Advice in the Context of Some Geometric Problems

(Matching and Packing)

Shahin Kamali (U. Manitoba)

August 28, 2020

Joined work with Prosenjit Bose[1], Paz Carmi[2], Stephane Durocher[3] and Arezoo Sajadpour[3]

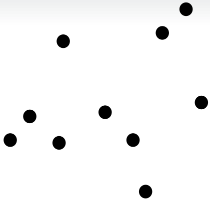
[1] Carleton University
[2] Ben-Gurion University
[3] University of Manitoba

Geometric Matching

https://www.houseandgarden.co.uk/gallery/animals-cities-coronavirus-lockdown

Monochromatic Non-crossing Matching

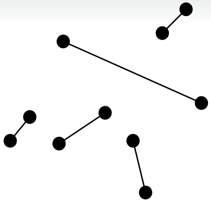
- The input is a set of n points in general position.



OLAWA 2020
Online Algorithms with Advice

Monochromatic Non-crossing Matching

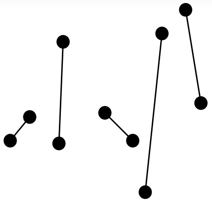
- The input is a set of n points in general position.
- The goal is to form a maximum matching s.t. the line segments between the matched points do not intersect.



OLAWA 2020
Online Algorithms with Advice

Monochromatic Non-crossing Matching

- The input is a set of *n* points in general position.
- The goal is to form a maximum matching s.t. the line segments between the matched points do not intersect.
    - In the offline setting, one can sort items (say by their *x*-coordinate) and match consecutive points.



Online Monochromatic Matching

- In the online setting, points arrive one by one.

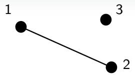
Online Monochromatic Matching

- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.

OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

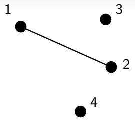
- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
  - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.



OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

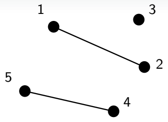
- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
  - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.
- Not all points can be matched in the online setting.
  - Given an adversarial sequence of n points, how many points can be matched?



OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

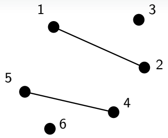
- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
  - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.
- Not all points can be matched in the online setting.
  - Given an adversarial sequence of n points, how many points can be matched?



OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

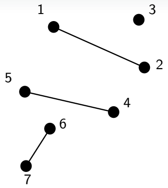
- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
    - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.
- Not all points can be matched in the online setting.
    - Given an adversarial sequence of n points, how many points can be matched?



OLAWA 2020 Online Algorithms with Advice

Online Monochromatic Matching

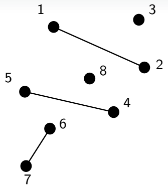
- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
    - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.
- Not all points can be matched in the online setting.
    - Given an adversarial sequence of n points, how many points can be matched?



OLAWA 2020
Online Algorithms with Advice

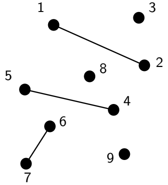
Online Monochromatic Matching

- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
  - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.
- Not all points can be matched in the online setting.
  - Given an adversarial sequence of n points, how many points can be matched?



Online Monochromatic Matching

- In the online setting, points arrive one by one.
- Upon arrival of a point p an algorithm can match p with an existing unmatched point or leave it unmatched.
  - Greedy algorithms do not leave a point unmatched if they can match it with some existing point.
- Not all points can be matched in the online setting.
  - Given an adversarial sequence of n points, how many points can be matched?



OLAWA 2020
Online Algorithms with Advice

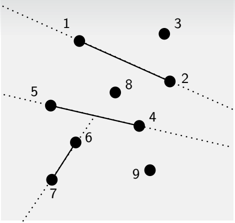
Online Monochromatic Matching

- In the worst case, a greedy algorithm has one unmatched point per each pair of matched points (it matches roughly $2n/3$ points).

OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

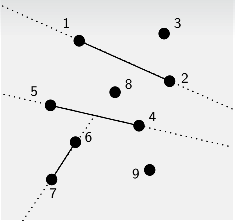
- In the worst case, a greedy algorithm has one unmatched point per each pair of matched points (it matches roughly $2n/3$ points).
    - The proof is based on partitioning the plane based on an extension of the greedy line segments.
    - There is at most one unmatched point per each convex partition.



OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

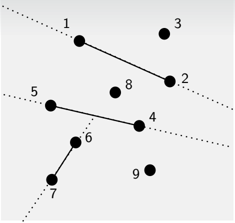
- In the worst case, a greedy algorithm has one unmatched point per each pair of matched points (it matches roughly $2n/3$ points).
  - The proof is based on partitioning the plane based on an extension of the greedy line segments.
  - There is at most one unmatched point per each convex partition.
- An adversarial argument shows that no deterministic algorithm can match more than $2n/3$ points in the worst case.



OLAWA 2020
Online Algorithms with Advice

Online Monochromatic Matching

- In the worst case, a greedy algorithm has one unmatched point per each pair of matched points (it matches roughly $2n/3$ points).
  - The proof is based on partitioning the plane based on an extension of the greedy line segments.
  - There is at most one unmatched point per each convex partition.
- An adversarial argument shows that no deterministic algorithm can match more than $2n/3$ points in the worst case.
- **Greedy algorithms are the optimal deterministic online algorithms.**



OLAWA 2020

Online Algorithms with Advice

Monochromatic Matching with Advice

- **Question 1: How many advice bits are necessary/sufficient to match all points?**

OLAWA 2020
Online Algorithms with Advice

Monochromatic Matching with Advice

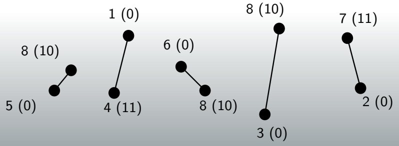
- **Question 1: How many advice bits are necessary/sufficient to match all points?**
- Upper bounds: $1.5n$ bits are sufficient.

OLAWA 2020
Online Algorithms with Advice

Monochromatic Matching with Advice

- **Question 1: How many advice bits are necessary/sufficient to match all points?**

- Upper bounds: $1.5n$ bits are sufficient.
  - Mimic an optimal matching based on x-coordinates.
  - For each point encode whether its partner I) appears later II) appears earlier on its left III) appears earlier on its right.
  - The online algorithm matches p with the leftmost point on its right or rightmost point on its left if its partner appears earlier.



OLAWA 2020
Online Algorithms with Advice

Monochromatic Matching with Advice

- **Question 1: How many advice bits are necessary/sufficient to match all points?**

OLAWA 2020
Online Algorithms with Advice

Monochromatic Matching with Advice

- **Question 1: How many advice bits are necessary/sufficient to match all points?**
- Can we use a reduction from the binary-guessing problem [Böckenhauer et al., 2014] to show a lower bound of $\Omega(n)$?
  - Maybe; we tried and failed!

OLAWA 2020 Online Algorithms with Advice

Monochromatic Matching with Advice

- **Question 1: How many advice bits are necessary/sufficient to match all points?**
- Can we use a reduction from the binary-guessing problem [Böckenhauer et al., 2014] to show a lower bound of $\Omega(n)$?
  - Maybe; we tried and failed!
- At least $\Omega(\log n)$ bits are necessary.
  - The proof is based on defining a family of n sequences with similar prefix and points on the boundary of a circle.

OLAWA 2020
Online Algorithms with Advice

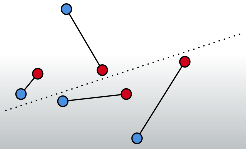
Monochromatic Matching with Advice

- **Question 1: How many advice bits are necessary/sufficient to match all points?**
- Can we use a reduction from the binary-guessing problem [Böckenhauer et al., 2014] to show a lower bound of $\Omega(n)$?
  - Maybe; we tried and failed!
- At least $\Omega(\log n)$ bits are necessary.
  - The proof is based on defining a family of n sequences with similar prefix and points on the boundary of a circle.

Question 2: How many points can be matched if the size of advice is a constant?

Bichromatic Non-crossing Matching

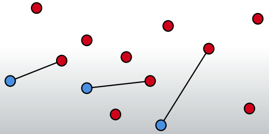
- In an input of $2n$ points, assume half are blue and half are red.
  - Each point should be matched to a point of opposite color.
- In the offline setting (ghost and ghost-buster problem), one can match (almost) all the points.
  - Find the ham-sandwich line that bisects the blue and red points, and apply a divide and conquer approach.



OLAWA 2020
Online Algorithms with Advice

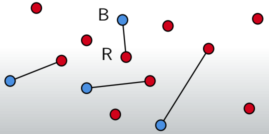
Bichromatic Non-crossing Matching

- In the online setting, we assume n red points are given and n blue points arrive in an online manner.



Bichromatic Non-crossing Matching

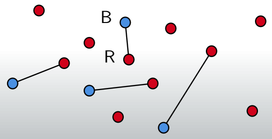
- In the online setting, we assume n red points are given and n blue points arrive in an online manner.
  - Greedy Median: match a blue point B with a red point R such the line BR bisects all suitable red points for B.



