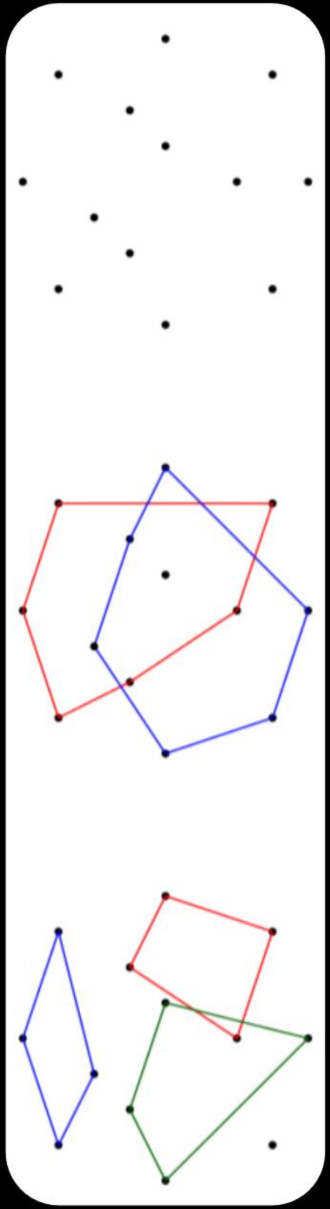
Bottleneck Convex Subsets: Finding *k* Large Convex Sets in a Point Set

[1] Stephane Durocher, [2] J. Mark Keil,
[3] Saeed Mehrabi, [2] Debajyoti Mondal

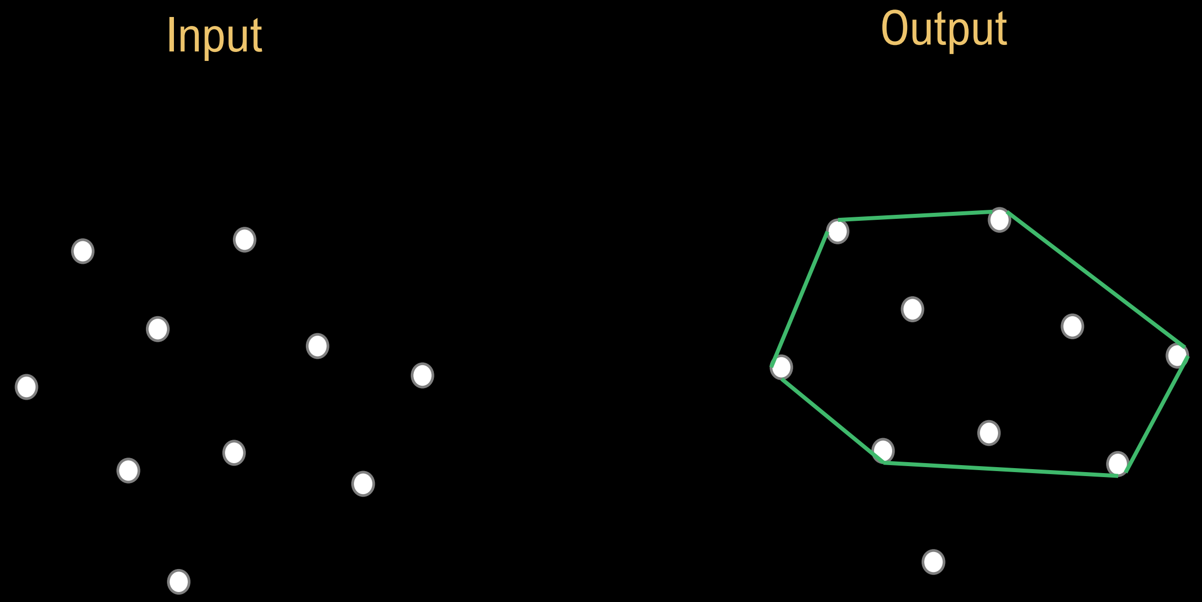
[1] University of Manitoba, Canada
[2] University of Saskatchewan, Canada
[3] Carleton University, Ottawa, Canada

1 University of Manitoba
2 University of Saskatchewan
3 Carleton University



Finding a largest convex point set

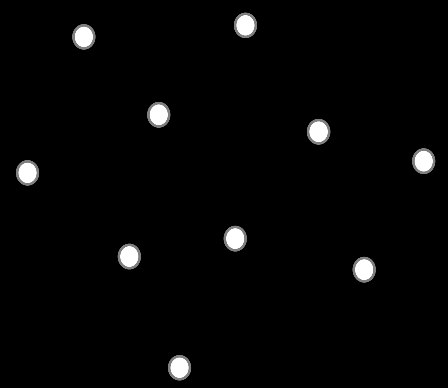
- Chvátal and Klincsek (1980) gave an $O(n^3)$-time and $O(n^2)$-space algorithm
- Edelsbrunner and Guibas (1989) improved the space complexity to O(n)



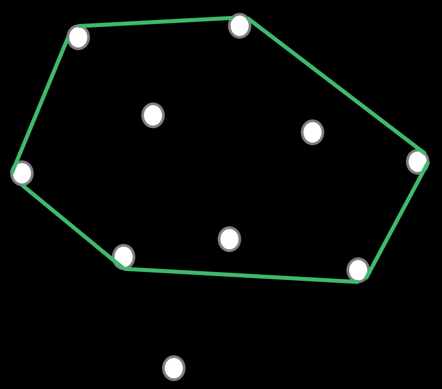
Finding *k* large convex point sets (maximize the minimum)

Given a set *P* of *n* points in the plane and a positive integer *k*, select *k* pairwise disjoint convex subsets of *P* such that the cardinality of the smallest subset is maximized.

