The Urinal Packing Problem in Higher Dimensions

Shane Guan [1] **Blair Chen** [1] **Skanda Kaashyap** [1]
{xguan, blairc, skaashya}@andrew.cmu.edu

1. Introduction

In this paper, mere hours before the deadline, we investigate the optimal urinal usage so that no two pee-ers are within a certain distance of each other so as to minimize awkwardness and maximize flow rate. Naturally, one must be able to address this problem in just more than one spatial dimension (string theory predicts 11 spatial dimensions and makes no guarantees that urinals only exist in the 3 dimensions we inhabit) so we explore the previously unnamed problem known as the "Urinal Packing Problem in Higher Dimensions," abbreviated UrPP(in)HD. Formally, given a set of n urinals in a d-dimensional metric space, what is the greatest number of urinals that can be in use simultaneously without assigning two pee-ers to a pair of urinals that are within r distance of each other?

2. Related Work

To our knowledge, the only people who have considered a similar problem to the one we are considering are the nice folks over at xkcd. They considered the case where, the urinals are on a 1 dimensional line, and each person who enters the bathroom to pee chooses the urinal that is furthest from any other urinal in use. The good folks at xkcd determined that under this situation, the number of urinals that result in the greatest use percentage is $2^k + 1$.

Note that their situation is a bit different from our situation. In our situation, we maintain that the people using the urinals have a 0-1 comfortness score of using a urinal – as long as they are using a urinal that is further than r distance away from another urinal in use, they are happy. But in xkcd's scenario, each peeing person has a continuous gradient of comfortness score, inversely related to the distance to the nearest urinal in use. Now, personally we believe that the 0-1 comfortness score is more realistic, but to each their own.

[1]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

3. 1D Case for the 0-1 comfortness

Again, this situation is where the urinals lie on a line and as long as each pee-er is further than r distance from the nearest pee-er then they are happy. In this case, a greedy algorithm (presented below) achieves the optimal packing, and it has runtime $O(nlog(n))$.

```
def solve_UrPP_in_1D(urinal_positions ,r):
  urinal_positions =
    sort(urinal_positions)
  picked = [None]
  for u in urinal_positions:
    if picked[-1] is None or
      abs(picked[-1]-u)< r:
      picked.append(u)
  return len(picked)-1
```

Here we present a proof for this. Let's fix the input positions and assume they are in sorted increasing order already. Our proof idea is that the greedy algorithm cannot perform worse than any other algorithm. Let $GREEDY$ denote the greedy algorithm. Let u_k to be the kth urinal in the input.

We proceed by induction on n, the length of the prefix of the input. Clearly $GREEDY$ is optimal for the first $n = 2$ urinals. Our induction hypothesis will be that $GREEDY$ matches at least one optimal algorithm for the first n urinals in the input. Let OPT be the name of that optimal algorithm that matches $GREEDY$ for the first n urinals.

Consider the case that u_{n+1} is too close to the most recent urinal chosen by $GREEDY$. Then neither $GREEDY$ nor OPT will select u_{n+1}, so $GREEDY$ matches OPT for the first $n + 1$ urinals.

Consider the case that u_{n+1} is far. Then $GREEDY$ will select it. Now let's assume that this is a mistake and no optimal algorithm will select u_{n+1}. Let v be the first urinal after u_{n+1} that OPT selects. Clearly OPT can replace v with u_{n+1} while still remaining optimal, which contradicts our assumption that picking u_{n+1} is a mistake. Hence $GREEDY$ must match at least one optimal algorithm for the first $n + 1$ urinals.

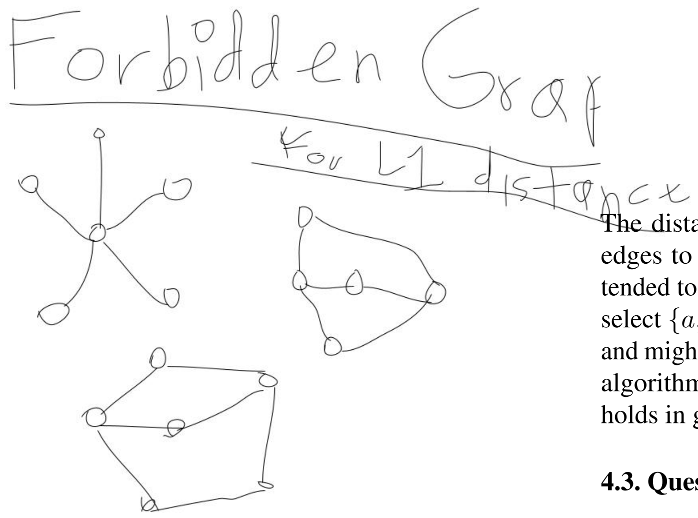
Since in both cases $GREEDY$ matches at least one optimal algorithm for the first $n + 1$ urinals, it follows by induction that $GREEDY$ must be optimal for the entire input.