OLAWA 2020
Online Algorithms with Advice

Bichromatic Non-crossing Matching

- In the online setting, we assume n red points are given and n blue points arrive in an online manner.
  - Greedy Median: match a blue point B with a red point R such the line BR bisects all suitable red points for B.
  - Greedy Median matches at least $\Omega(\log n)$ points, and no deterministic algorithm can do better.



OLAWA 2020
Online Algorithms with Advice

Bichromatic Matching with Advice

- **Question 3: How many advice bits are necessary/sufficient to match all points?**

OLAWA 2020
Online Algorithms with Advice

Bichromatic Matching with Advice

- **Question 3: How many advice bits are necessary/sufficient to match all points?**
  - $\Theta(n \log n)$ bits are both sufficient and necessary.

OLAWA 2020
Online Algorithms with Advice

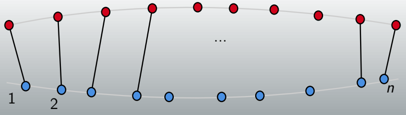
Bichromatic Matching with Advice

- **Question 3: How many advice bits are necessary/sufficient to match all points?**
  - $\Theta(n \log n)$ bits are both sufficient and necessary.
  - Upper bound is trivial: for each blue point encode exactly what red point it is matched to in an optimal packing.

OLAWA 2020
Online Algorithms with Advice

Bichromatic Matching with Advice

- **Question 3: How many advice bits are necessary/sufficient to match all points?**
  - $\Theta(n \log n)$ bits are both sufficient and necessary.
  - Upper bound is trivial: for each blue point encode exactly what red point it is matched to in an optimal packing.
  - Lower bound is based on points appearing on the exterior of a circle.
    - Consider a family of $n!$ sequences, each associated with ordering of blue items labelled from left to right; each sequence need a different advice. That is, $\Omega(n!)$ bits of advice is required to separate two members of the family.



OLAWA 2020
Online Algorithms with Advice

Non-crossing Matching Summary

- Monochromatic setting:
  - All points can be matched in the offline setting.
  - The best deterministic algorithm matches roughly two-third of points in the worst case.
  - In order to match all n points, at least $\Omega(\log n)$ and at most $O(n)$ bits of advice are needed.

OLAWA 2020
Online Algorithms with Advice

Non-crossing Matching Summary

- Monochromatic setting:
  - All points can be matched in the offline setting.
  - The best deterministic algorithm matches roughly two-third of points in the worst case.
  - In order to match all n points, at least $\Omega(\log n)$ and at most $O(n)$ bits of advice are needed.
- Bichromatic setting:
  - All points can be matched in the offline setting (almost).
  - The best deterministic algorithm matches only $O(\log n)$ points in the worst case.
  - In order to match all n points, $\Theta(n \log n)$ bits are necessary and sufficient.

OLAWA 2020
Online Algorithms with Advice

Non-crossing Matching Summary

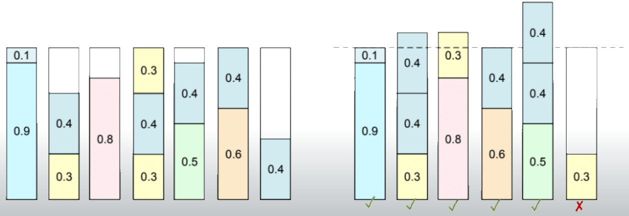
- Monochromatic setting:
  - All points can be matched in the offline setting.
  - The best deterministic algorithm matches roughly two-third of points in the worst case.
  - In order to match all n points, at least $\Omega(\log n)$ and at most $O(n)$ bits of advice are needed.
- Bichromatic setting:
  - All points can be matched in the offline setting (almost).
  - The best deterministic algorithm matches only $O(\log n)$ points in the worst case.
  - In order to match all n points, $\Theta(n \log n)$ bits are necessary and sufficient.
- Other variants: more colors, other objective functions (e.g., minimizing the length of segments), and replacing points with "objects" (e.g., convex polygons) [e.g., Aloupis et al. 2010].

Geometric Packing

https://www.cnbc.com/2020/04/10/coronavirus-empty-streets-around-the-world-are-attracting-wildlife.html

Bin Packing & Bin Covering

- The input is a multiset of n items with sizes in the range $(0, 1]$
  - **Bin packing:** place items into a **minimum** number of bins s.t. the total size of items in each bin is **at most** 1.
  - **Bin covering:** cover a **maximum** number of bins s.t. the total size of items in each bin is **at least** 1.



bin packing

bin covering

OLAWA 2020
Online Algorithms with Advice

Bin Packing & Bin Covering

- **Offline setting:**
  - Both bin packing and bin covering are NP-hard, and asymptotic polytime approximation schemes exist for both bin packing [Hoberg and Rothvoss, 2017] and bin covering [Jansen and Solis-Oba, 2003].