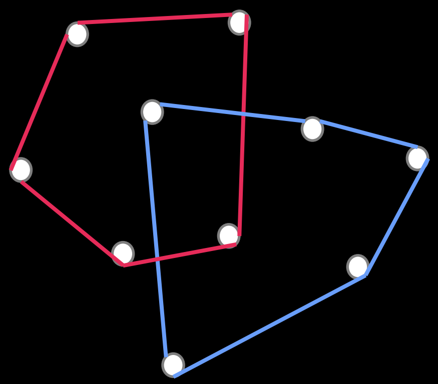
Input



Output when k = 1

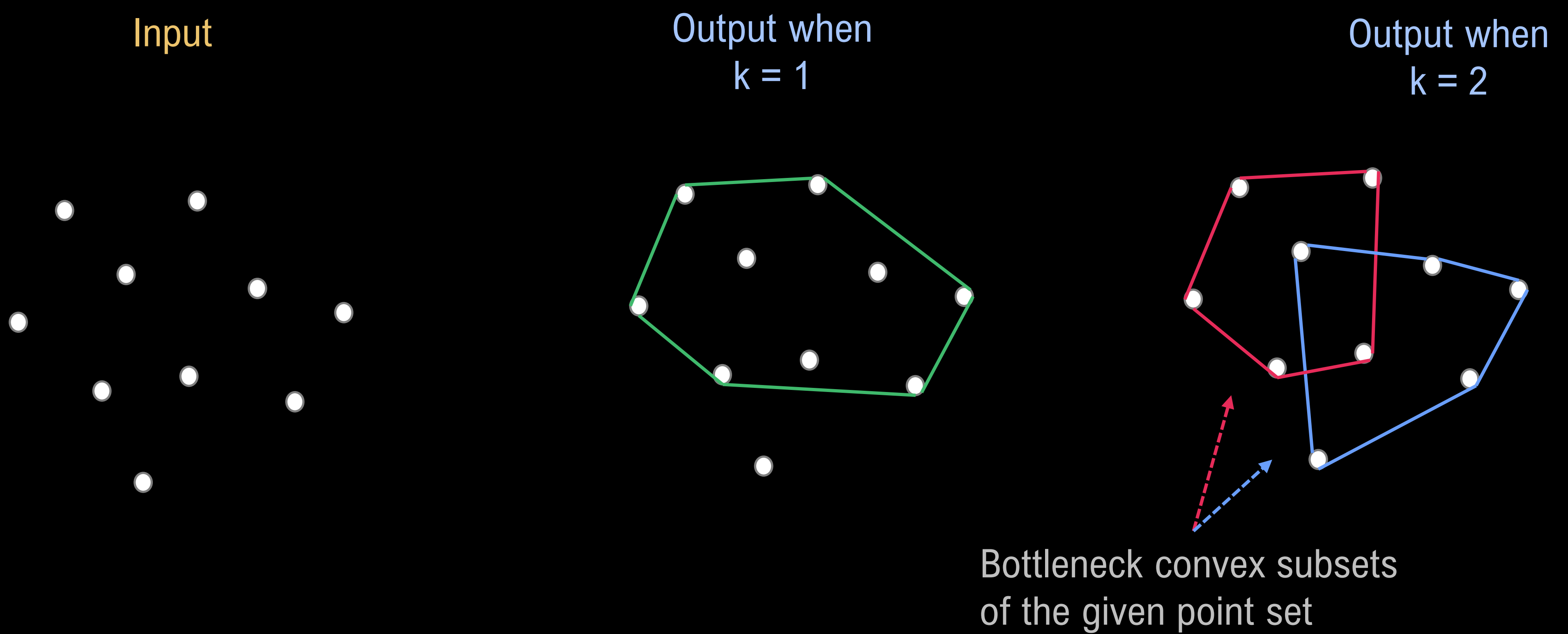


Output when k = 2



Bottleneck Convex Subsets

Given a set *P* of *n* points in the plane and a positive integer *k*, select *k* pairwise disjoint convex subsets of *P* such that the cardinality of the smallest subset is maximized.

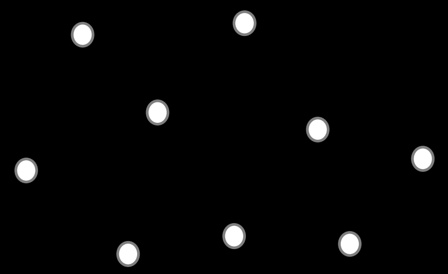


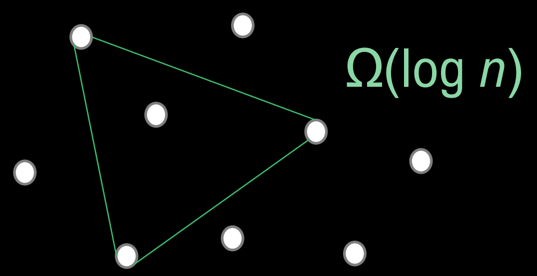
Our Contributions

- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
- Bottleneck Convex Subsets is NP-hard when *k* is an arbitrary input parameter.
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set

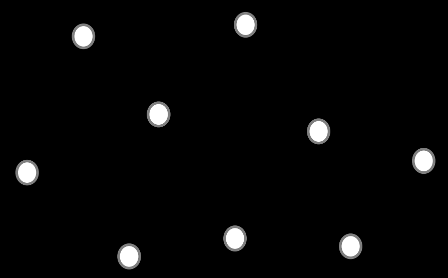
Related Research – lower bound when k∈ O(n/ log n)

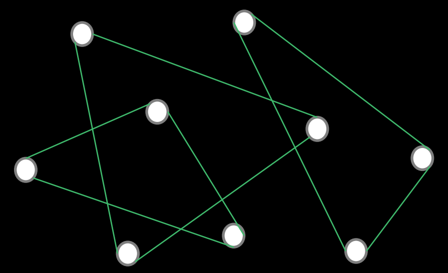
- Erdős-Szekeres theorem (1935): Every point set contains a convex a Ω(log *n*)-gon





- Urabe (1996): One can partition a point set into O(n/ log n) convex subsets, each of size O(log n)

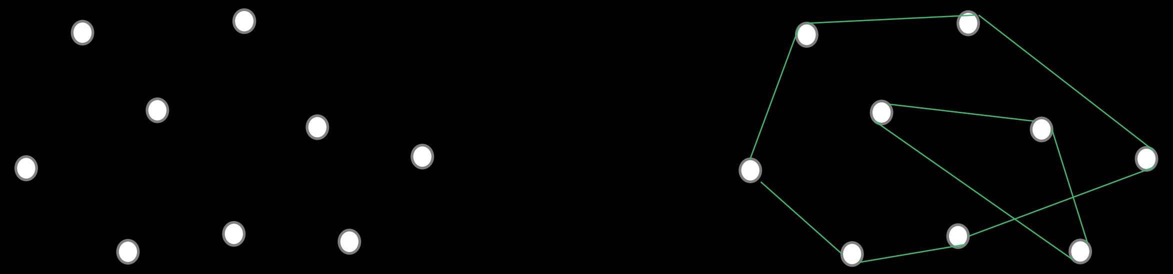




- Erdős et al. (1996): For sufficiently large n and with k ≤ (n − Ramsey-remainder(k))/k one can find *k* point sets where the size of the smallest convex set is at least *k* → Ramsey Reminder Problem

Related Research – convex cover number

- Find the minimum number of disjoint convex sets that covers the given point set

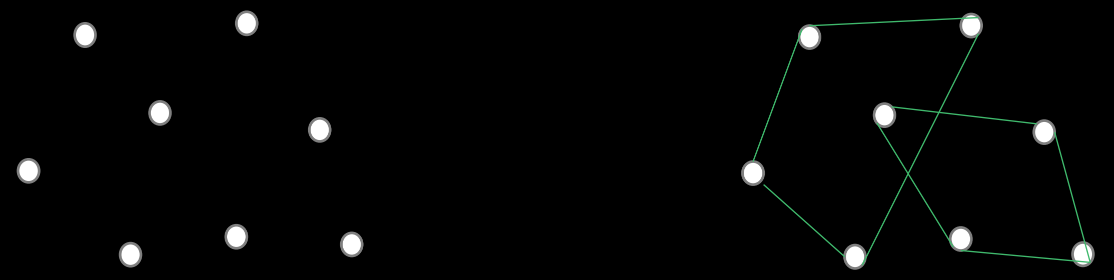


- Arkin et al. (2003): Finding the convex cover number of a point set is NP-hard
- Arkin et al. (2003): A polynomial-time O(log n)-approximation for convex cover number

