Kernels for Preserving Connectivity

WorKer 2015

M. S. Ramanujan
University of Bergen, Norway

Joint work with

Manu Basavaraju, Fedor Fomin, Petr Golovach, Pranabendu Misra and Saket Saurabh

Connectivity

- A graph is λ-(edge) connected if there is no cut-set of size less than λ.

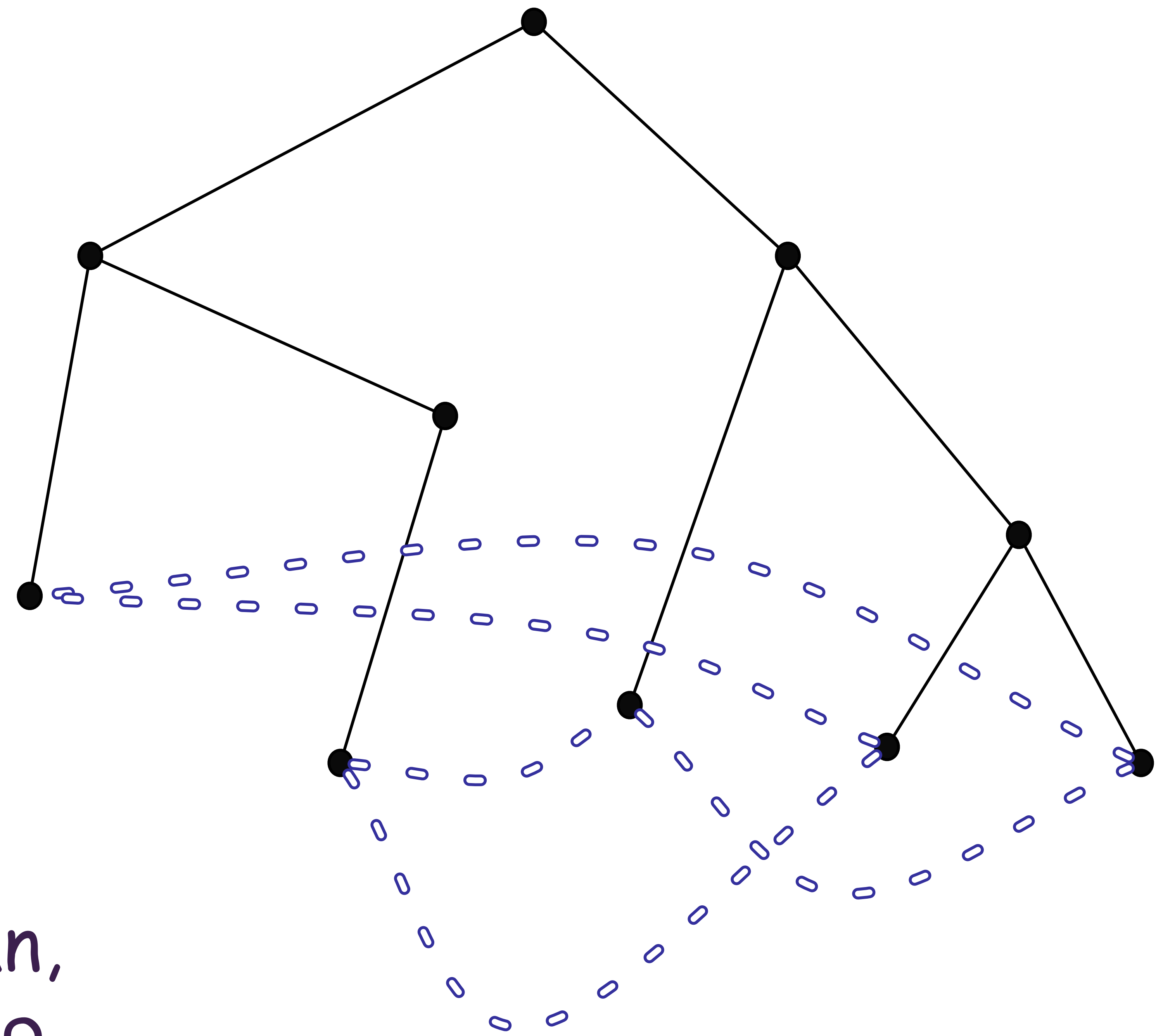
- Fundamental problem: how few non-edges can we add to a given graph to increase its connectivity to a required value?

Connectivity

- Several variants of this problem.
- When the set of non-edges we are allowed to add is unrestricted then polynomial time solvable [Eswaran and Tarjan + Plesnik 76, Watanabe and Nakamura 87].
- If the set of allowed non-edges is specified, then NP-hard in general.

Connectivity

NP-hard even when the links form a cycle on the leaves!



Ravi, Jordan, Cheriyan 99

Preserving Connectivity

I/p: Graph G, set of links L, integer k, G is λ edge connected, G+L is (λ+c) edge connected.

Question 1: Is there a set S of at most k links such that G+S is (λ+c) connected?

Question 2: Is there a set S of at least k links such that G+(L\S) is (λ+c) connected?

Preserving Connectivity

c=1!

I/p: Graph G, set of links L, integer k, G is λ edge connected, G+L is (λ+1) edge connected.

Question 1: Is there a set S of at most k links such that G+S is (λ+1) connected?

Question 2: Is there a set S of at least k links such that G+(L\S) is (λ+1) connected?

Preserving Connectivity

(Adding k links)

- Question 1 answered by Nagamochi (2003), and Marx and Vegh (2013) - $2^{O(k \log k)}$ algorithms.

- Guo and Uhlmann 2010 give a quadratic kernel when λ is odd, Marx and Vegh give a cubic kernel (both λ odd/even) Question 1 (also for weighted case).

- Case when λ is odd seems much easier than the case when λ is even!

This work

(Deleting k links)

- Answer Question 2 - a $2^{O(k)}$ poly(n) algorithm.

- a linear kernel for λ odd and quadratic kernel for λ even.

- Improve dependence on k for the best algorithms for Question 1 from $2^{O(k \log k)}$ to $2^{O(k)}$

This work

(Deleting k links)

- Answer Question 2 - a $2^{O(k)}$ poly(n) algorithm.

- a linear kernel for λ odd and quadratic kernel for λ even.

- Improve dependence on k for the best algorithms for Question 1 from $2^{O(k \log k)}$ to $2^{O(k)}$

Structure of min-cuts (Dinits, Karzanov, Lomonosov 76)

