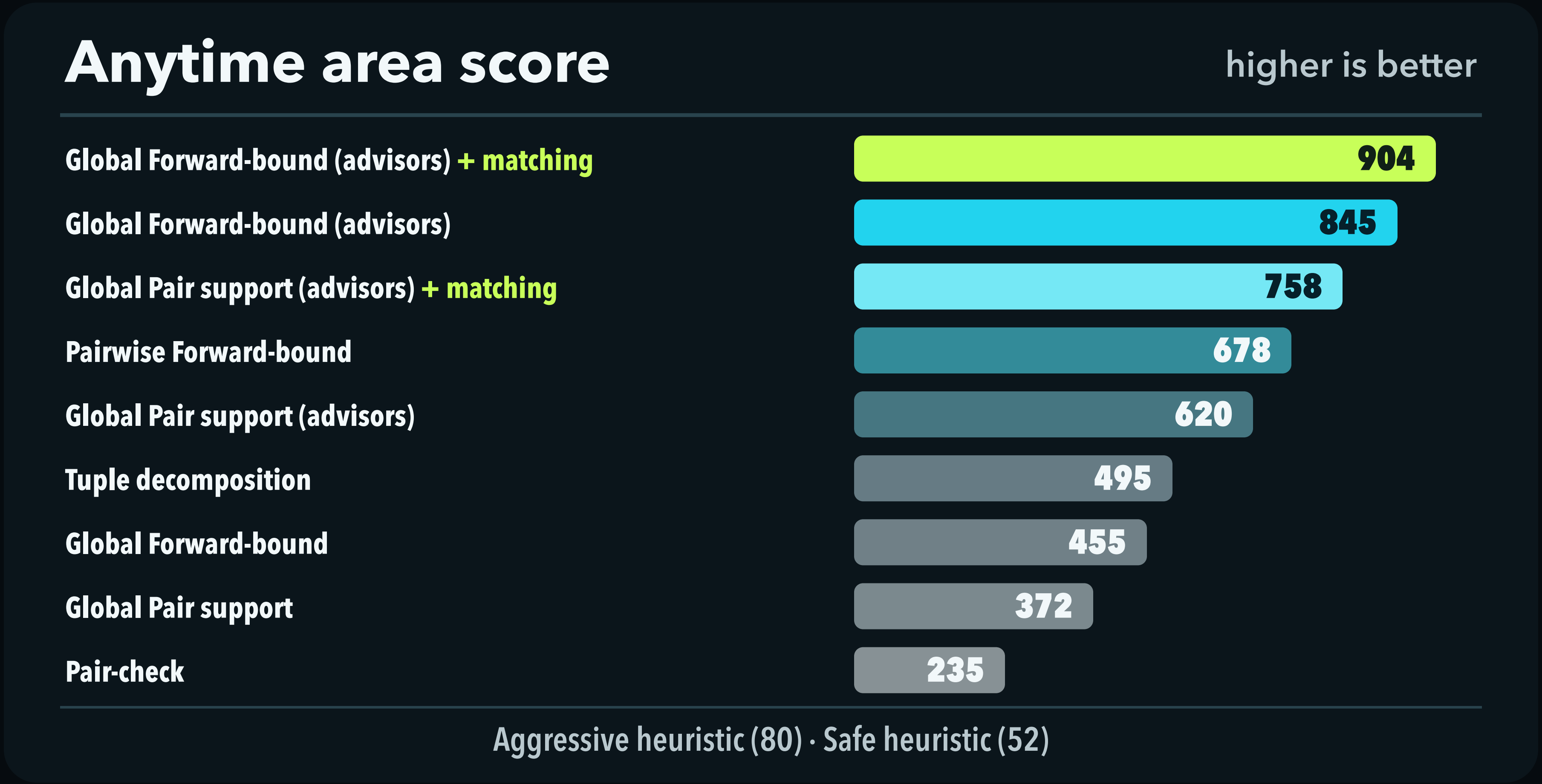
therefore z < 5

the next level is 4

orange = conflict · lime = four-edge matching · proof that z < 5

Four matched conflicts leave at most four sites, proving $z < 5$.

ADVISORS AND MATCHING ARE USEFUL



Some propagators are easy to implement

Advisors are useful

Avoid quadratic numbers of propagators and variables

Matching gives a useful upper bound for $min\_distance$