Competitive Strategies for Online Clique Clustering

Christoph Dürr

joint work with

Marek Chrobak

Aleksander Fabijan

Bengt J. Nilsson

Oct 2016

Online graph arrival model

Vertices arrive one by one, revealing edges to existing vertices.

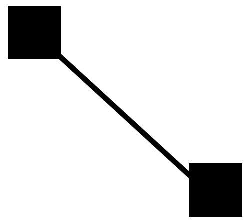
Irrevocable decisions have to be made at each step.

Online graph arrival model

Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

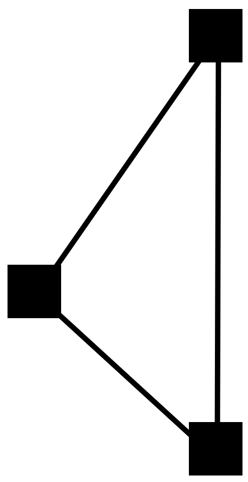
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

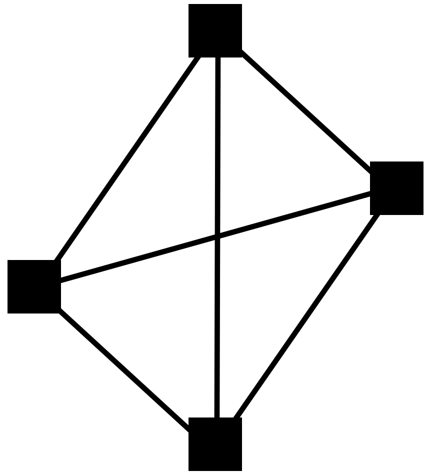
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

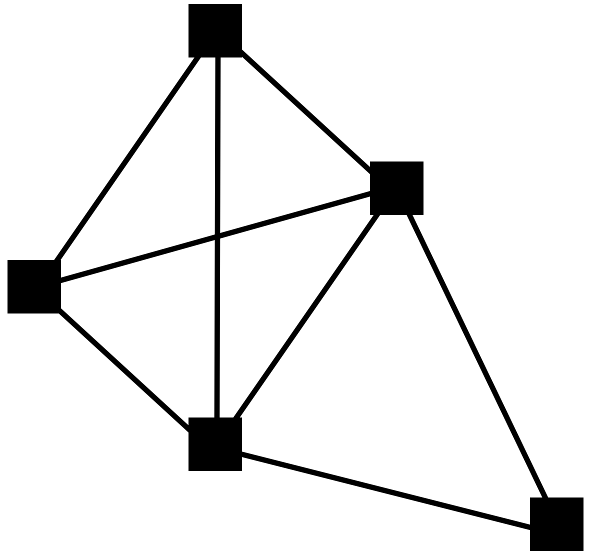
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

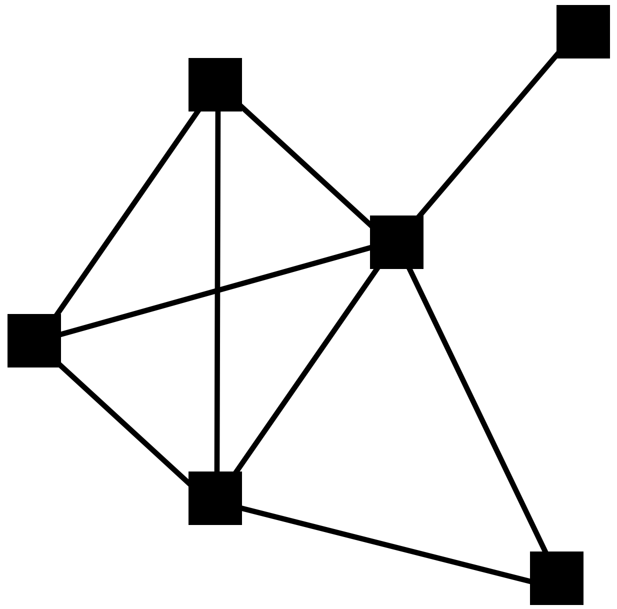
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

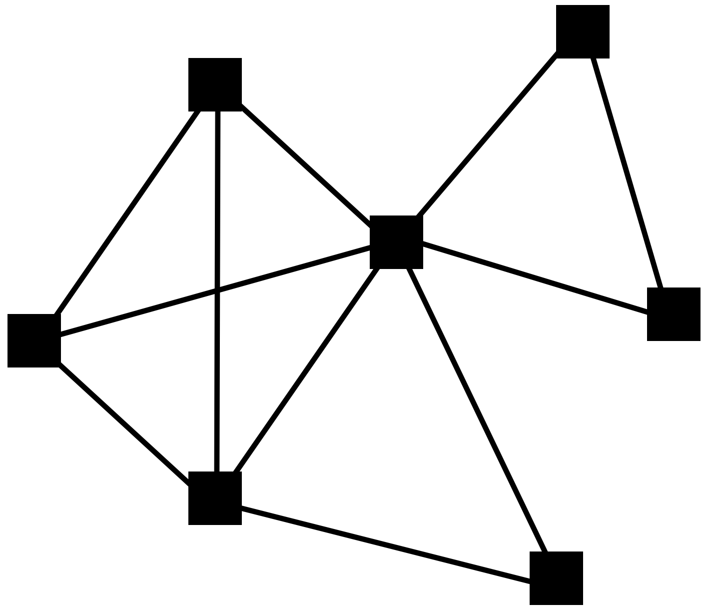
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