If G is λ(odd) connected

then

the λ-cuts in G can be represented as the edges of a tree

If G is λ(even) connected

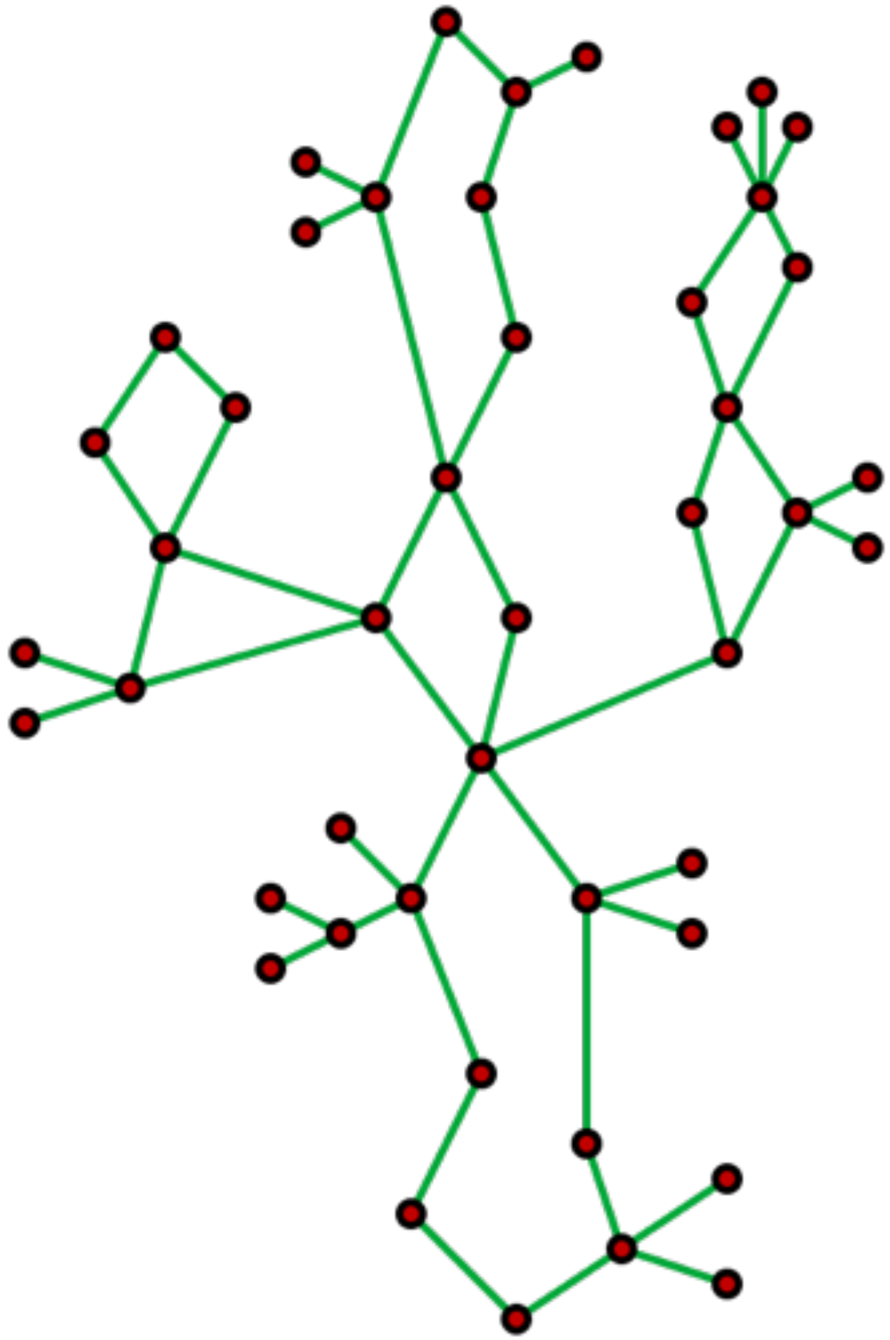
then

the λ-cuts in G can be represented as the edges of a cactus

Vertices of the tree/cactus = λ-inseparable sets

λ-inseparable sets = vertices that cannot be separated using λ-cuts

Cactus graphs



A connected graph where any 2 cycles intersect in at most 1 vertex.

Source: wikipedia

Preserving Connectivity

I/p: Graph G, set of links L, integer k, G is a tree/cactus, G+L is λ=2/3 edge connected.

Question 1: Is there a set S of k links such that G+S is (λ+1) connected?

Question 2: Is there a set S of k links such that G+ (L\S) is (λ+1) connected?

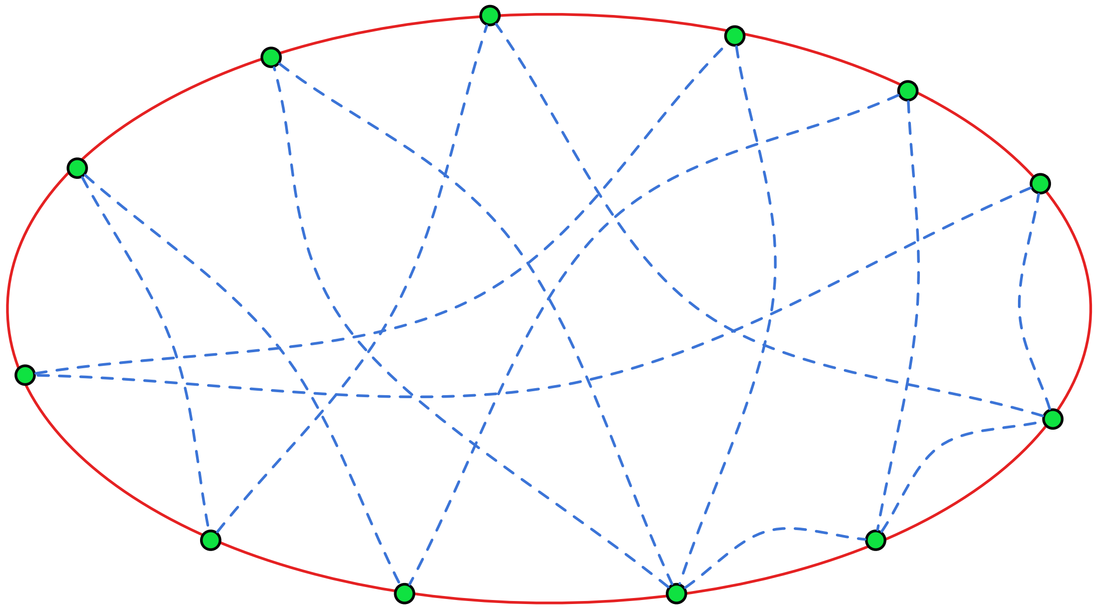
This Talk: Question 2 + Cycle Augmentation

I/p: Cycle C, set of chords L, integer k, C+L is 3 edge connected.

Question: Is there a set S of k chords such that C+(L\S) is 3 edge- connected?

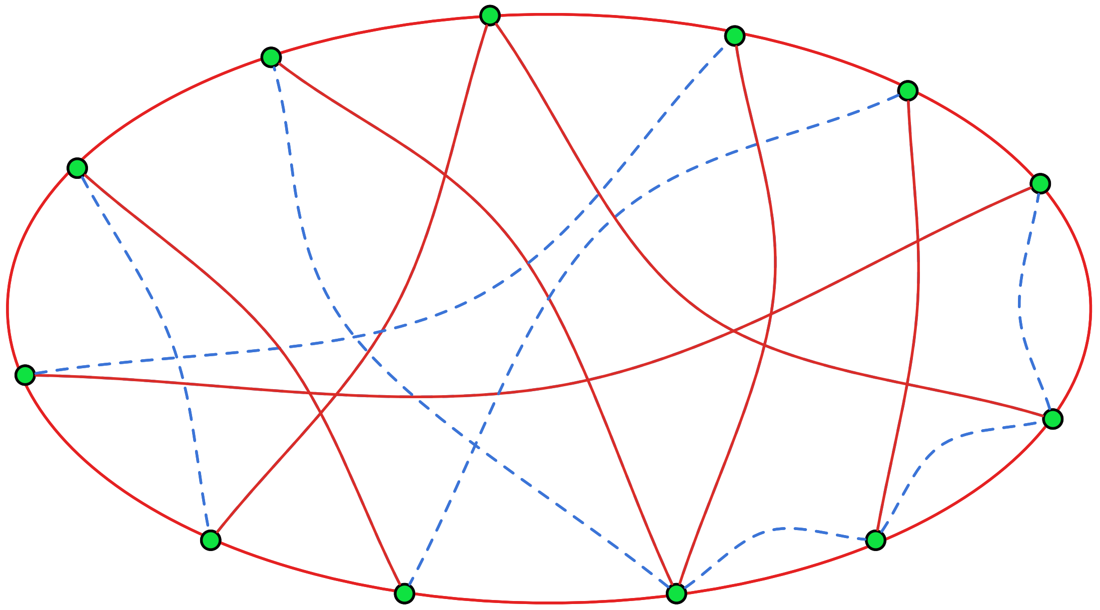
Also NP-hard.

