

Ten Points, Disjoint Disks



April 2026

A covering puzzle from the Monthly Mindbenders.¹

Problem

Your adversary chooses ten points on the plane, and your mission is to cover them with disjoint unit disks. If the points are close together, it is easy: one disk will cover them all. If they are far apart, you can cover each point with its own disk. Can you *always* do it?

Solution

Yes, always. The two easy cases in the statement are worth taking seriously, because they say where the difficulty is not. Ten points inside a disk of radius 1 need one disk. Ten points at mutual distances above 2 can each be given a private disk, since two unit disks are disjoint as soon as their centres are more than 2 apart. What is awkward is the middle: points a little more than 2 apart, where a disk placed greedily on one point blocks the disk you wanted for the next. Any method that treats the points one at a time will meet a configuration that defeats it, so the argument below does not treat them one at a time. It chooses all the disks at once, before looking at the points at all.

Take the densest packing of unit disks in the plane: centres on a triangular lattice with nearest-neighbour spacing 2, so that neighbouring disks touch without overlapping. The packing is disjoint by construction, and it stays disjoint however it is moved. Its fundamental cell is a rhombus of area $2\sqrt{3}$ holding exactly one disk of area π , so the fraction of the plane it covers is

$$\frac{\pi}{2\sqrt{3}} = 0.9068996 \dots,$$

and the uncovered part, the curved triangular gaps where three disks meet, has density

$$1 - \frac{\pi}{2\sqrt{3}} = 0.0931003 \dots$$

Now randomise. Let the adversary commit to the ten points $p_1, \dots, p_{10}$ first. Only then do you pick a vector T uniformly at random from one fundamental cell and

¹Peter Winkler, *Monthly Mindbenders*, National Museum of Mathematics, April 2026, momath.org/mindbenders, devised by Naoki Inaba and sent by Iwasawa Hirokazu.

translate the entire packing by T . Fix any one point p_i . Translating the packing by a uniform vector of the cell is the same as leaving the packing still and moving p_i to a uniformly random position within a cell, so

$$\Pr(p_i \text{ is uncovered}) = 1 - \frac{\pi}{2\sqrt{3}}$$

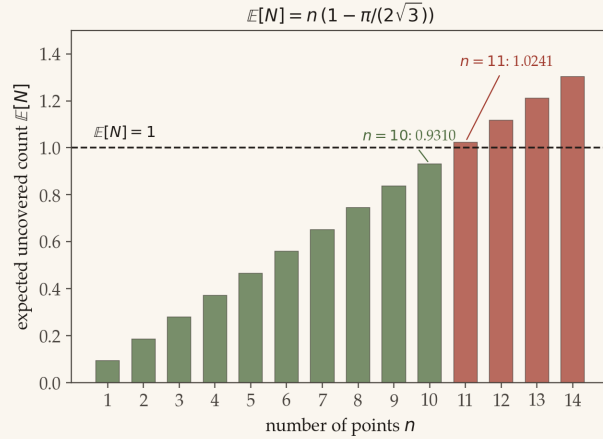
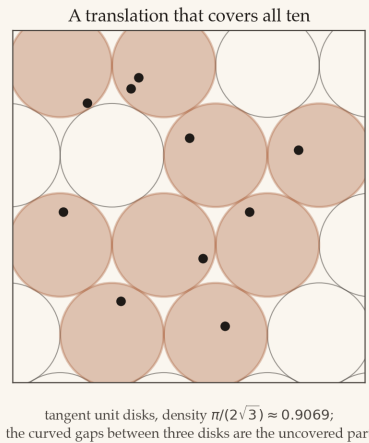
exactly, whatever p_i is. Let N be the number of the ten points left uncovered, that is, $N = \sum_{i=1}^{10} \mathbf{1}\{p_i \text{ uncovered}\}$. Linearity of expectation needs no independence between the indicators, which is the whole point: the ten points may sit in any arrangement at all, and they may have been chosen with full knowledge of your strategy. So

$$\mathbb{E}[N] = 10 \left(1 - \frac{\pi}{2\sqrt{3}} \right) = 0.93100 \dots < 1.$$

N is a non-negative integer. A non-negative integer variable whose mean is strictly below 1 must take the value 0 somewhere: if $N \geq 1$ always, then $\mathbb{E}[N] \geq 1$. Hence $\Pr(N = 0) > 0$, and in particular there exists a translation t for which the shifted packing covers all ten points at once. Its disks are pairwise disjoint. Throw away the ones that contain no point of the configuration, and what remains is a set of disjoint unit disks covering the ten points.

Yes: any ten points in the plane can be covered by disjoint unit disks.

The order of quantifiers is where the argument earns its keep. The adversary moves first and the bound $\mathbb{E}[N] < 1$ is then computed for that fixed configuration, so it holds for every configuration whatsoever. Had the adversary been allowed to see T before placing the points, it would simply drop all ten into the gaps and win.



Left: the packing of tangent unit disks, translated so that all ten adversary points fall inside disks; the disks in use are shaded, the rest drawn faintly, and the curved gaps between three tangent disks are what “uncovered” means. Right: the expected uncovered count grows linearly in n and crosses 1 between ten and eleven.