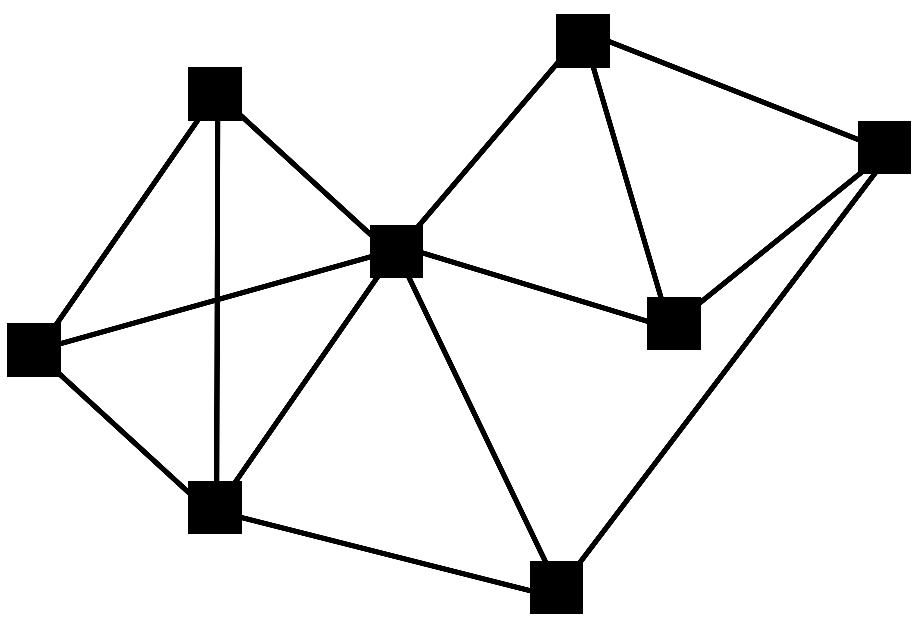
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

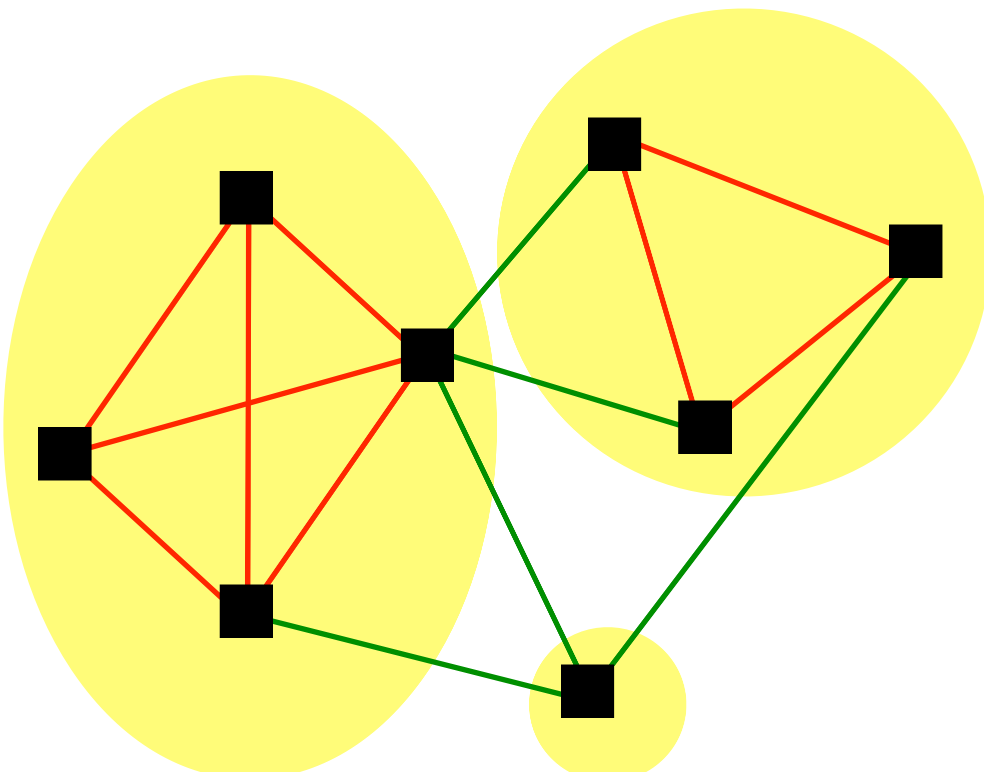
Online graph arrival model



Vertices arrive one by one, revealing edges to existing vertices.

Irrevocable decisions have to be made at each step.

Partition a graph into cliques



contribution of a clique of size k is $\binom{k}{2}$

Decide:
maintain partition into cliques by merging. Splitting not allowed.

Objective values:

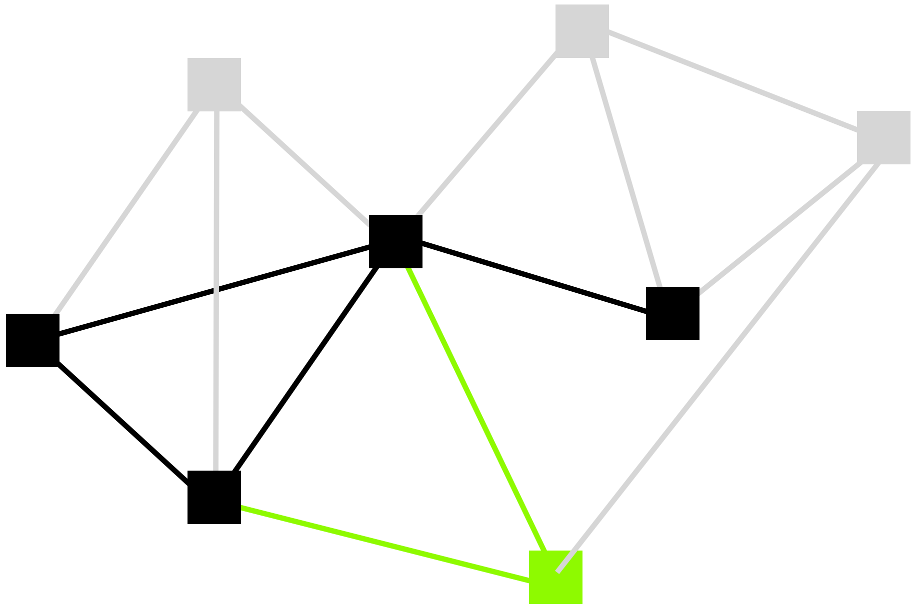
MaxCC
maximize number of edges inside the cliques