Related Research – Finding *k* large convex sets for large *k*

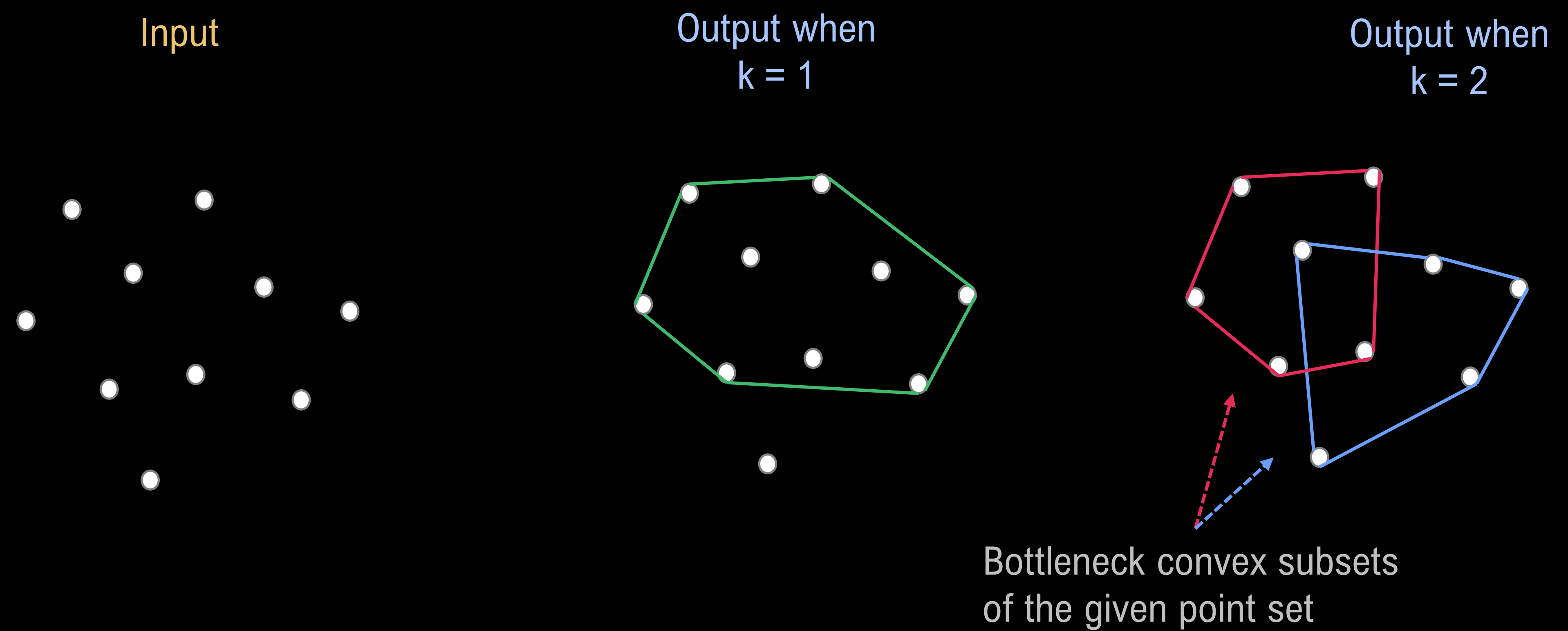
- If we want at least n/3 convex sets (i.e., $k \geq n/3$), then depending on the point set cardinality we can greedily partition the point set into balanced sets to maximize the minimum subset

- Károlyi (2003): An $O(n \log n)$-time algorithm algorithm to decide whether a partition into convex quadrilaterals exists. Therefore, we can leverage to find polynomial-time algorithm for the bottleneck convex subset problem when $k \geq n/4$



Bottleneck Convex Subsets

Given a set *P* of *n* points in the plane and a positive integer *k*, select *k* pairwise disjoint convex subsets of *P* such that the cardinality of the smallest subset is maximized.