OLAWA 2020
Online Algorithms with Advice

Bin Packing & Bin Covering

- **Offline setting:**
  - Both bin packing and bin covering are NP-hard, and asymptotic polytime approximation schemes exist for both bin packing [Hoberg and Rothvoss, 2017] and bin covering [Jansen and Solis-Oba, 2003].
- **Online setting:**
  - The best bin packing algorithm has an asymptotic competitive ratio in the range (1.54278,1.57829] [Balogh et al., 2012, Balogh et al., 2018] .
  - The best bin covering algorithm has a competitive ratio of 0.5 [Csirik and Totik, 1988].

OLAWA 2020
Online Algorithms with Advice

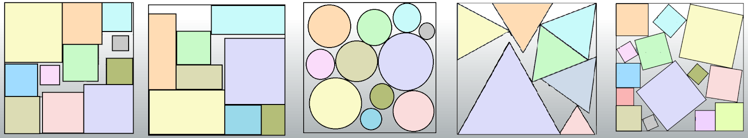
Bin Packing & Bin Covering

- **Offline setting:**
  - Both bin packing and bin covering are NP-hard, and asymptotic polytime approximation schemes exist for both bin packing [Hoberg and Rothvoss, 2017] and bin covering [Jansen and Solis-Oba, 2003].
- **Online setting:**
  - The best bin packing algorithm has an asymptotic competitive ratio in the range (1.54278,1.57829] [Balogh et al., 2012, Balogh et al., 2018] .
  - The best bin covering algorithm has a competitive ratio of 0.5 [Csirik and Totik, 1988].
- **Advice setting:**
  - $O(1)$ bits of advice suffices for a bin packing algorithm to achieve a competitive ratio strictly better than all online algorithms (ratio 1.4702) [Angelopoulos., 2015].
  - $\Theta(\log\log n)$ bits are necessary and sufficient for a bin covering algorithm to achieve a competitive ratio better than all online algorithms (a c.r. of $0.5\bar{3}$) [Boyar et al., 2019].

OLAWA 2020
Online Algorithms with Advice

2-dimensional Bin Packing

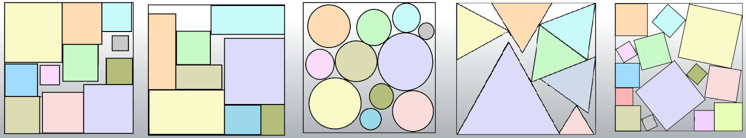
- Bins are assumed to be squares of side-length 1, while items can be squares, rectangles, disks, triangles, etc. of different sizes.



OLAWA 2020
Online Algorithms with Advice

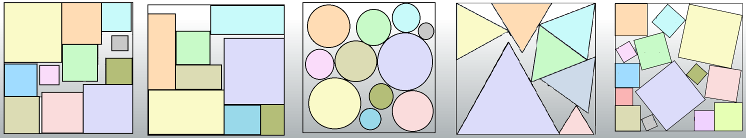
2-dimensional Bin Packing

- Bins are assumed to be squares of side-length 1, while items can be squares, rectangles, disks, triangles, etc. of different sizes.
- Offline setting: almost problems are NP-hard.



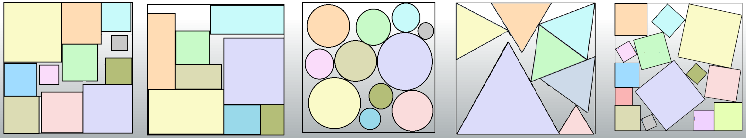
2-dimensional Bin Packing

- Bins are assumed to be squares of side-length 1, while items can be squares, rectangles, disks, triangles, etc. of different sizes.
- Offline setting: almost problems are NP-hard.
  - There is an APTAS for packing squares while there is inapproximability results for packing rectangles [Bansal et al. 2006].



2-dimensional Bin Packing

- Bins are assumed to be squares of side-length 1, while items can be squares, rectangles, disks, triangles, etc. of different sizes.
- Offline setting: almost problems are NP-hard.
  - There is an APTAS for packing squares while there is inapproximability results for packing rectangles [Bansal et al. 2006].
  - In the presence of rotation, the problem might be ∃R-hard, [Abrahamsen et al. 2019] but an APTAS exists when bins are **augmented** [Kamali and Nikbakht, 2020].