MinCC
minimize number of edges between the cliques

Motivation: DNA clone classification, gene expression profiling

Offline problem: not approximable within a factor of $n^{1-o(\varepsilon)}$ under some complexity hypothesis

Online model



- vertices arrive one by one, revealing adjacent edges
- strategy is allowed to merge cliques, but can never split them again
- **competitive ratio**: between obj. value of algorithms solution over optimal obj. value for current graph.

competitive ratio	LB	LB for doubling technique	UB
MaxCC	6	10.927	22.641
MinCC	n-2		n-2

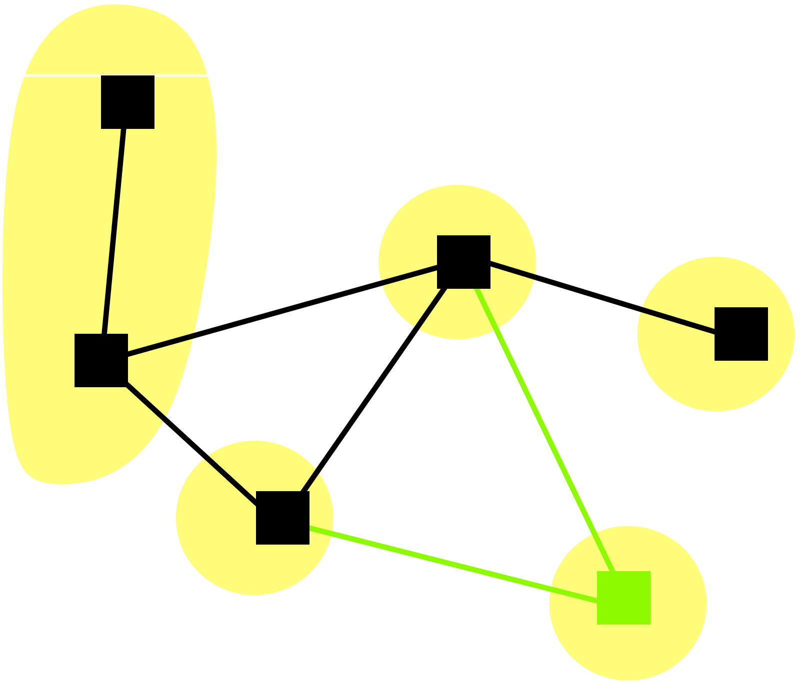
- There was a first paper by Fabijan, Nilsson, Persson, CIAC 2013

MaxCC

Online strategy model

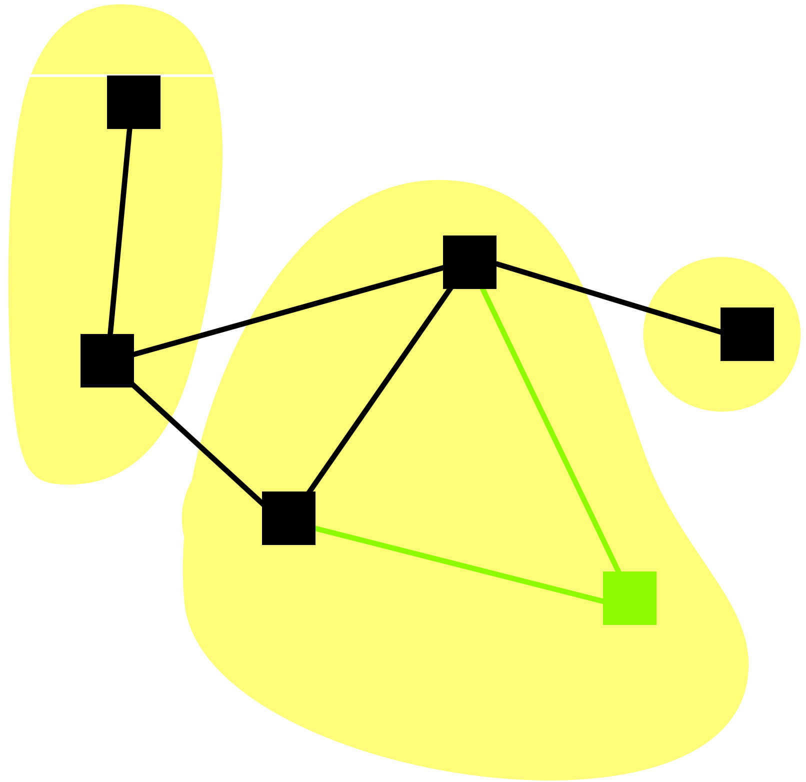
- Our strategy needs an oracle that tells him the optimal clustering for some subgraph
- In that sense it is not really an online algorithm serving requests in polynomial time, but rather a "strategy"
- Hence it is ok, for the problem to have constant competitive ratio, even though the offline problem cannot be approximated in polynomial time within a constant ratio