Cycle Augmentation



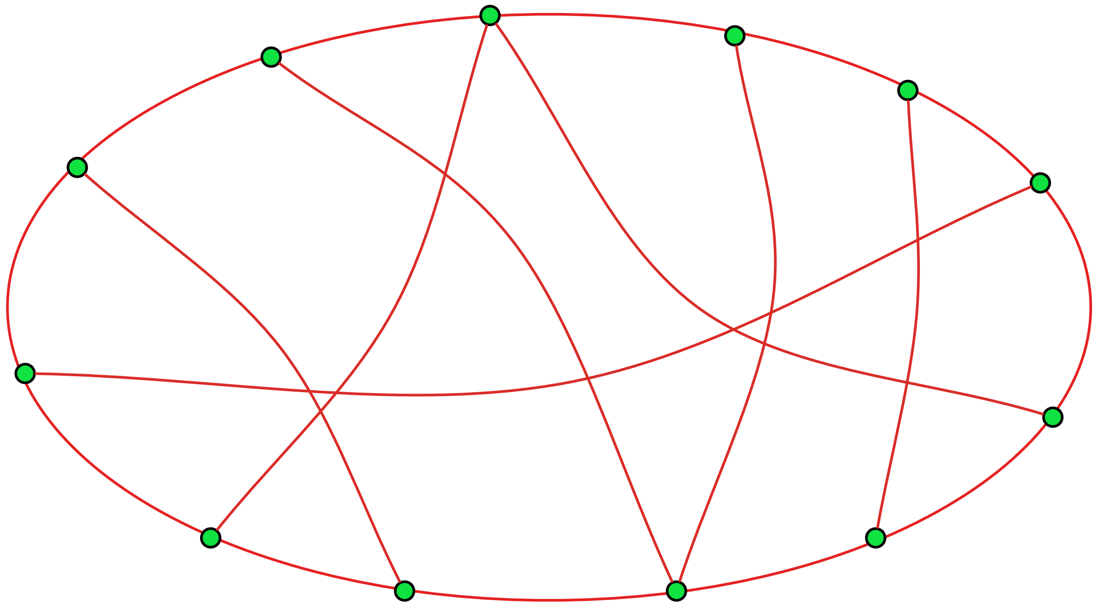
Given a cycle with chords, ..

Cycle Augmentation



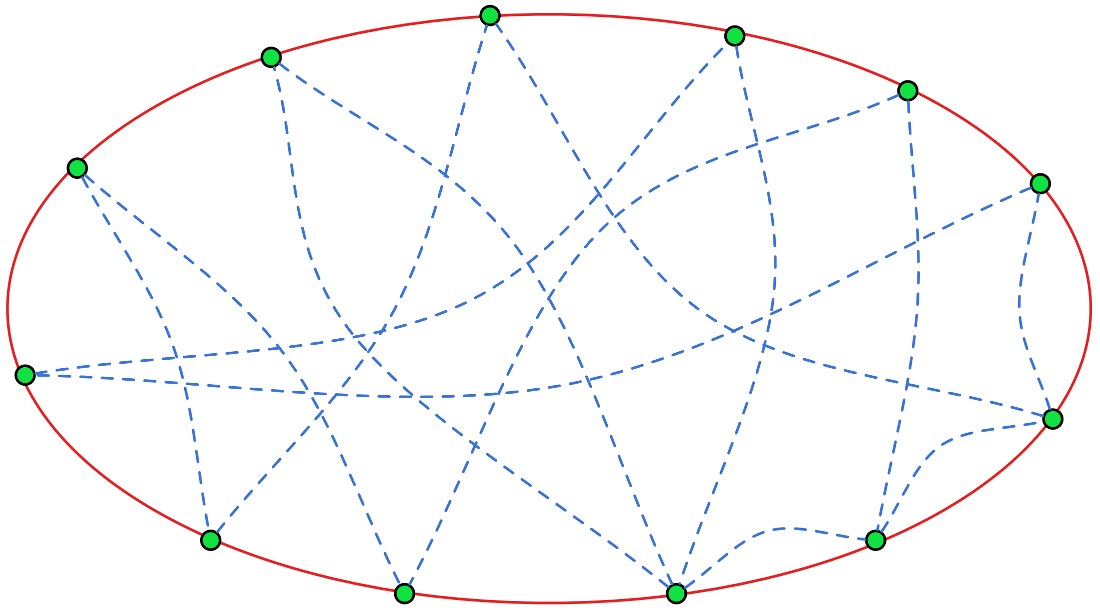
Find an `Augmenting Set'.

Cycle Augmentation



What do Augmenting Sets look like?

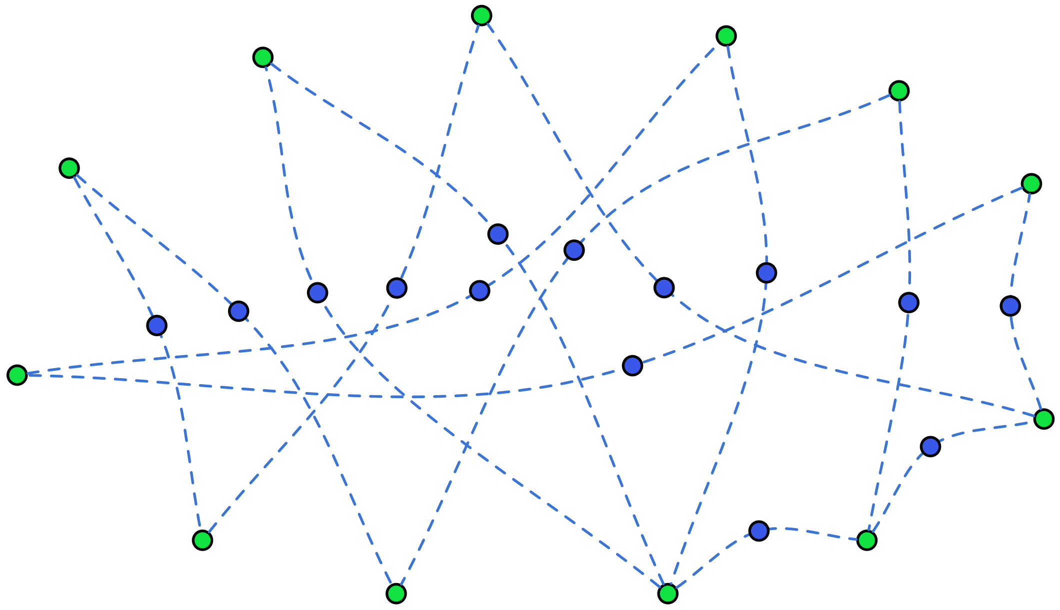
Cycle Augmentation



Define an auxiliary graph H

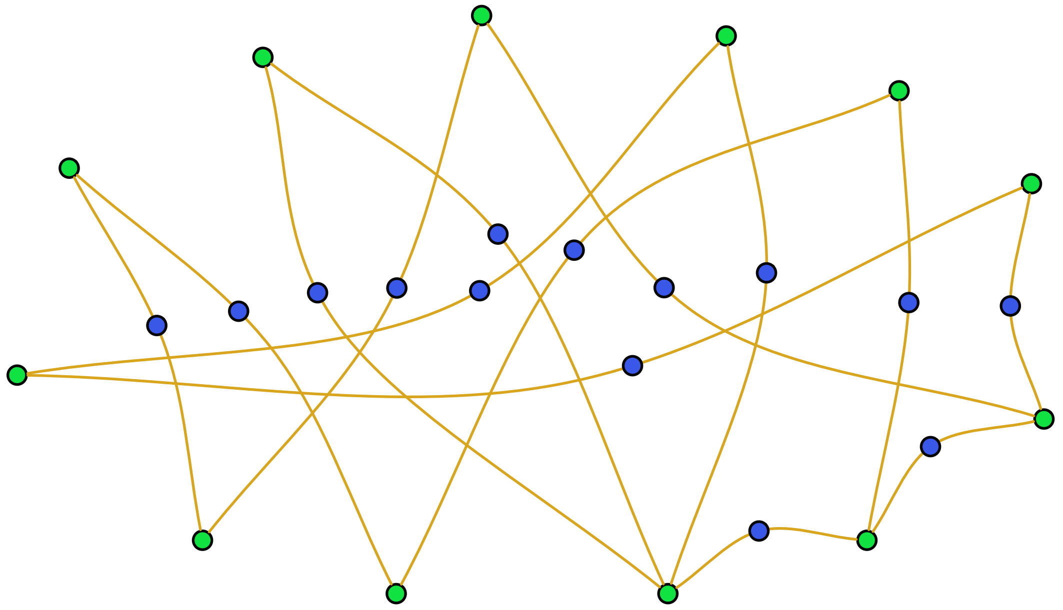
Start with H=G[L]