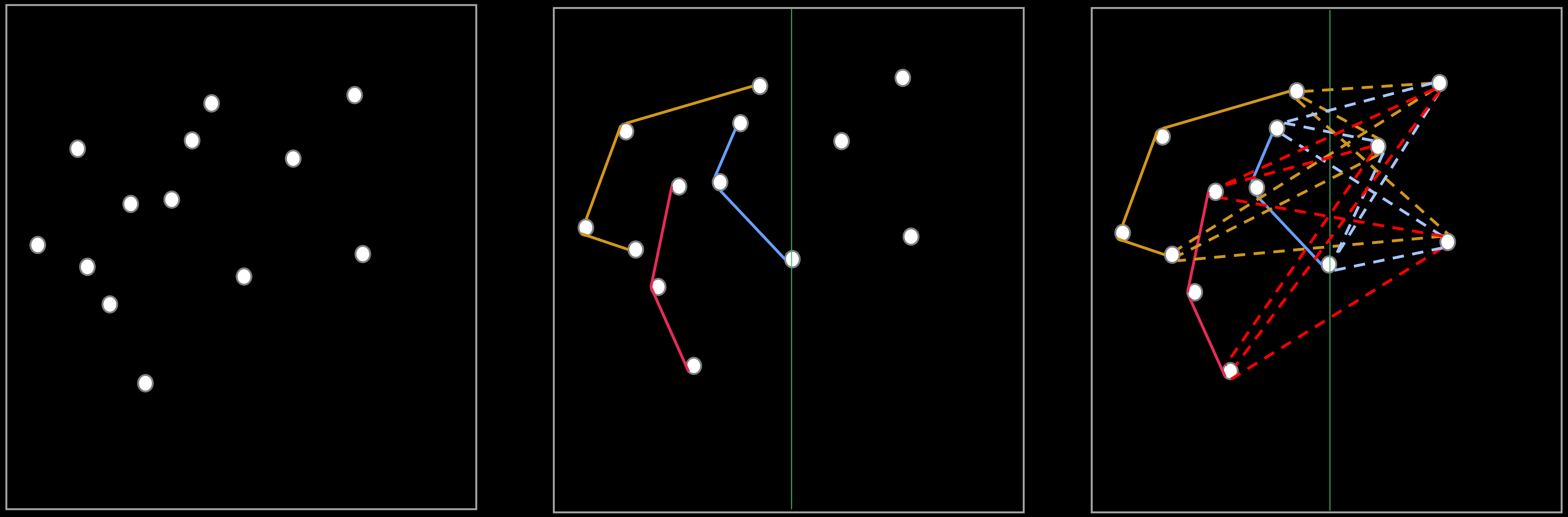


Our Contributions

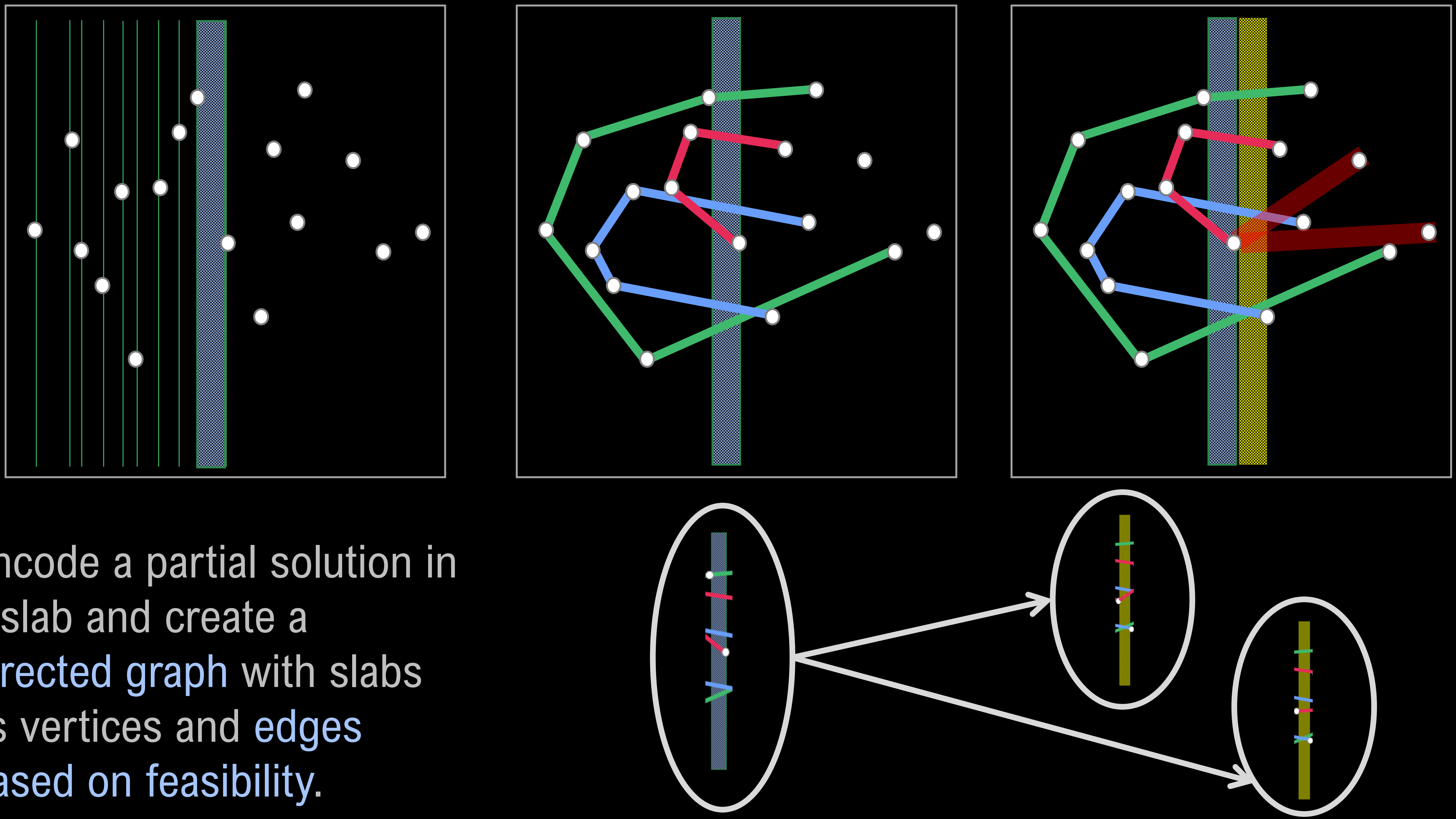
- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
- Bottleneck Convex Subsets is NP-hard when *k* is an arbitrary input parameter.
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set

A polynomial-time algorithm when k is fixed

Observation: Given any partial solution with k convex set, we have only $n^{O(k)}$ feasible options for extending these convex sets.

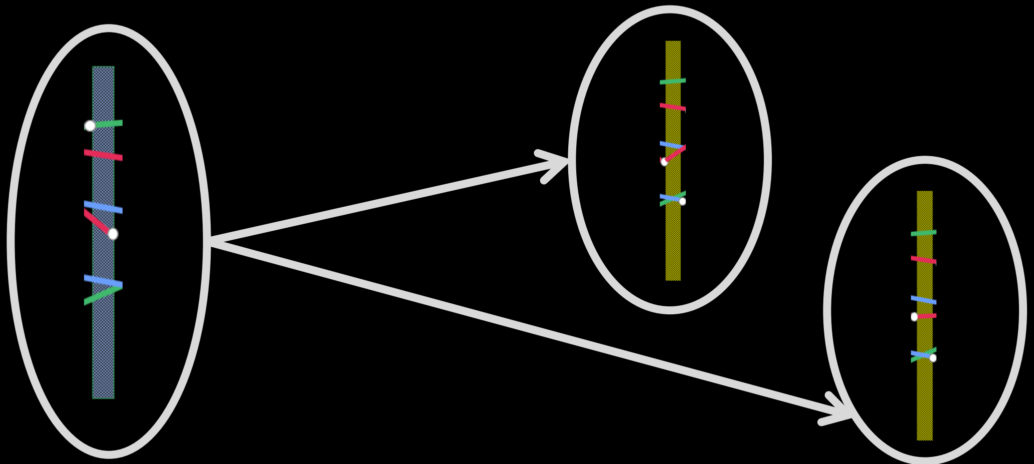


A polynomial-time algorithm when k is fixed



Encode a partial solution in a slab and create a directed graph with slabs as vertices and edges based on feasibility.

A polynomial-time algorithm when *k* is fixed



- The size of the directed graph is $O(n^{5k+3})$
- A solution is found using breadth-first search in this graph.

Given a set P of *n points* in the plane, and a *positive integer k*, Bottleneck Convex Subsets can be solved in $O(n^{5k+3})$ time

Our Contributions

- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
- Bottleneck Convex Subsets is NP-hard when *k* is an arbitrary input parameter.
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set

NP-hard when *k* is an arbitrary input parameter

- Reduce Numerical 3-Dimesional Matching to Angle Partition
- Reduce Angle Partition to Bottleneck Convex Subsets

Numerical 3-Dimesional Matching to Angle Partition

- Numerical 3-Dimesional Matching.
  Three sets $A=\{a_1,\ldots,a_n\}$, $B=\{b_1,\ldots,b_n\}$, $C=\{c_1,\ldots,c_n\}$, each with *n* distinct positive integers. Decide whether there exist *n* triples (a_i, b_j, c_k), such that $a_i + b_j = c_k$.