4. 2D Case

In this case, the urinals lie on the 2D plane instead of simply on a line. At first glance, this might not seem to be much more different than the 1D case, but actually it's a lot different because you have 2 dimensions to be greedy in. How would you even define a greedy algorithm in the 2d case?

The first thing we can do is to reduce this problem to that of Max Independent Set, which is to find the largest set of vertices in a graph which are all disjoint from one another. How you ask? Merely by transforming the original input of urinal positions on the 2d plane to a graph. Each urinal will have its corresponding node, and two nodes have an edge between them iff the two corresponding urinals are close (in whatever metric we're using). Then if we can find the maximal independent set in the graph, we can find a maximal packing of the urinal usage.

But, clearly, as the wikipedia page suggests, the problem of Max Independent Set is NP-Hard, which means it is pretty hard. So are we at a loss? We do not think so. First, not every arbitrary graph corresponds to a set of urinal positions on the 2d plane. For instance, suppose our distance metric was the L1 norm (so basically continuous taxicab distance), then the following graphs do not have a corresponding urinal position set.



It is easy to see why the above graphs are forbidden for L1 distance in UrPP-2D. The 5 leaf hub is illegal because the most leaves you can have on a hub is 4. The graph directly to the right of the 5 leaf hub is illegal because a 3 leaf hub implies that the hub is in the convex hull of the 3 leaves, yet there is another node that is the convex hull of the same 3 leaves while still being far from the first hub. The last graph is illegal for similar reasons.

4.1. Conjecture:

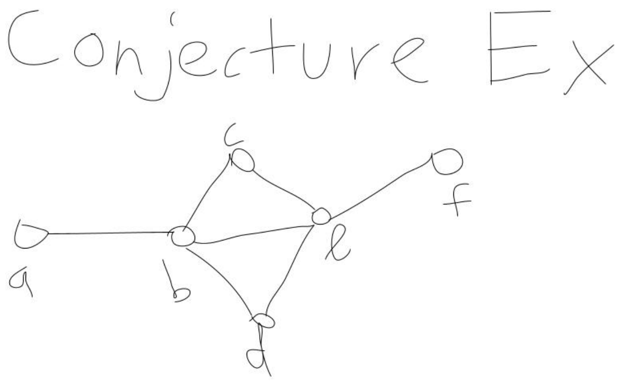
For the UrPP-2D using the L1 metric, the greedy algorithm is at least

$$\frac{1}{2}$$

approximative of the optimal packing. We define the greedy algorithm as the algorithm that starts at the left-most point and just greedily selects the next nearest point to the points already chosen, breaking ties arbitrarily.

4.2. Support for the Conjecture:

Our conjecture does not stand without reason. Behold, the following example will elicit why we think the worst the greedy algorithm can do is 50%.



The distances are not drawn to scale so we included the edges to help the reader understand which urinals we intended to be close to each other. The optimal algorithm will select $\{a, c, d, f\}$ while the greedy algorithm will start at a and might end up choosing $\{a, e\}$. So in this case the greedy algorithm performs half as good as optimal. We think this holds in general.

4.3. Questions to Answer

- How would you even begin to solve UrPP-2D?
- this thing reduces to max independent set. What is the class of graphs that UrPP-2D reduces to? Is it easier to solve max-indp on that class of graphs?
- how about higher dimensions? other metrics?
- greedy might not perform all that terribly. How would

one even begin to simulate such a thing? Take a fixed square and uniformly sample to make a urinal position set? The pdf that you choose will affect the class of graphs that you can make

- but it still might be interesting to see how greedy performs on these different pdfs. how would you parameterize the different pdfs?

- in the meantime, could you figure out the approximation ability of the greedy? is greedy epsilon approximative for some fixed epsilon?

4.4. Manhattan Distance

Here's an algorithm that might work:

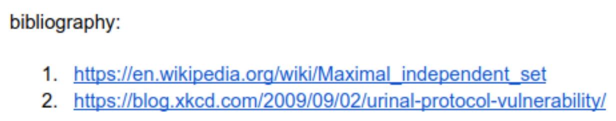
```
def solve_UrPP_in_2d_manhattan(urinal_positions,r):
  divide the input positions into sections of
  close points, where each point in a section
  is close to another point within the
  same section, but is
  far from all points in another section

  for each section:
    overlay a grid of sidelength r,
    with the leftmost point
    coincident with
    the boundary of a gridbox

    using any enumeration
    of the gridboxes
    in this section,
    greedily select points

  return the union of all points
  selected for each section
```

5. Conclusion

Our conclusion is that this problem gets actually kinda hard after you leave one dimension, but we still got somewhere which is impressive. We cook up some food for thought in section 3.4 and hope future studies explore these avenues further. We foresee this exploration having serious implications on humanity so long as pee-ers continue to draw breath and stay hydrated.

6. Bibliography

Ok here it is:


bibliography:

1. https://en.wikipedia.org/wiki/Maximal_independent_set
2. https://blog.xkcd.com/2009/09/02/urinal-protocol-vulnerability/