Cycle Augmentation



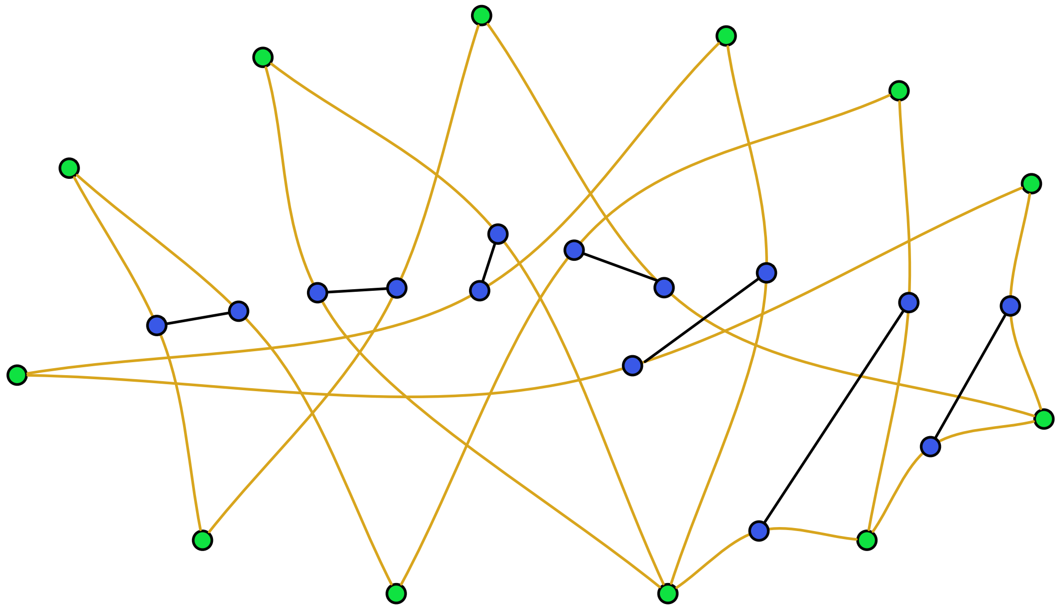
Subdivide every chord

Cycle Augmentation



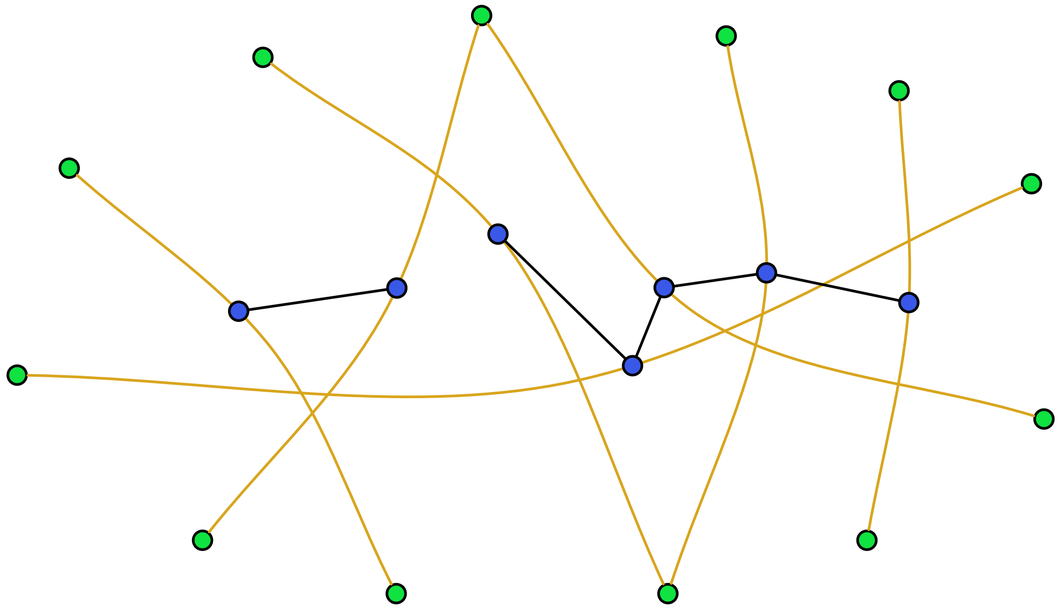
Subdivide every chord

Cycle Augmentation



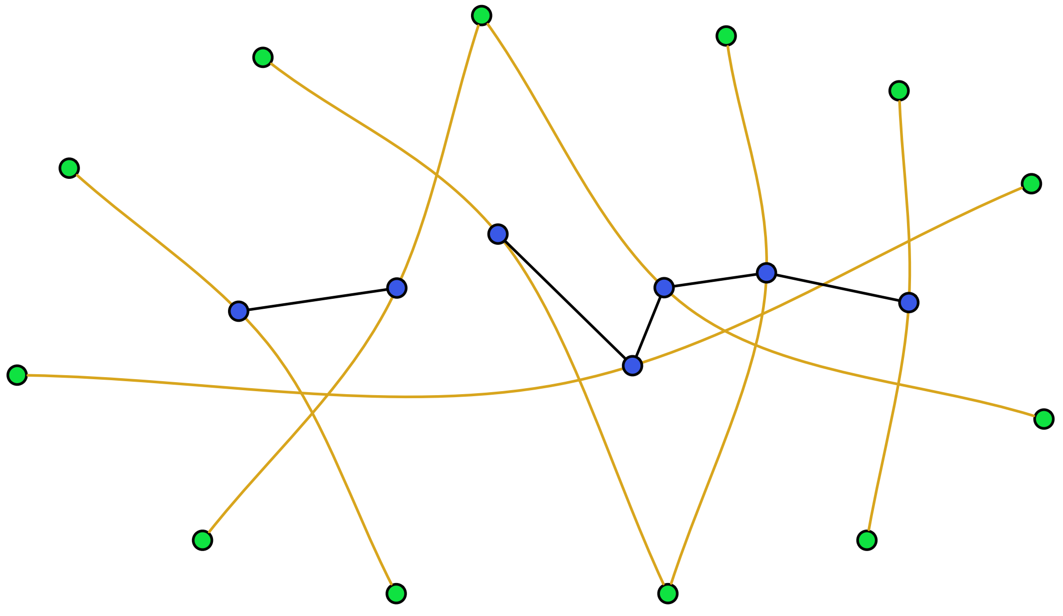
Add an edge between the subdivision vertices
if the corresponding chords 'cross'.

Cycle Augmentation



Observation: the subdivision vertices corresponding to the solution seem to 'connect' the original vertices.

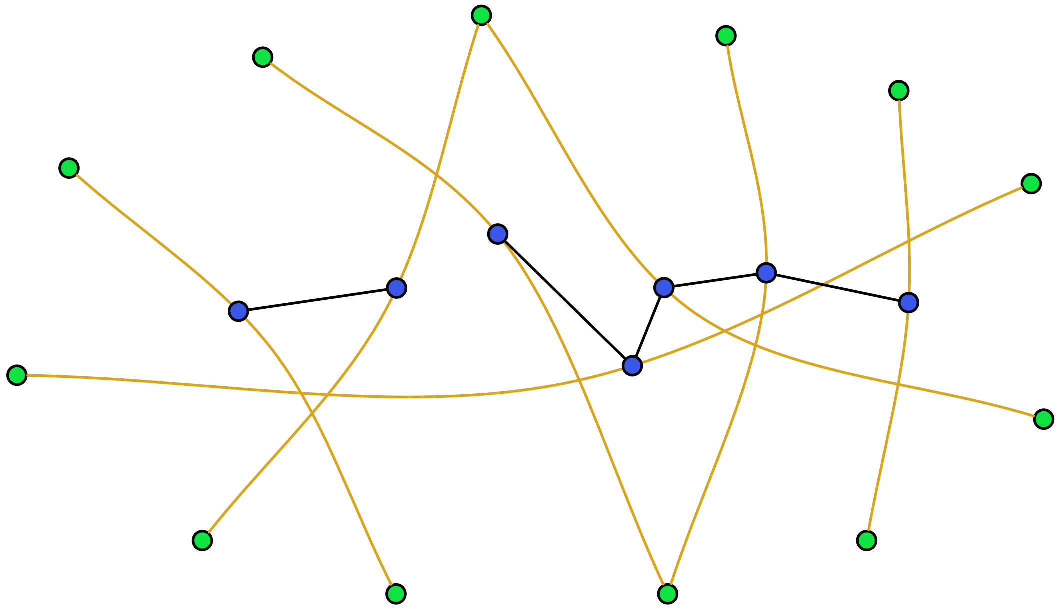
Cycle Augmentation



Lemma: X is an augmenting set for C if and only if H[X ∪ V] is connected.