Our strategy OCC



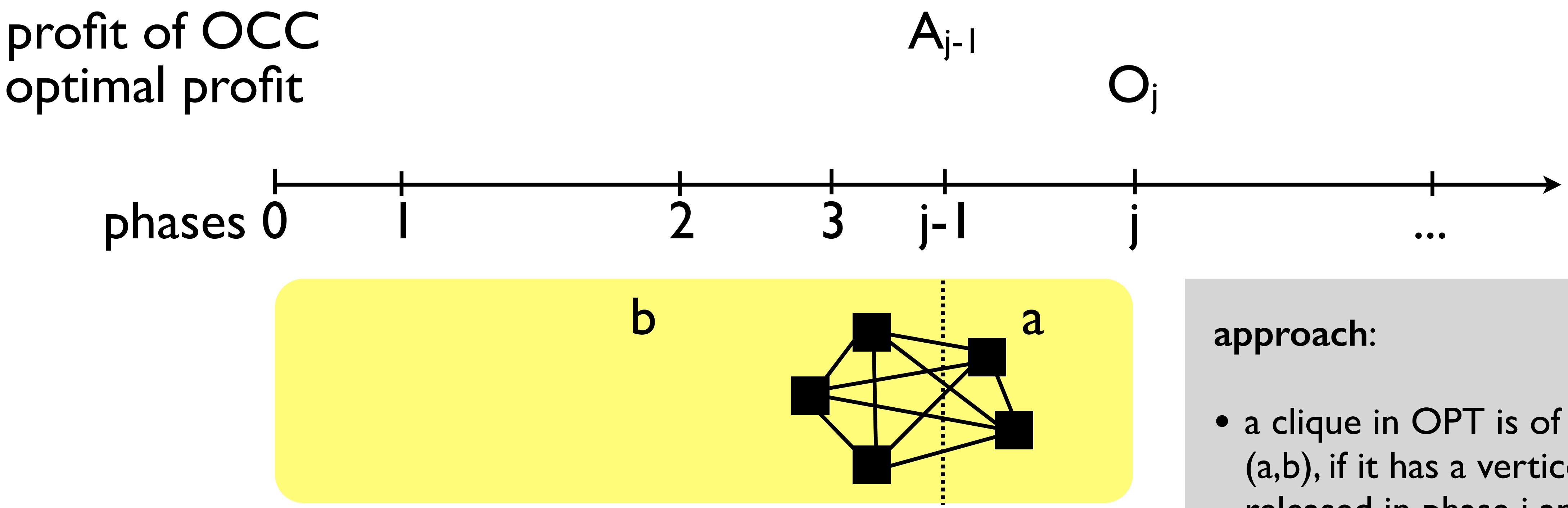
- Parameter $\gamma>1$
- stages, starting with $j=0$
- at each new vertex :
  - place it in a singleton clique
  - if singletons can form a clique partition with profit $\geq \gamma^j$, then merge according to this partition and move to stage $j+1$

Our strategy OCC



- Parameter $\gamma > 1$
- stages, starting with $j=0$
- at each new vertex :
  - place it in a singleton clique
  - if singletons can form a clique partition with profit $\geq \gamma^j$, then merge according to this partition and move to stage $j+1$

Upper bound on ratio of OCC



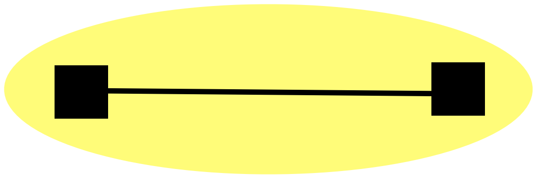
- .. is bounded by a geometric sequence
  $$A_j \geq \gamma^0 + \gamma^1 + \dots + \gamma^j \geq (\gamma^{j+1} - 1)/(\gamma - 1)$$
- ...
- we obtain a rough bound
  $$\text{assymptotic ratio} \leq (3+\sqrt{13})(5+\sqrt{13})^2 / (12(\sqrt{13}-1)) \approx 15.645$$
  for
  $$\gamma = (3+\sqrt{13})/2 \approx 3.303$$

approach:

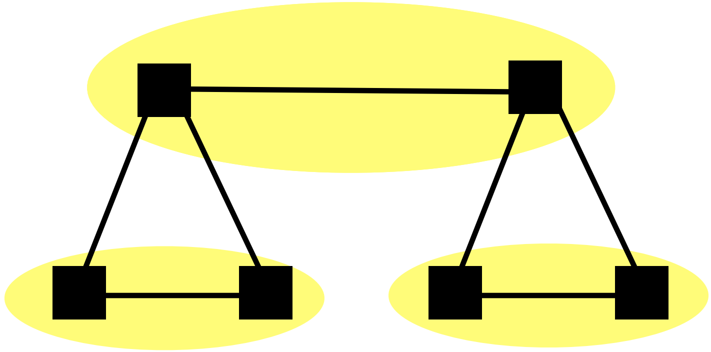
- a clique in OPT is of type (a,b), if it has a vertices released in phase j and b from previous phases.
- k_{ab} := number of cliques in OPT of this type.
- Use these numbers to bound A_{j-1}, and O_j

The hard life of an online strategy

- **Usual Game**: adversary release some vertex, strategy makes possibly some clustering, adversary releases ... (and so on, say forever)
- If strategy wants to reach competitive ratio R, he can postpone all merging until
  value of current solution * R < optimal solution for current graph