OLAWA 2020
Online Algorithms with Advice

Online 2-dimensional Bin Packing

- It is harder to close/tighten the gap between upper and lower bounds for the competitive ratio (compared to the 1-dimensional bin packing).
    - **Square packing:** the competitive ratio of the best algorithm is in the range (1.6406, 2.1187] [Epstein and van Stee, 2005, Han et al., 2010].
    - **Rectangle packing without rotation:** the competitive ratio of the best algorithm is in the range (1.91004, 2.5545] [Epstein, 2019, Han et al., 2011].
    - **Rectangle packing with rotation:** the competitive ratio of the best algorithm is in the range (2.45, 2.535356] [Epstein, 2010].
    - **Equilateral triangle packing:** the competitive ratio of the best algorithm is in the range (1.509, 2.474] [Kamali et al., 2015].

OLAWA 2020 Online Algorithms with Advice

2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?

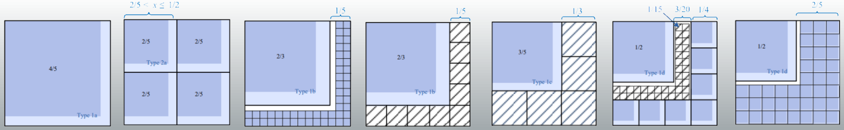
2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?
  - Square packing: an algorithm that receives advice of size $O(\log n)$ can achieve a competitive ratio of 1.84 [Kamali and López Ortiz, 2014].

OLAWA 2020 Online Algorithms with Advice

2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?
  - Square packing: an algorithm that receives advice of size $O(\log n)$ can achieve a competitive ratio of 1.84 [Kamali and López Ortiz, 2014].
    - Classify square-items based on their sizes, and receive the number of items in each class as advice.
    - Round-up the size of items, except the smallest class (tiny items), and create a partial packing.
    - Tiny items are placed in the remaining area in the partial packing.



2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?
  - **Square packing:** $O(\log n)$ bit suffices to achieve a c.r. of 1.81, compared to that of 2.1187 of the best existing algorithm.

OLAWA 2020
Online Algorithms with Advice

2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?
  - **Square packing:** $O(\log n)$ bit suffices to achieve a c.r. of 1.81, compared to that of 2.1187 of the best existing algorithm.
    - It is expected can achieve the same competitive ratio with $O(1)$ bits of advice (ongoing research).
    - Instead of encoding the exact number of items in each class in $O(\log n)$ bits, encode their frequency.

OLAWA 2020 Online Algorithms with Advice

2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?
  - **Square packing:** $O(\log n)$ bit suffices to achieve a c.r. of 1.81, compared to that of 2.1187 of the best existing algorithm.
    - It is expected can achieve the same competitive ratio with $O(1)$ bits of advice (ongoing research).
    - Instead of encoding the exact number of items in each class in $O(\log n)$ bits, encode their frequency.
  - The problem remains open for other geometric packing problems (e.g, rectangle packing).

OLAWA 2020
Online Algorithms with Advice

2-dimensional Bin Packing with Advice

- **Question 4:** How many bits of advice is required to achieve a competitive ratio strictly better than all (existing) online algorithms?
  - **Square packing:** $O(\log n)$ bit suffices to achieve a c.r. of 1.81, compared to that of 2.1187 of the best existing algorithm.
    - It is expected can achieve the same competitive ratio with $O(1)$ bits of advice (ongoing research).
    - Instead of encoding the exact number of items in each class in $O(\log n)$ bits, encode their frequency.
  - The problem remains open for other geometric packing problems (e.g, rectangle packing).
- What about geometric bin covering?

Conclusions

https://www.cnbc.com/2020/04/10/
coronavirus-empty-streets-around-the-world-are-attracting-wildlife.html

OLAWA 2020
Online Algorithms with Advice

Concluding Remarks

- Some geometric problems that are trivial in the offline setting have "interesting" nature when studied under online setting, e.g., non-crossing matching problems.

OLAWA 2020 Online Algorithms with Advice

Concluding Remarks

- Some geometric problems that are trivial in the offline setting have "interesting" nature when studied under online setting, e.g., non-crossing matching problems.
- Some combinatorial "1-dimensional" problems can be extended to geometric problems that can be studied in the online setting with interesting advice complexity.
    - Bin packing, bin covering, knapsack problem (e.g., [Chen et al. 2011], [Böckenhauer et al., 2014]), and dual bin packing (e.g., [Renault, 2017], [Borodin et al., 2018]).