Cycle Augmentation

FPT algorithm —> with some preprocessing, get treewidth of H down to O(k) and then find a steiner tree.

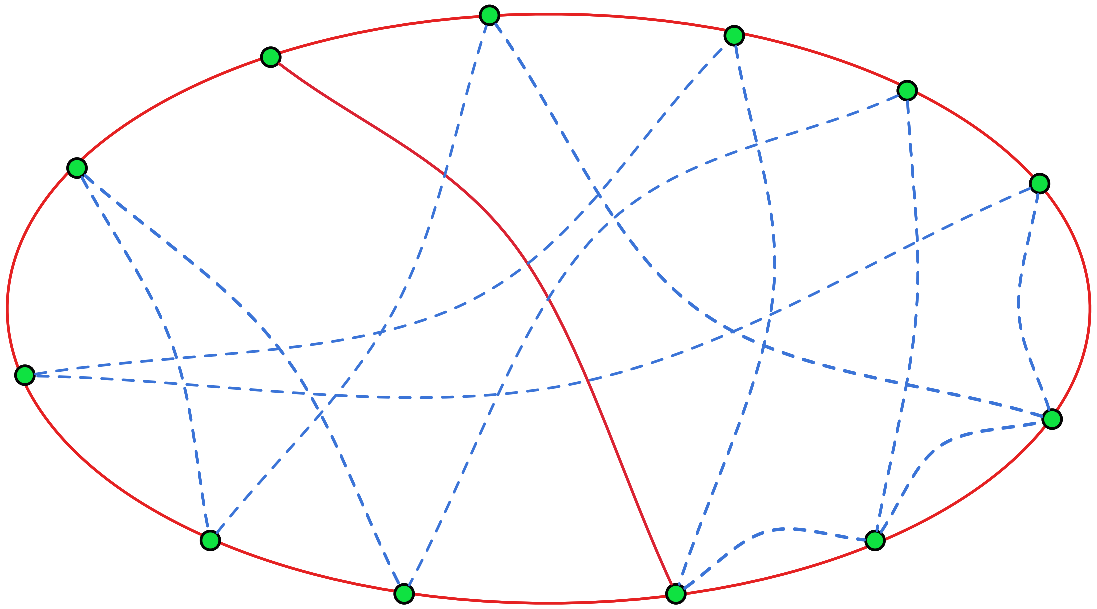


Lemma: X is an augmenting set for C if and only if H[X ∪ V] is connected.

Cycle Augmentation

Preprocessing

Cycle Augmentation

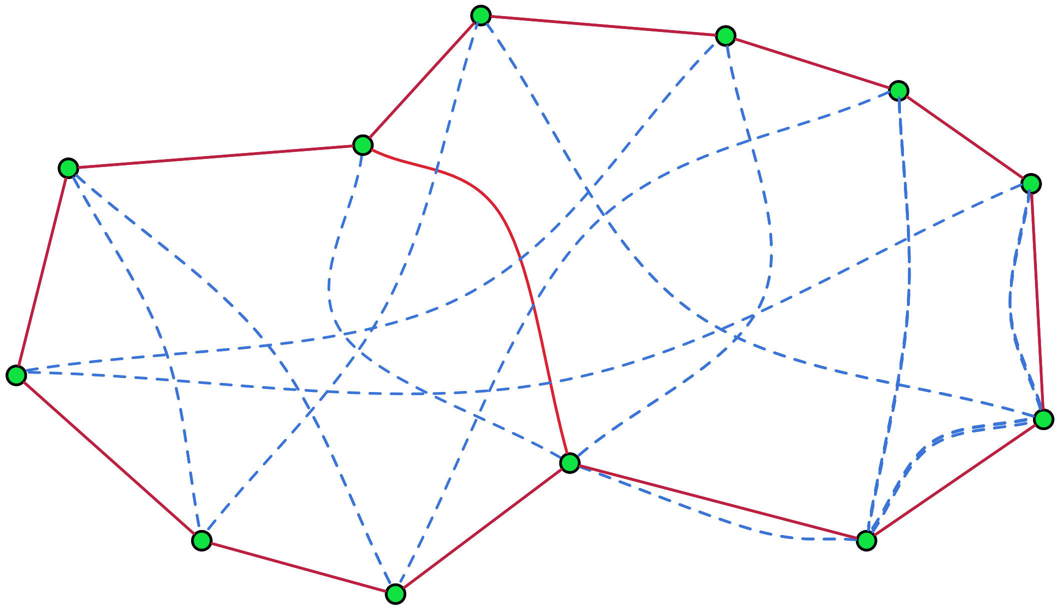


When a link is the only one crossing a 2-cut/only one incident on a vertex then it must be added.

So...we just add all such links

Cycle Augmentation

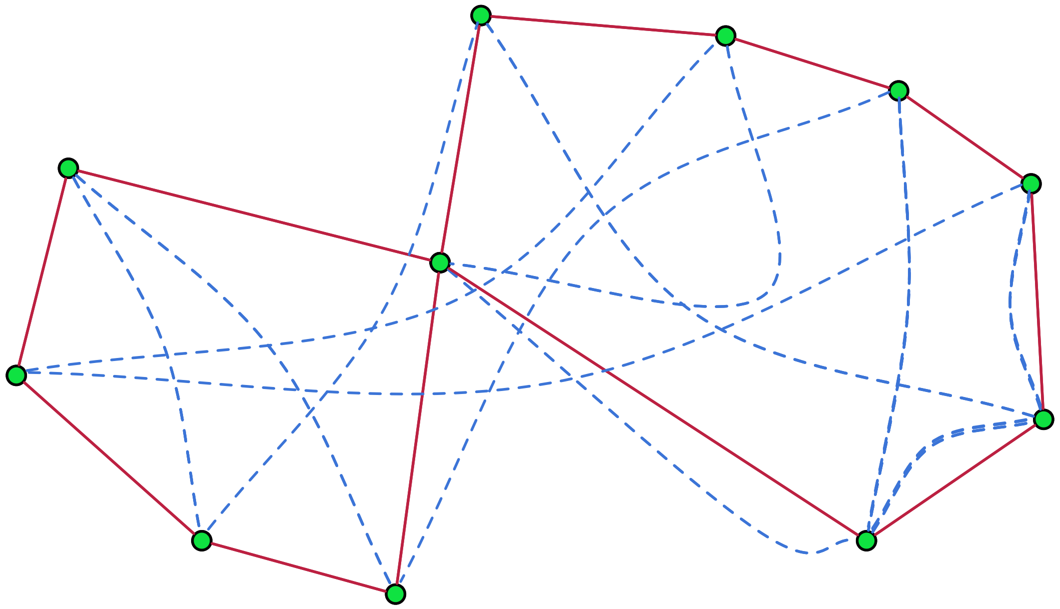
- If a link is necessary, add it to the graph.



Identify the endpoints of the links to get a cactus!

Cycle Augmentation

- If a link is necessary, add it to the graph.

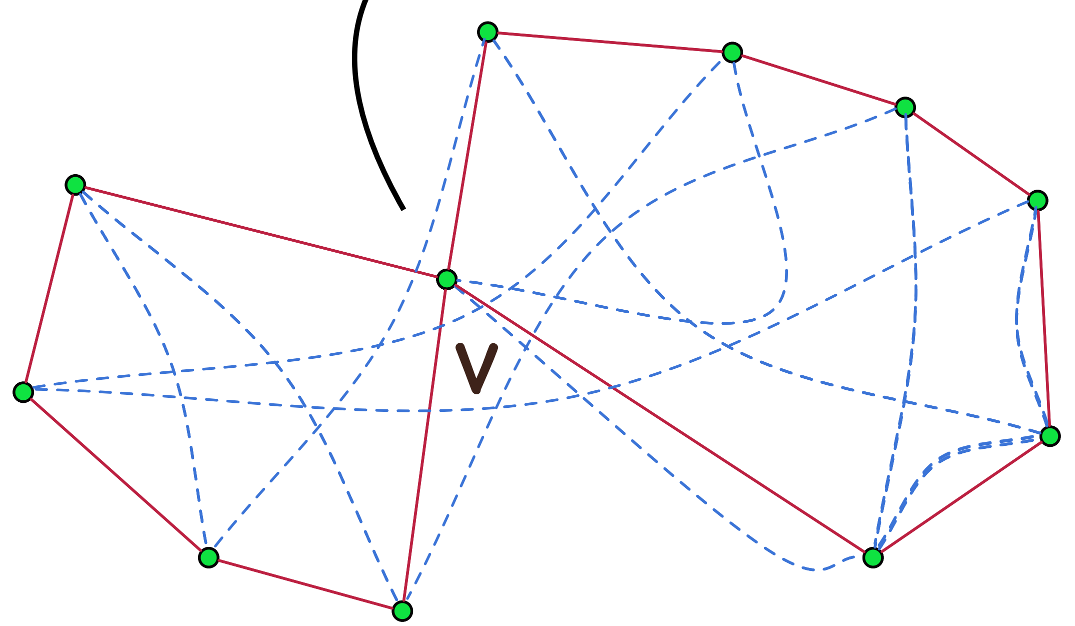


Identify the endpoints of the links to get a cactus!