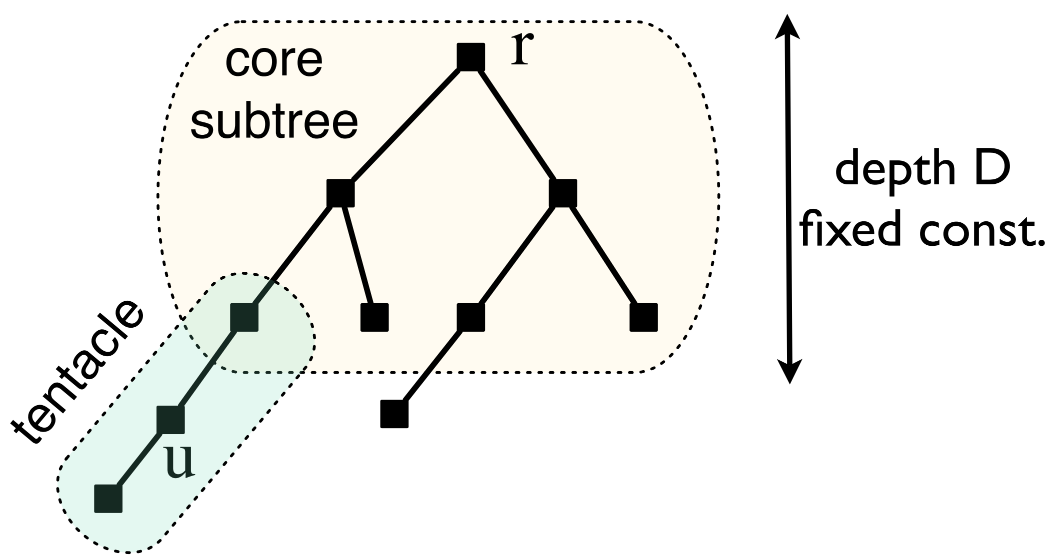
- Strategy needs to take the first edge, otherwise ratio is 1/0

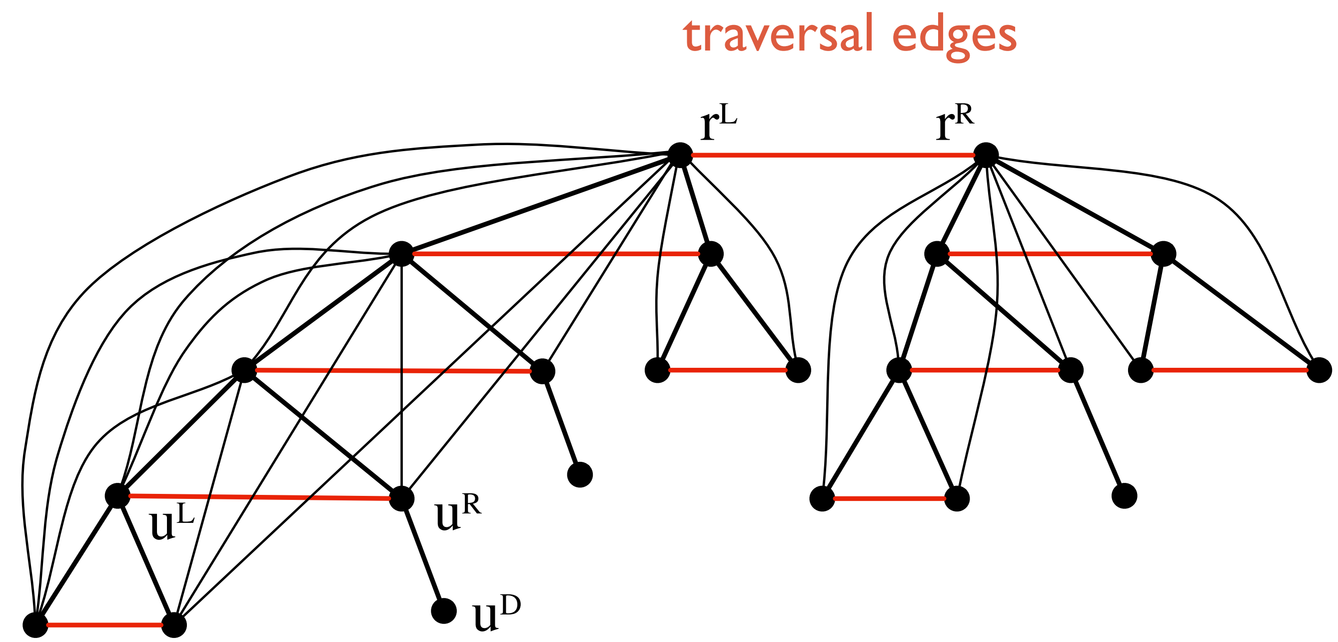


- But if endpoint is part of a clique, algo. can obtain only part of that clique. Ratio is already 6/3, let's improve it …