  Input: $A = \{16,14,10\}$, $B=\{8,6,12\}$, $C=\{18,28,20\}$

  Output: $T = \{(16,12,28), (14,6,20), (10,8,18)\}$

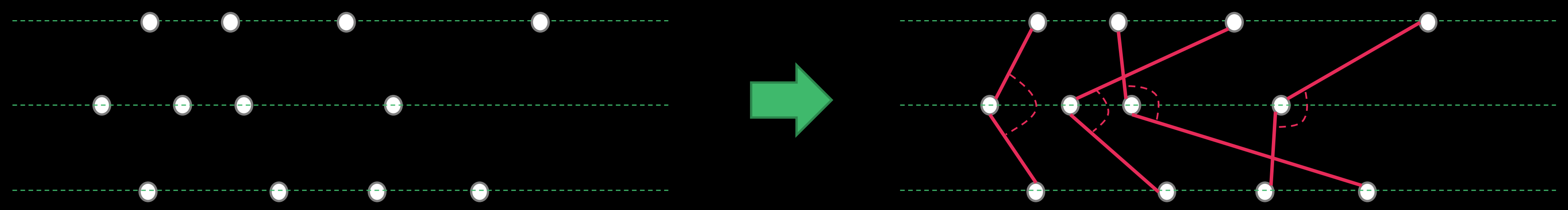
Numerical 3-Dimesional Matching to Angle Partition

- Numerical 3-Dimesional Matching.
  Three sets $A=\{a_1,\ldots,a_n\}$, $B=\{b_1,\ldots,b_n\}$, $C=\{c_1,\ldots,c_n\}$, each with *n* distinct positive integers. Decide whether there exist *n* triples (a_i, b_j, c_k), such that $a_i + b_j = c_k$.

  Input: $A = \{16,14,10\}$, $B=\{8,6,12\}$, $C=\{18,28,20\}$

  Output: $T = \{(16,12,28), (14,6,20), (10,8,18)\}$

- Angle Partition.
  Given a point set *P* of 3n points lying on three horizontal lines. Partition P into at most n y-monotone angles, where none of them are right facing



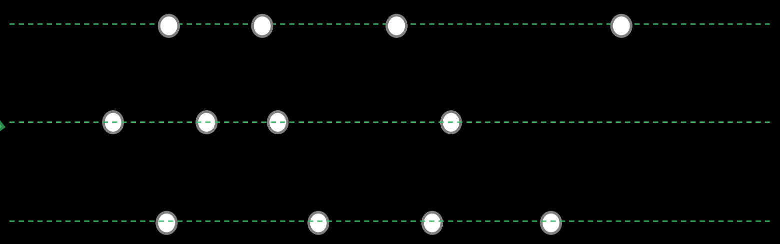
Numerical 3-Dimesional Matching to Angle Partition

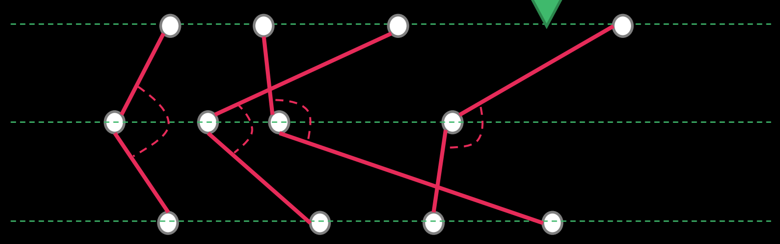
- **Numerical 3-Dimesional Matching.**
  Three sets $A=\{a_1,\ldots,a_n\}$, $B=\{b_1,\ldots,b_n\}$, $C=\{c_1,\ldots,c_n\}$, each with *n* distinct positive integers. Decide whether there exist *n* triples (a_i, b_j, c_k), such that $a_i + b_j = c_k$.

  Input: $A = \{16,14,10\}$, $B=\{8,6,12\}$, $C=\{18,28,20\}$

  Output: $T = \{(16,12,28), (14,6,20), (10,8,18)\}$

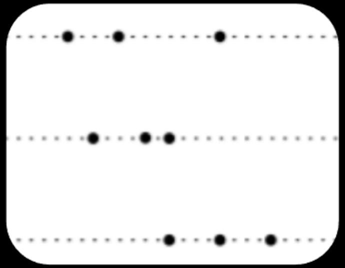
- **Angle Partition.**
  Given a point set *P* of 3n points lying on three horizontal lines. Partition P into at most n y-monotone angles, where none of them are right facing