Cycle Augmentation

- If a link is necessary, add it to the graph.

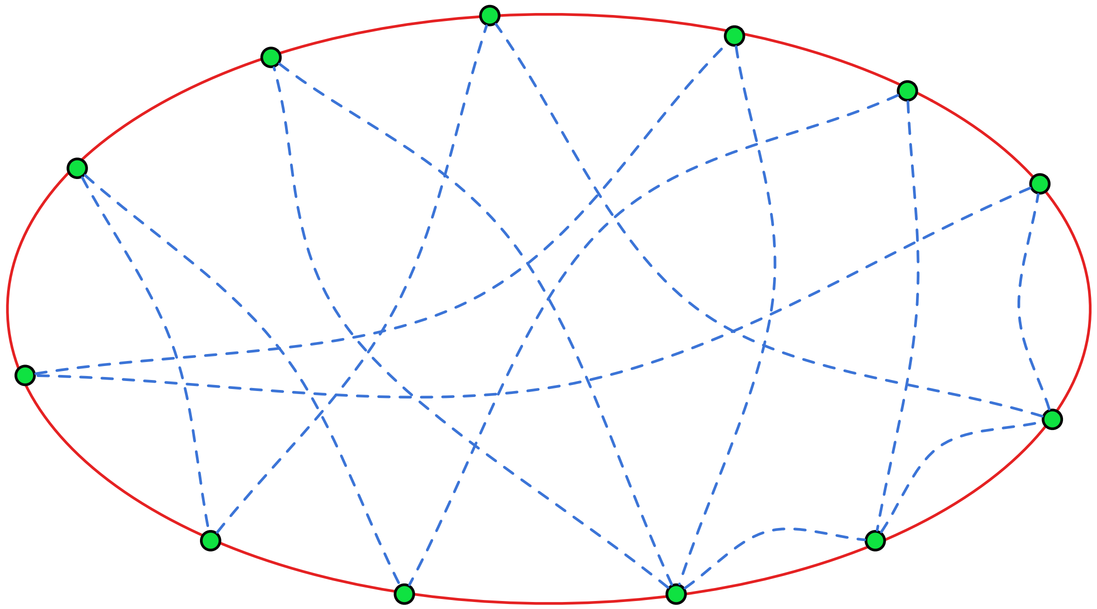
Projectively incident on v



Identify the endpoints of the links to get a cactus!

Cycle Augmentation

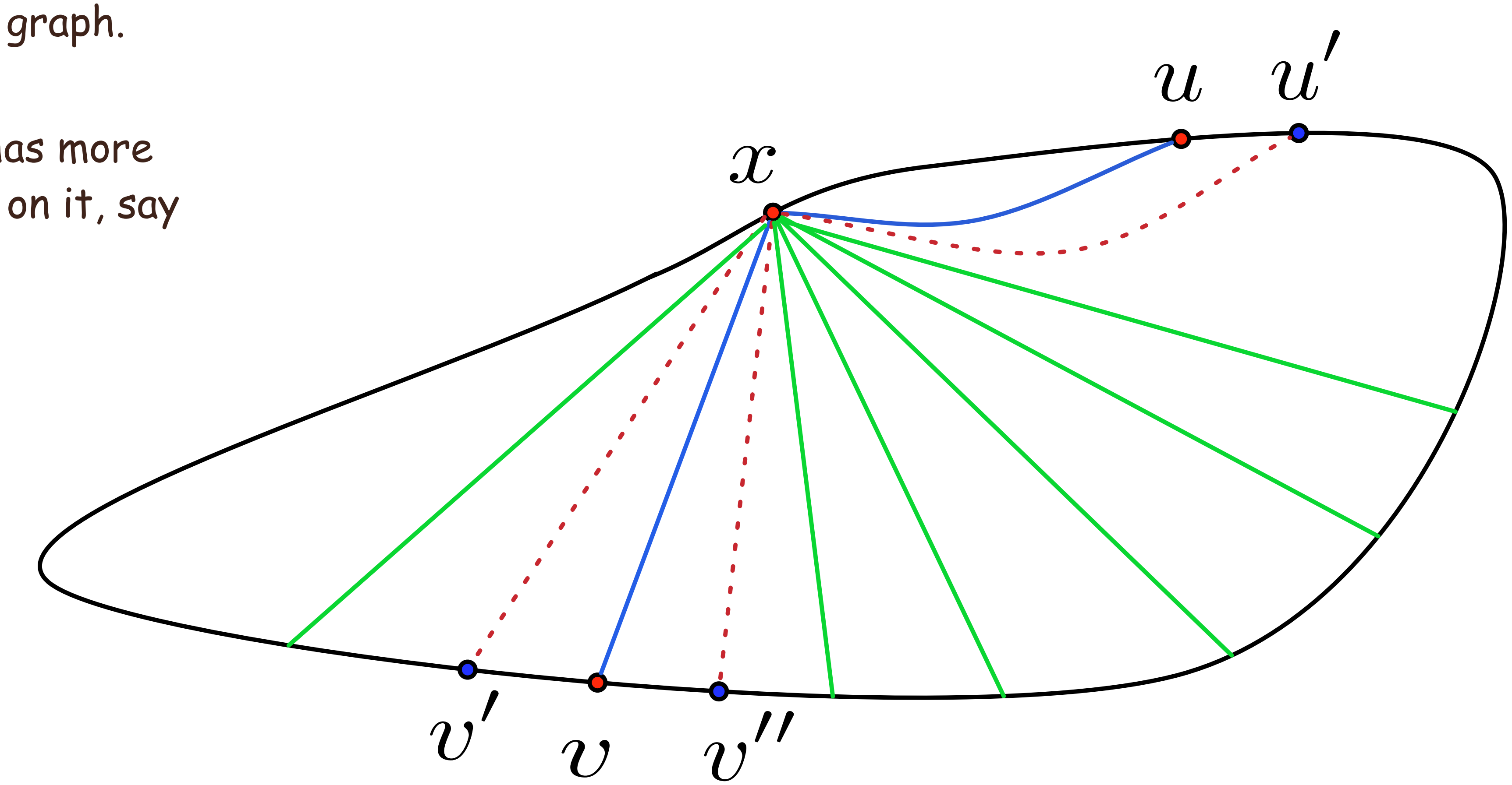
- If a link is necessary, add it to the graph.



If a vertex has more than 4k links projectively incident on it, then YES!