The threshold at ten

Nothing in the calculation used the number ten until the last line, so run it for n points:

$$\mathbb{E}[N] = n \left(1 - \frac{\pi}{2\sqrt{3}} \right) = 0.0931003 n,$$

which is below 1 exactly when $n \leq 10$, since $1/0.0931003 \dots = 10.741$. The two values on either side of the threshold are

$$n = 10 : 0.9310, \quad n = 11 : 1.0241,$$

so the puzzle is posed at the last value where this argument works at all. That is presumably why it says ten.

It is worth being exact about what happens at eleven. The bound fails, and nothing more. A mean of 1.0241 is perfectly consistent with $\Pr(N = 0)$ being large: a random variable can have mean above 1 and still be zero most of the time. So the failure at $n = 11$ is a failure of this proof, not a proof of failure, and it does not exhibit eleven points that defeat you. Whether every set of n points admits a disjoint-unit-disk cover for $n \geq 11$ is a genuinely harder question, and no claim is made here in either direction.

Two directions would be needed to settle it. To push the positive result past ten, one would have to improve on a fixed lattice: rotations as well as translations, non-lattice packings, or an argument that uses the geometry of the configuration rather than only the covering density. To settle it negatively, one would have to construct a configuration and prove that no arrangement of disjoint unit disks covers it, which the density bound cannot do, since it says nothing about any particular set of points. The numerical experiments below are consistent with covers existing comfortably beyond ten, but a search that always succeeds is evidence, not proof.

Python code

The listing builds the packing honestly (triangular lattice, spacing 2, a point counted as covered when it lies within distance 1 of some centre) and then does what the proof only asserts: it searches. Three families of ten-point configurations are tried, including one that puts every point at a gap centre of the untranslated packing, where the distance to the nearest disk centre is as large as it can be. For each, random translations are drawn until one covers all ten. The last line measures how often a single random translation succeeds for the hardest family, against the crude guarantee $1 - \mathbb{E}[N] = 0.069$ that the proof extracts by way of Markov's inequality.

```
import numpy as np

A1, A2 = np.array([2.0, 0.0]), np.array([1.0, np.sqrt(3)])
IJ = np.mgrid[-9:10, -9:10].reshape(2, -1).T.astype(float)
CEN = IJ @ np.vstack([A1, A2])      # triangular lattice, spacing 2
FRAC = np.pi / (2*np.sqrt(3))      # one disk per cell of area 2 sqrt 3
print(f"covered fraction pi/(2 sqrt 3) = {FRAC:.4f}")
for n in (10, 11):
    e = n*(1 - FRAC)
    print(f" n={n}: E[N] = {e:.4f} {'<' if e < 1 else '>='} 1")

def nunc(pts, t):                    # points outside every unit disk
```

```

    return int((np.linalg.norm(pts[:,None] - (CEN+t), axis=2).min(1) > 1).sum())

rng = np.random.default_rng(20260416)
H = np.vstack([CEN + [1, np.sqrt(3)/3], CEN + [1, -np.sqrt(3)/3]])
H = H[np.argsort((H**2).sum(1))][:10] # ten gap centres of the packing

def config(kind):
    # ten adversary points
    if kind == "box":
        return rng.uniform(0, 5, (10, 2))
    base = 1.05*CEN[np.argsort((CEN**2).sum(1))][:10] if kind[0] == "s" else H
    a = rng.uniform(0, 2*np.pi) # random rotation and offset
    R = np.array([[np.cos(a), -np.sin(a)], [np.sin(a), np.cos(a)]])
    return base @ R.T + rng.uniform(0, 2, 2)

for kind in ("box", "spacing 2.1", "gap sites"):
    tries, ok = [], 0
    for _ in range(400):
        P = config(kind)
        for k in range(1, 20001): # draw translations until one covers
            if nunc(P, rng.random()*A1 + rng.random()*A2) == 0:
                tries.append(k); ok += 1; break
    print(f"{kind}>11}: covered {ok}/400 configs,"
          f" mean {np.mean(tries):.1f} translations")

P = config("gap sites") # hardest family, many translations
hits = sum(nunc(P, rng.random()*A1 + rng.random()*A2) == 0 for _ in range(20000))
print(f"P(one random translation works) = {hits/20000:.3f}"
      f" vs bound 1-E[N] = {1 - 10*(1-FRAC):.3f}")
# covered fraction pi/(2 sqrt 3) = 0.9069
# n=10: E[N] = 0.9310 < 1
# n=11: E[N] = 1.0241 >= 1
# box: covered 400/400 configs, mean 2.7 translations
# spacing 2.1: covered 400/400 configs, mean 2.5 translations
# gap sites: covered 400/400 configs, mean 2.9 translations
# P(one random translation works) = 0.271 vs bound 1-E[N] = 0.069

```

Every one of the 1200 configurations was covered, and the average search took under three draws. The gap between 0.271 and 0.069 is the usual slack in this kind of bound: Markov's inequality is told only the mean of N and must assume the worst about everything else, whereas the real distribution of N puts most of its weight on 0 and 1. The proof needs none of that slack. All it needs is that the mean falls below one, which for ten points, and only just, it does.