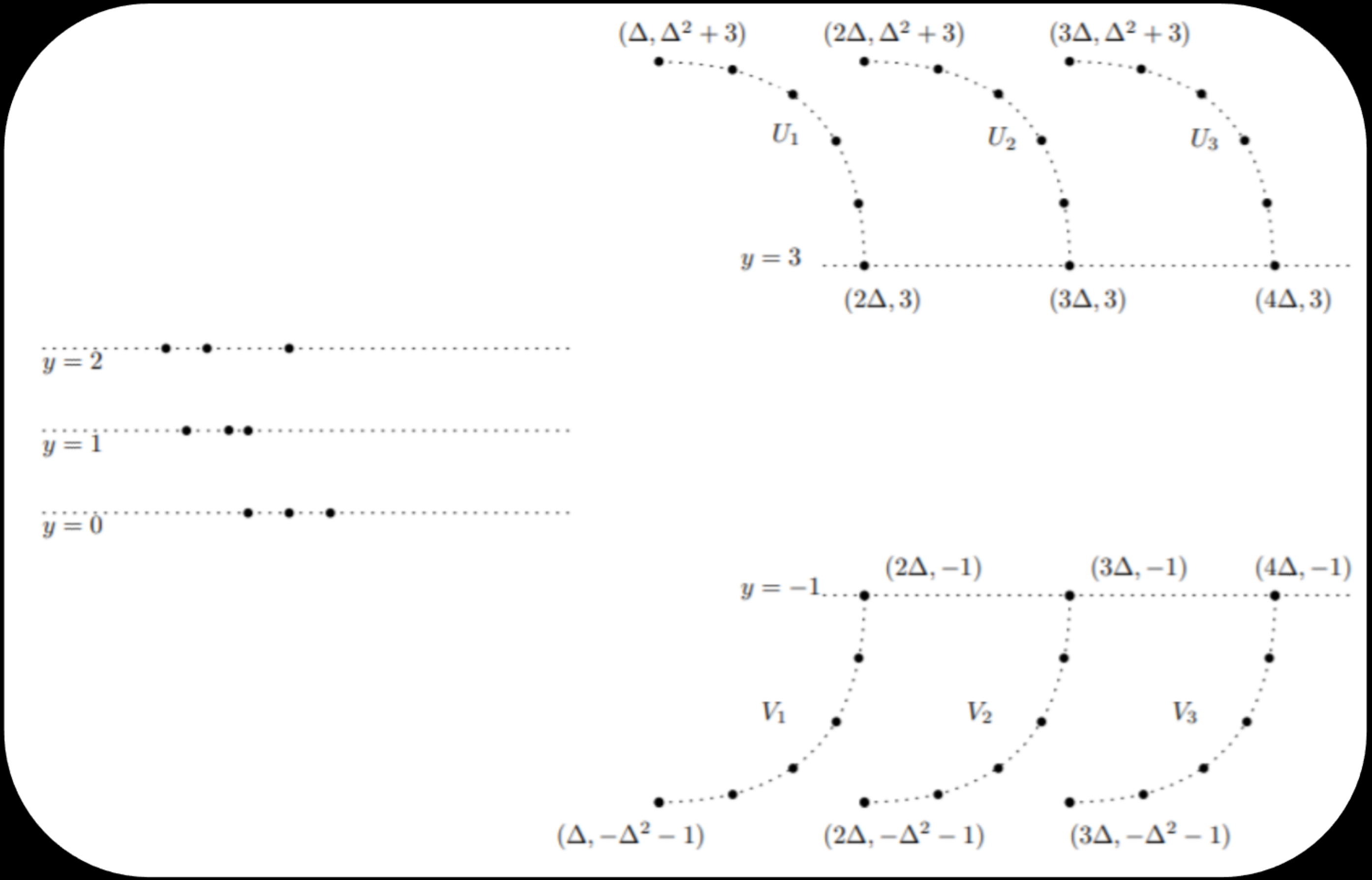


Angle Partition to Bottleneck Convex Subsets

Instance of Angle Partition
with $3n$ points on three lines

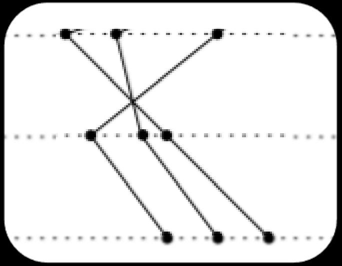


Instance of Bottleneck Convex Subsets
with $n(4n + 7)$ points, and $k = n$

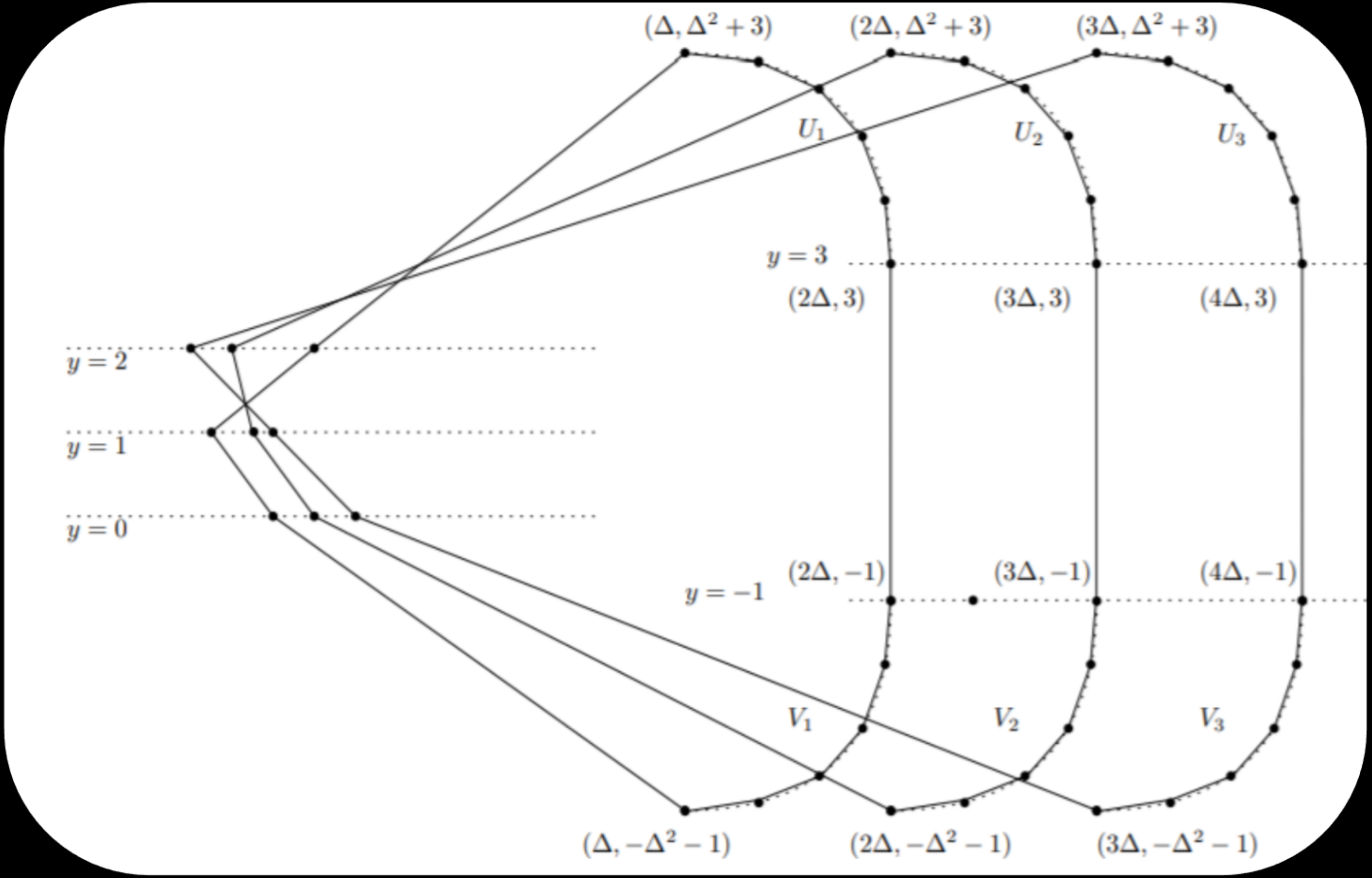


Angle Partition to Bottleneck Convex Subsets

Instance of Angle Partition
with $3n$ points on three lines

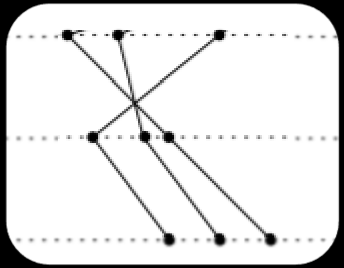


Instance of Bottleneck Convex Subsets
with $n(4n + 7)$ points, and $k = n$