Cycle Augmentation

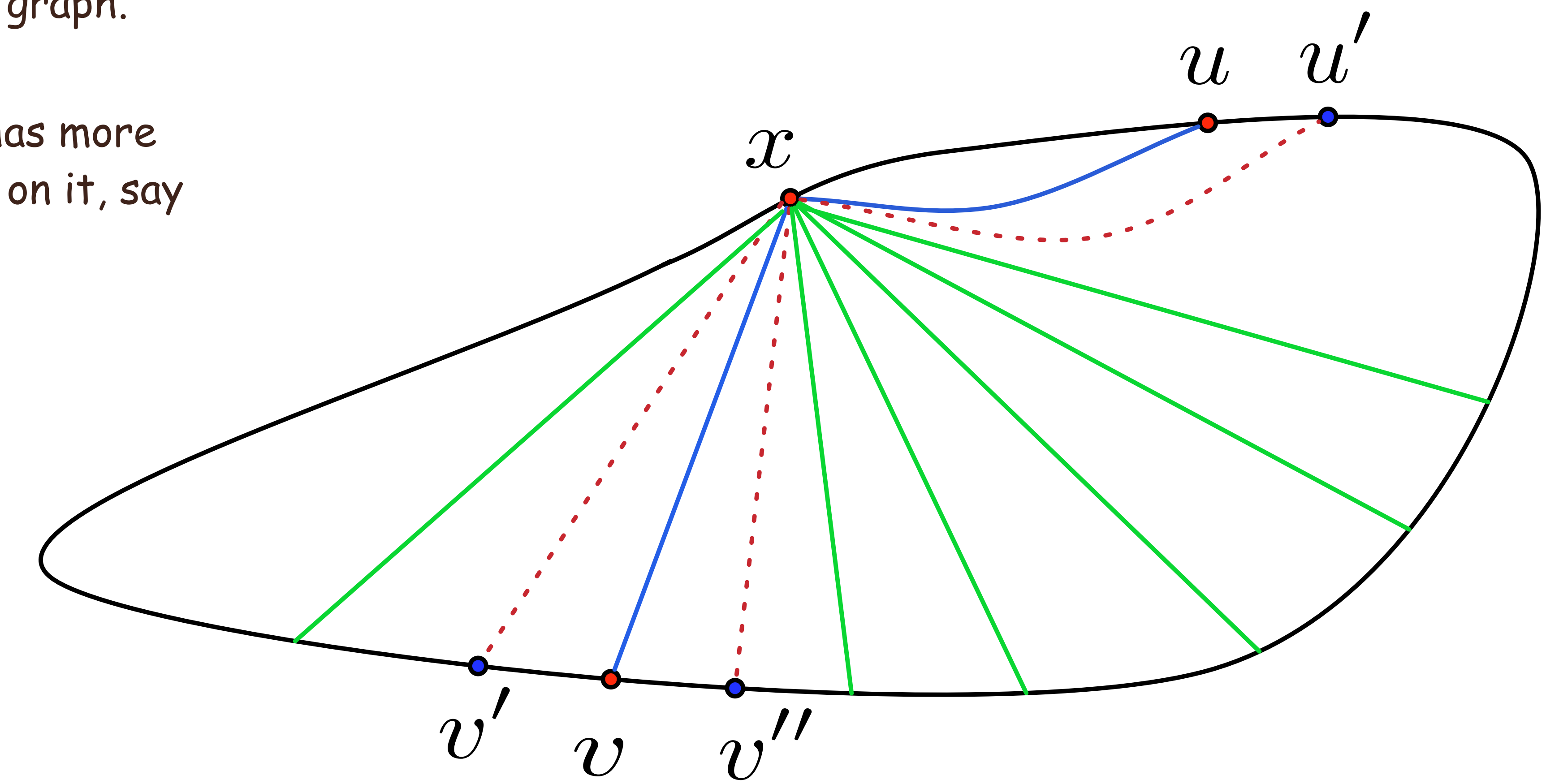
- If a link is necessary, add it to the graph.
- If a vertex has more than 4k links on it, say YES.



If a vertex has more than 4k links projectively incident on it, then YES!

Cycle Augmentation

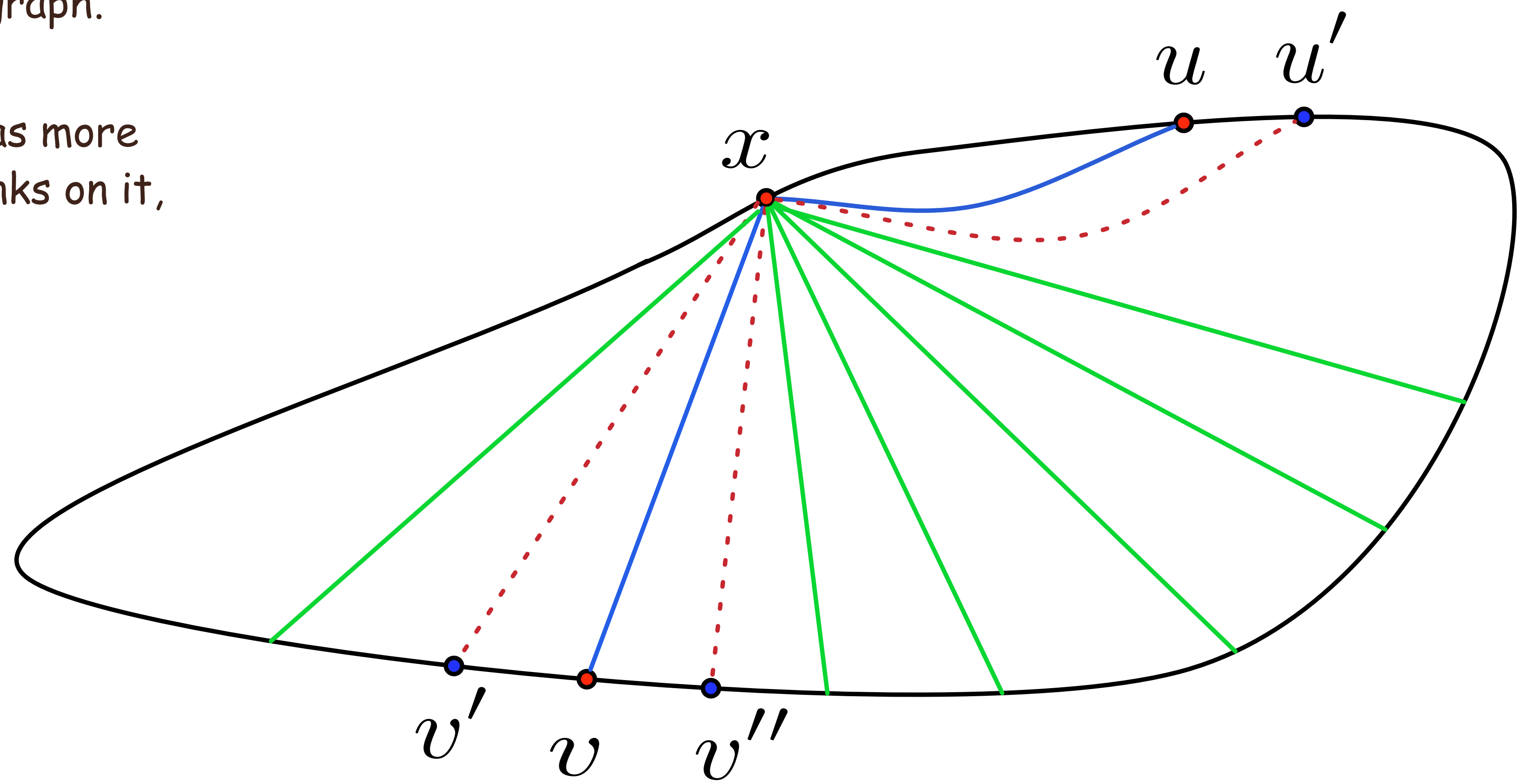
- If a link is necessary, add it to the graph.
- If a vertex has more than 4k links on it, say YES.



Stronger version: If there are more than |C| + 4k links projectively incident on the cycle, then YES!

Cycle Augmentation

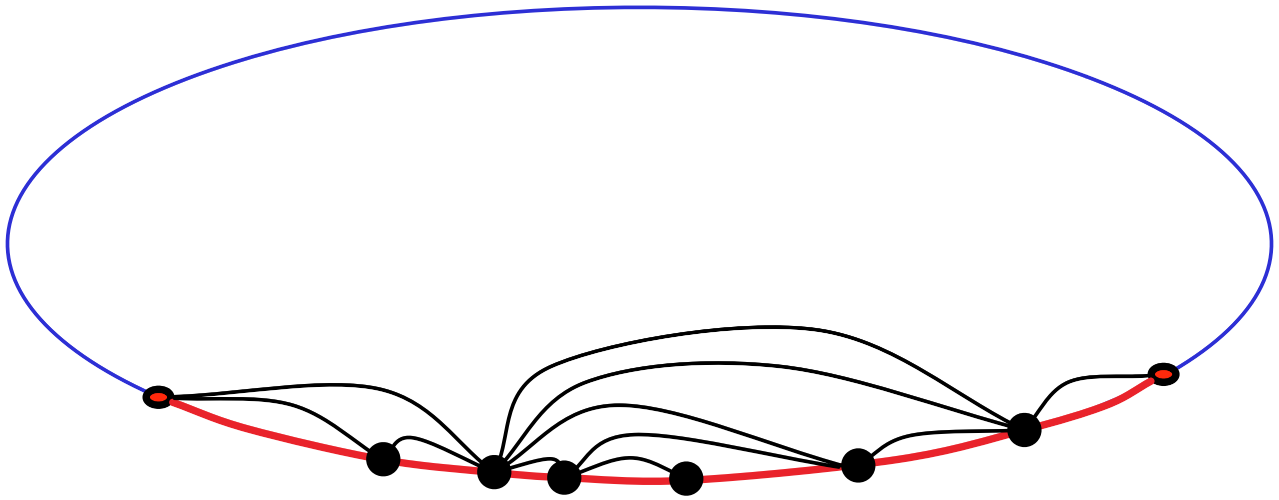
- If a link is necessary, add it to the graph.
- If a cycle C has more than |C|+4k links on it, say YES.

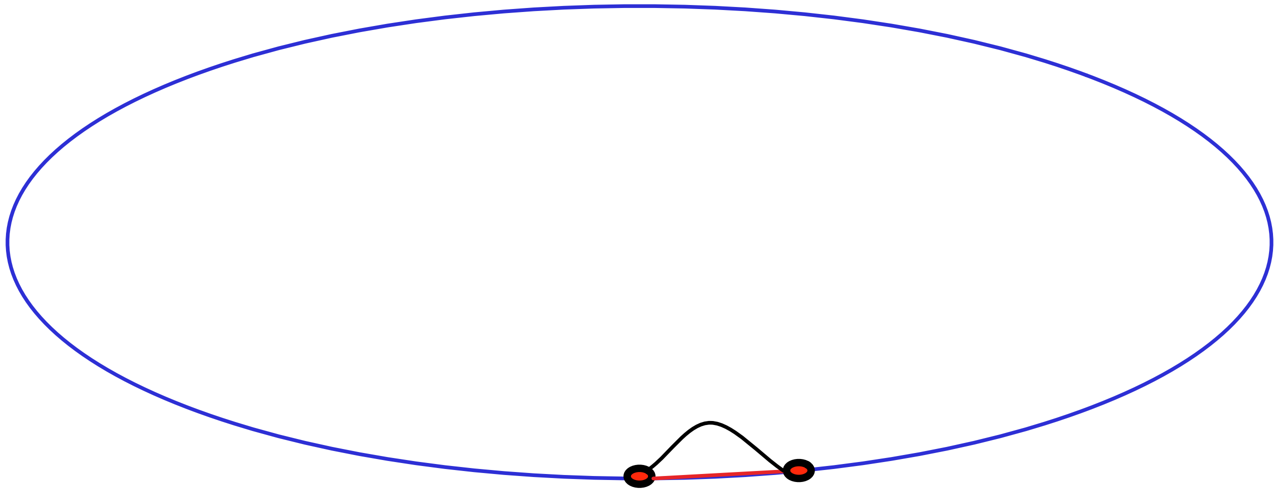


Stronger version: If there are more than |C| + 4k links projectively incident on the cycle, then YES!

Cycle Augmentation

- If a link is necessary, add it to the graph.
- If a cycle C has more than |C|+4k links on it, say YES.

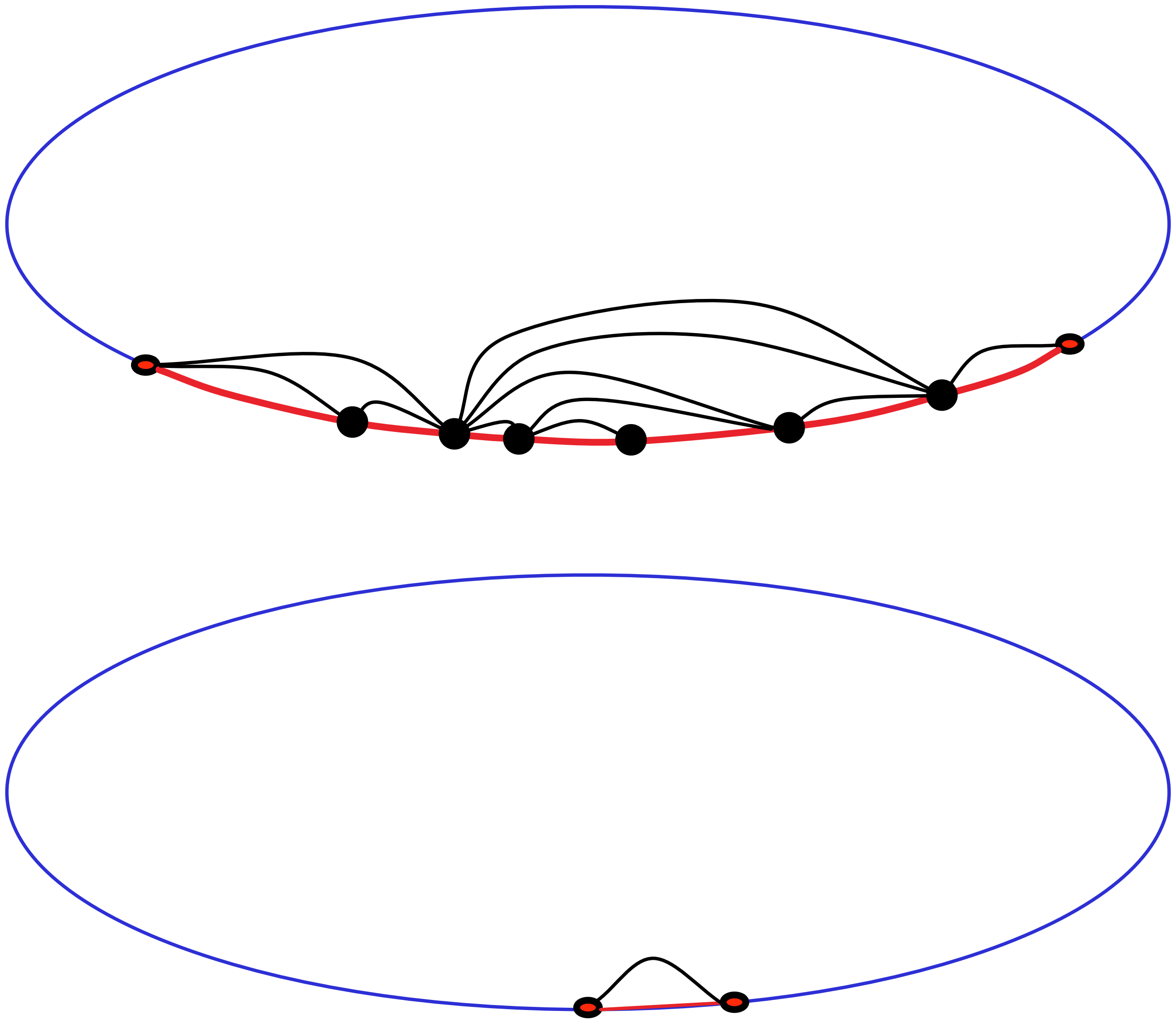




Laminar Strips: Optimal solution uses either t-1 or t links from the laminar strip.

Cycle Augmentation

- If a link is necessary, add it to the graph.
- If a cycle C has more than |C|+4k links on it, say YES.
- Shrink Laminar strips.



Reducing Laminar strips: Optimal solution uses either t-1 or t links from the laminar strip.

We can simulate this by gadgeting.

Cycle Augmentation : Reduction Rules

- If a link is necessary, add it to the graph.
- If a cycle C in the resulting cactus has $|C|+4k$ links incident on it then say YES.
- Shrink all laminar strips.

Cycle Augmentation

Lemma: When the reduction rules do not apply, every cycle has length $O(k)$.

We are still not quite done —> there could be too many cycles now in the cactus.

Lemma: In polynomial time, we can reduce number of cycles to $O(k)$.

Conclusion and open problems

- Connectivity augmentation problems— lots to explore w.r.t FPT and kernels.
- What about λ to $\lambda+c$ where $c>1$?
- What about vertex connectivity?
- Weighted variant of Question 2 (Deleting max weight set of at least k links)?

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!

Thank you for your attention!