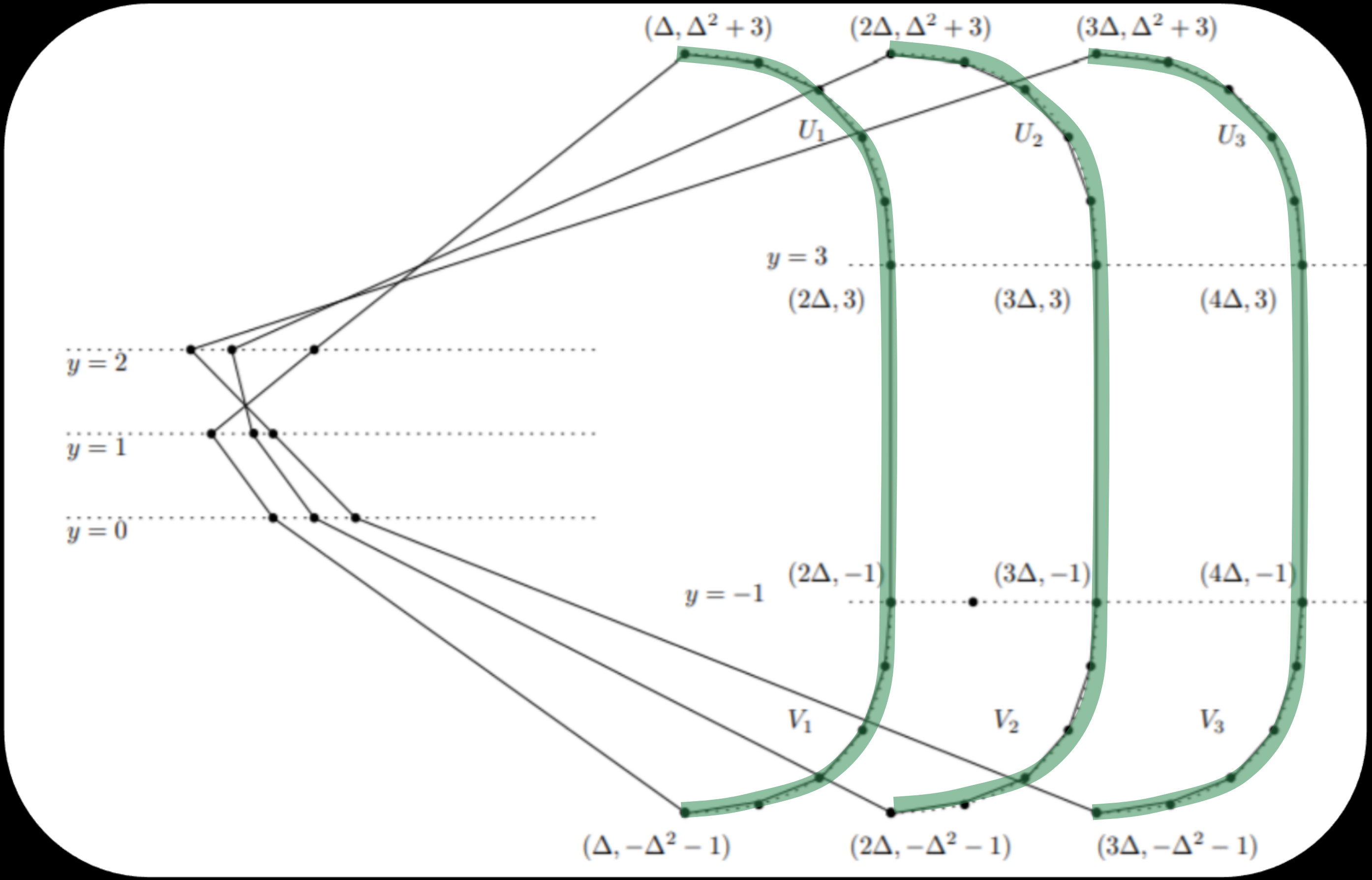


Angle Partition to Bottleneck Convex Subsets

Instance of Angle Partition
with $3n$ points on three lines



Instance of Bottleneck Convex Subsets
with $n(4n + 7)$ points, and $k = n$

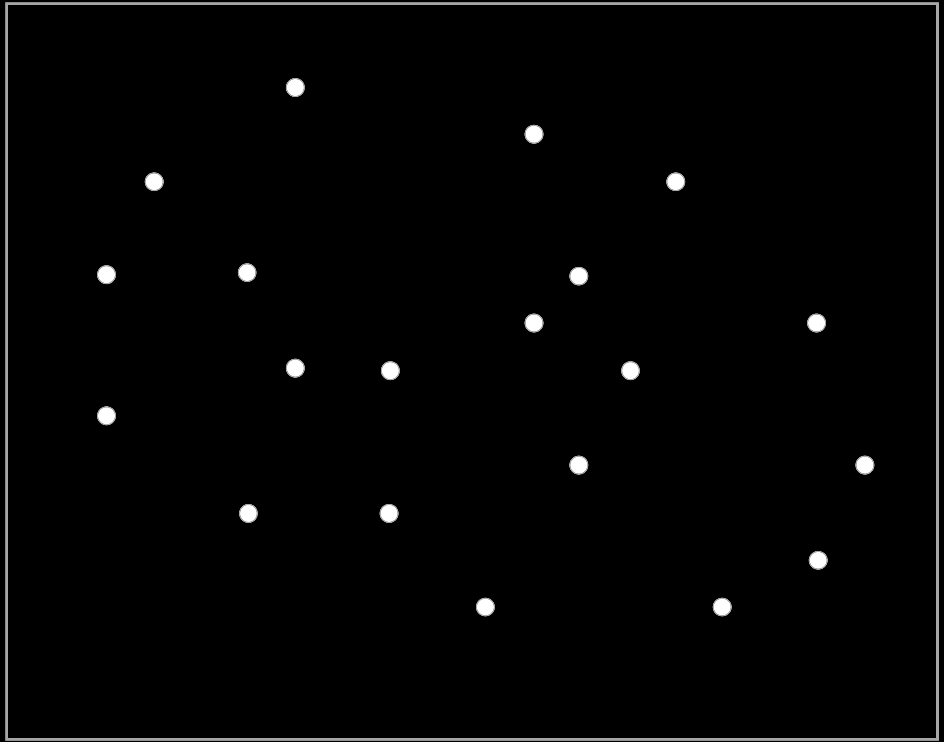


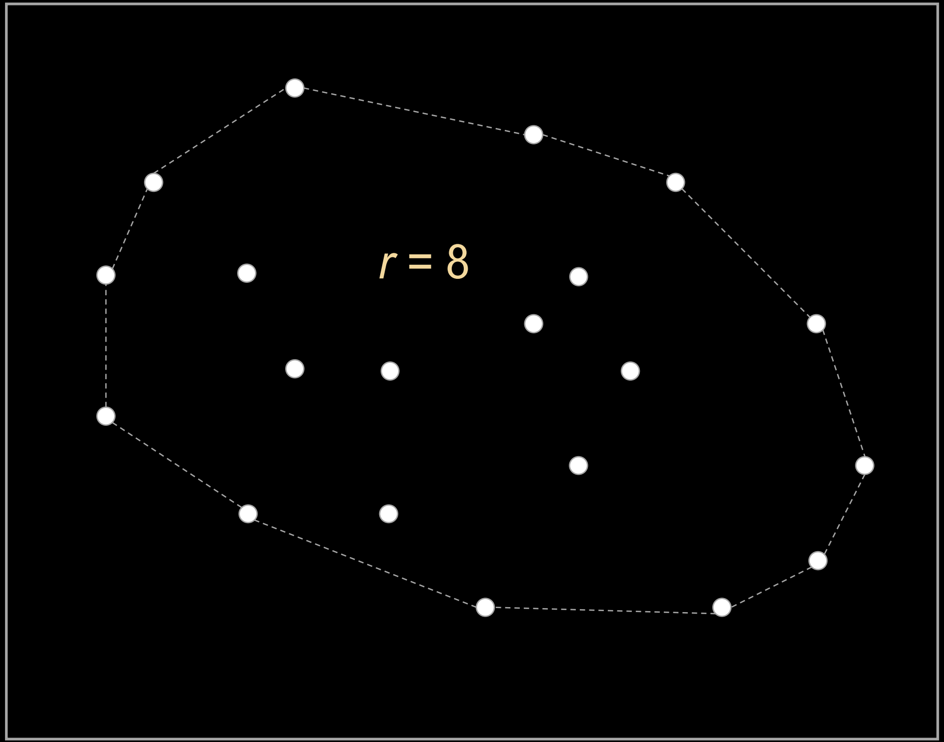
Our Contributions

- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
- Bottleneck Convex Subsets is NP-hard when *k* is an arbitrary input parameter.
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set

Fixed-parameter tractable algorithm

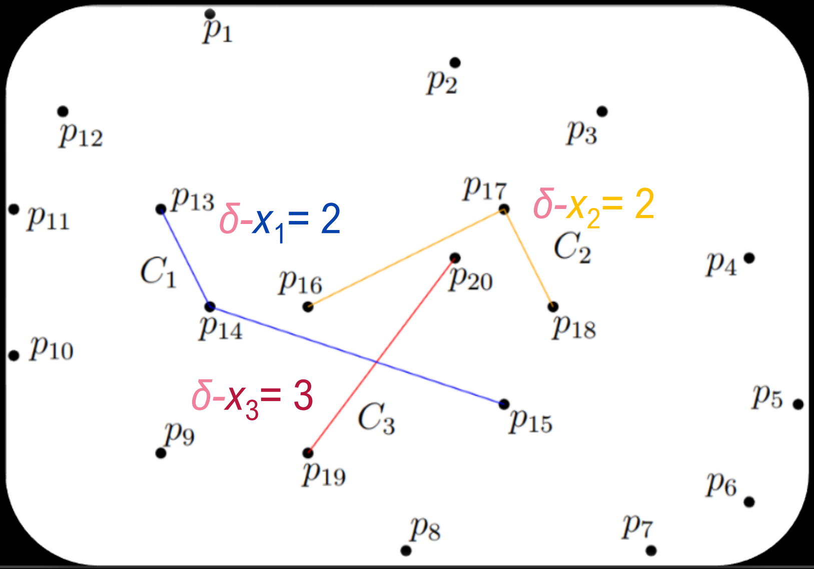
- A $f(r) \cdot n^{O(1)}$ time algorithm for bottleneck convex subsets, where r is the number of points interior to the convex hull





Fixed-parameter tractable algorithm

- A $f(r) \cdot n^{O(1)}$ time algorithm for bottleneck convex subsets, where r is the number of points interior to the convex hull
  - Guess the structure of at most k convex sets inside the convex hull
  - Guess the size δ of the minimum convex set in the solution
  - Each partial convex set of size x now requires (δ-x) *more points*
  - These (δ-x) points must come from the convex hull

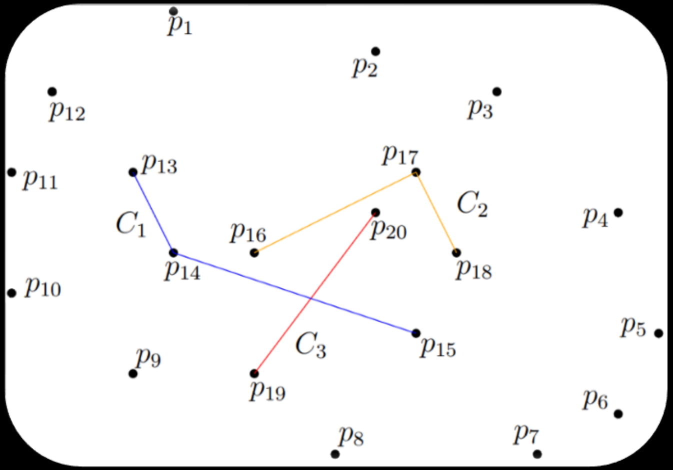


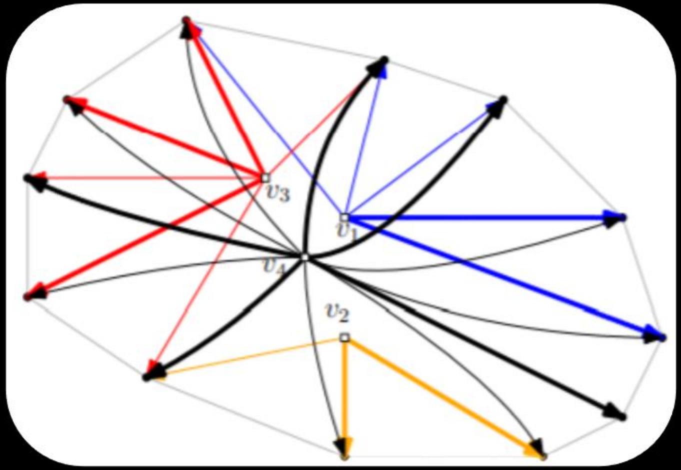
k = 4

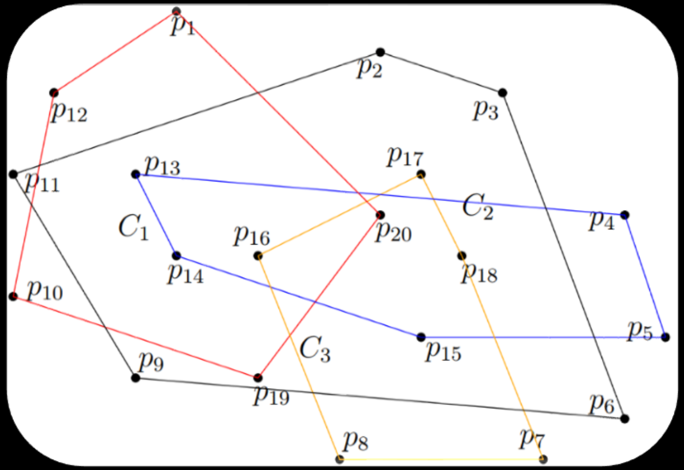
Guess for δ is *5*

Fixed-parameter tractable algorithm

- A $f(r) \cdot n^{O(1)}$ time algorithm for bottleneck convex subsets, where r is the number of points interior to the convex hull
  - Model the problem as a maximum flow problem
  - Create a graph with each partial set having a production of (δ-x) units of flow and each convex hull point as a sink that can consume 1 unit of flow







The overall time complexity becomes $O(r\,2^{r^{k+2}}\,g(n)\log n)$, where $g(n)$ is for computing maximum flow and term $\log n$ is for the guess and verify

Our Contributions

- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
- Bottleneck Convex Subsets is NP-hard when *k* is an arbitrary input parameter.
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set

Open Problems

- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
  - Does there exist a fixed-parameter tractable algorithm parameterized by k?
- Bottleneck Convex Subsets is NP-hard when k is an arbitrary input parameter.
  - Does there exist a polynomial-time algorithm for the case when $k \in \Theta(n)$?
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set
  - How about other point-set parameters such as number of convex layers and minimum line cover number?

Open Problems

- A polynomial-time algorithm that solves Bottleneck Convex Subsets for any fixed k
  - Does there exist a fixed-parameter tractable algorithm parameterized by k?
- Bottleneck Convex Subsets is NP-hard when k is an arbitrary input parameter.
  - Does there exist a polynomial-time algorithm for the case when $k \in \Theta(n)$?
- A fixed-parameter tractable algorithm parameterized by the number of points that are strictly interior to the convex hull of the given point set
  - How about other point-set parameters such as number of convex layers and minimum line cover number?

Thank You!

University of Manitoba
University of Saskatchewan
Carleton University
NSERC CRSNG