Lower bound of 6 on any strategy

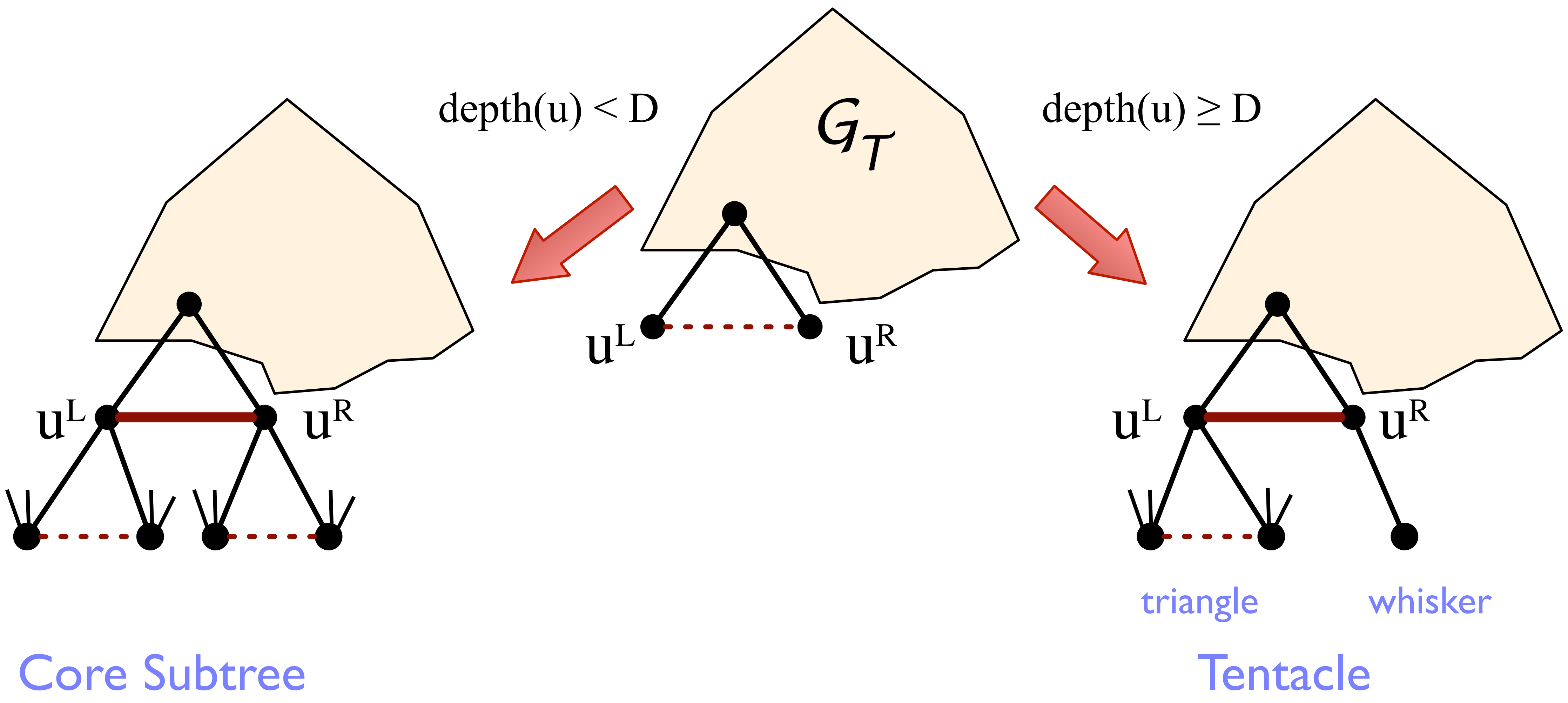
- graph generated by strategy is described by a "skeleton tree"





How the adversary extends the tree

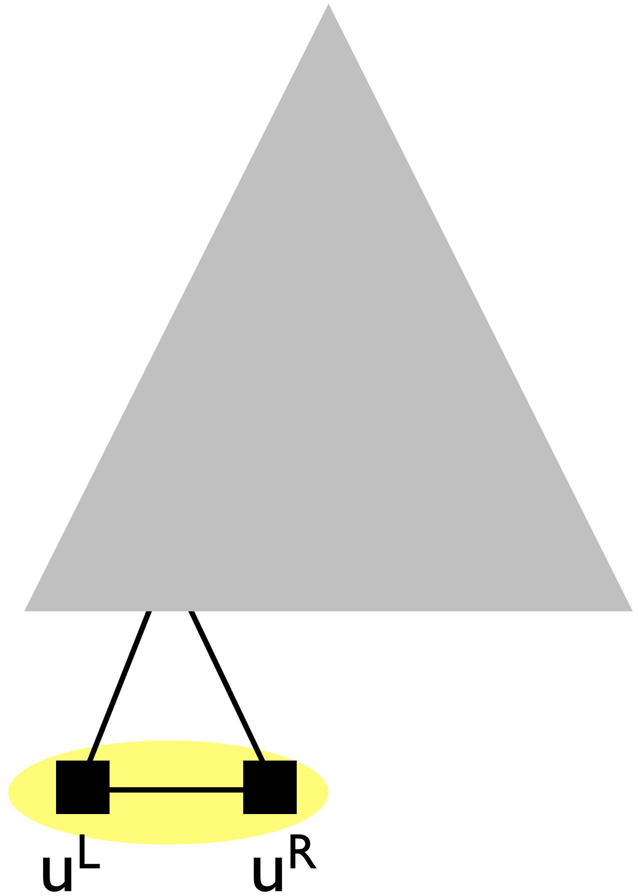
- if the strategy collects the edge (u^L,u^R)

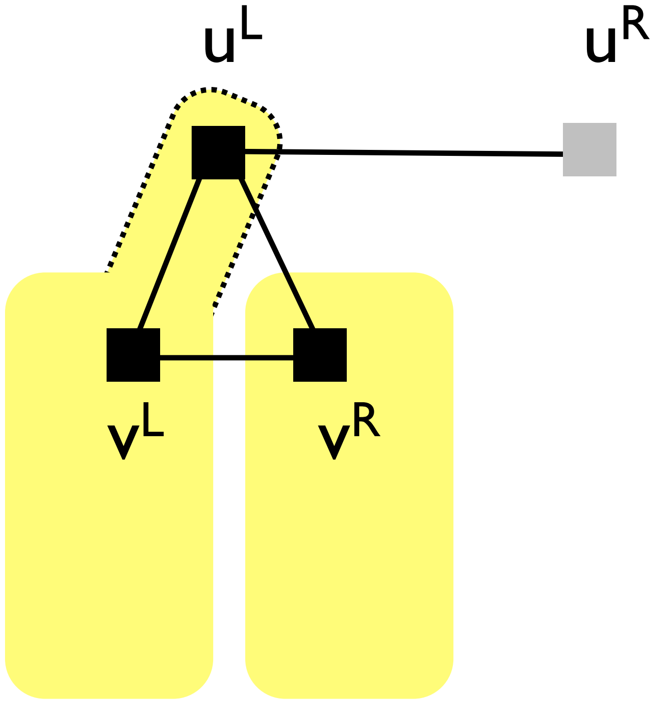


An adversarial clique partition

Bottom up description:

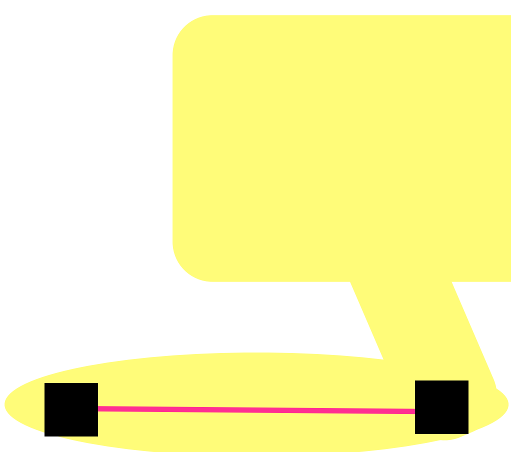
- If u is a leaf, (u^L, u^R) are in a clique
- If u has a left son v, u^L is added to one of the cliques containing v^L or V^R
- If u has a right son v, (idem)

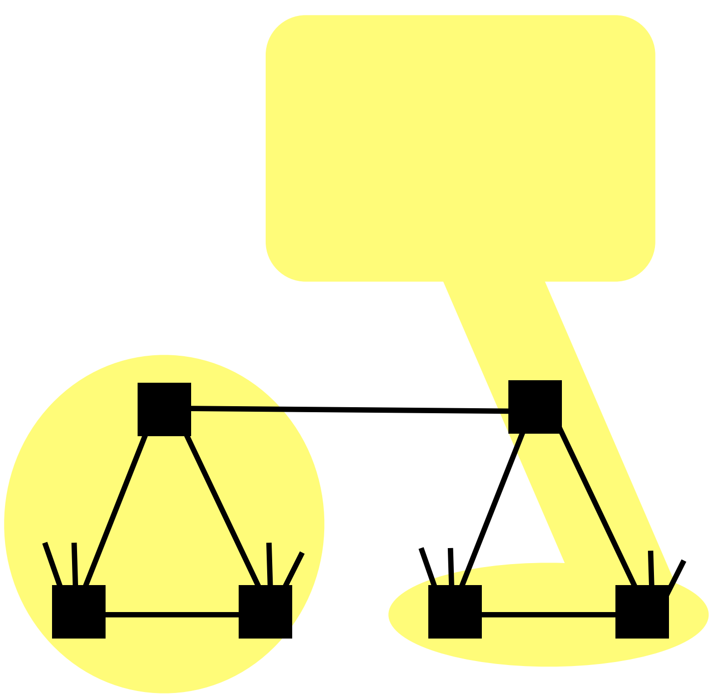


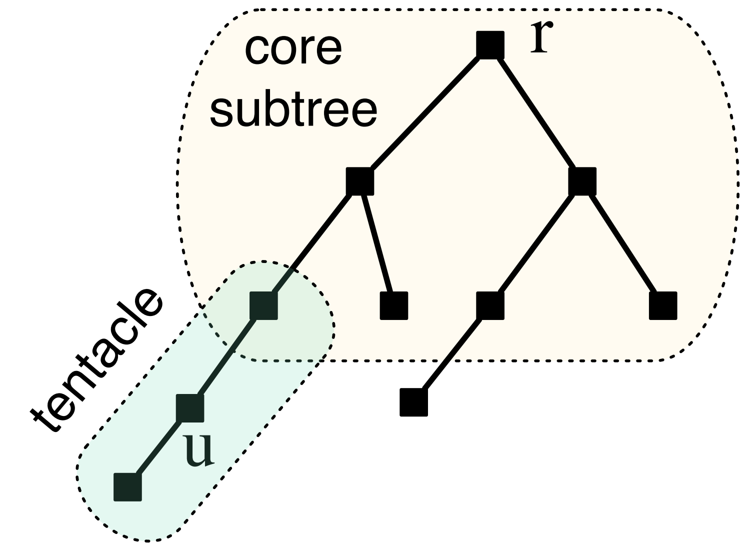


The argument

- We show that for every traversal edge collected by the strategy, the adversary releases more vertices and can increase his gain by at least 6
- If the strategy decices to stop at some point, the ratio is ≥6
- If the game lasts long enough, the adversary could stop, as his gain is quadratic in tentacle length, while strategy's gain is only linear







MinCC

Any strategy has ratio at least n-2

- Adversarial argument:

Any strategy has ratio at least n-2

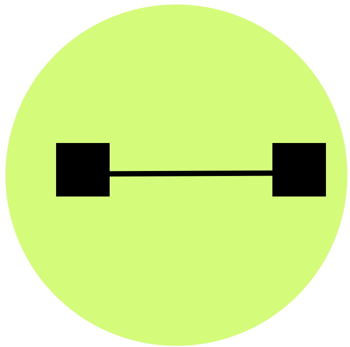
- Adversarial argument:

- Algorithm has to merge this edge, otherwise ratio would be 1/0

Any strategy has ratio at least n-2

- Adversarial argument:

- Algorithm has to merge this edge, otherwise ratio would be 1/0



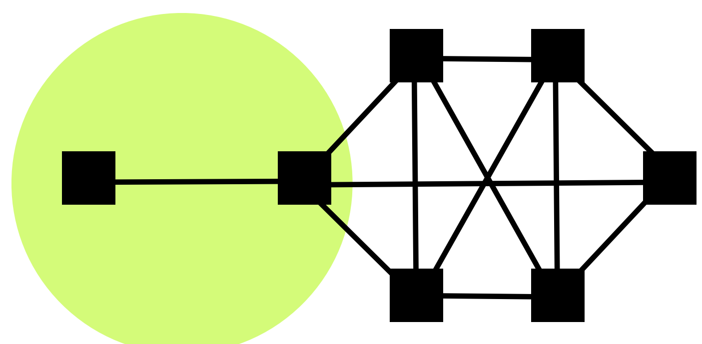
Partition produced by the algorithm

Any strategy has ratio at least n-2

- Adversarial argument:

- Algorithm has to merge this edge, otherwise ratio would be 1/0

- Algorithm can merge only part of the clique. Ratio is (n-2)/1



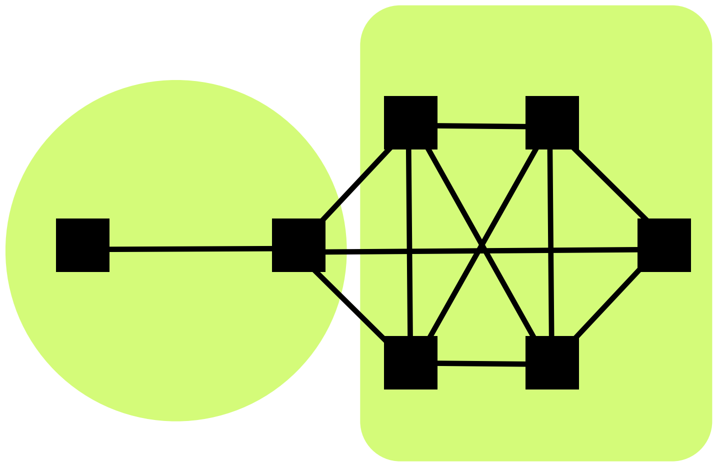
Partition produced by the algorithm

Any strategy has ratio at least n-2

- Adversarial argument:

- Algorithm has to merge this edge, otherwise ratio would be 1/0

- Algorithm can merge only part of the clique. Ratio is (n-2)/1

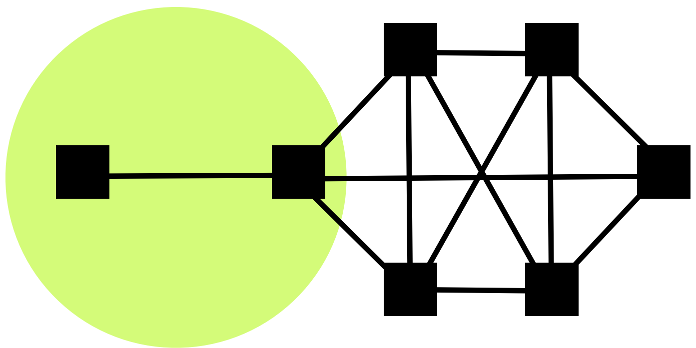


Partition produced by the algorithm

Any strategy has ratio at least n-2

- Adversarial argument:

- Algorithm has to merge this edge, otherwise ratio would be 1/0

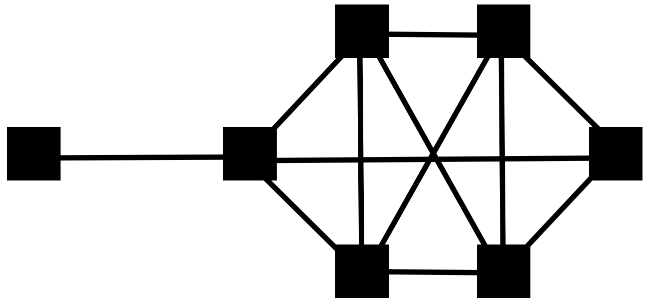


Partition produced by the algorithm

- Algorithm can merge only part of the clique. Ratio is (n-2)/1

Any strategy has ratio at least n-2

- Adversarial argument:
- Algorithm has to merge this edge, otherwise ratio would be 1/0
- Algorithm can merge only part of the clique. Ratio is (n-2)/1

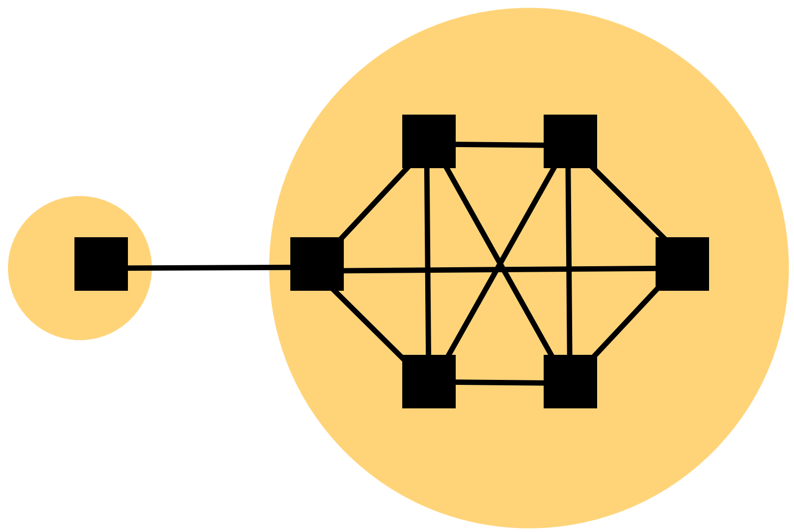


Any strategy has ratio at least n-2

- Adversarial argument:

- Algorithm has to merge this edge, otherwise ratio would be 1/0

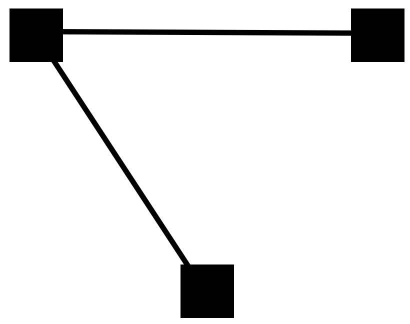
- Algorithm can merge only part of the clique. Ratio is (n-2)/1



Optimal partition

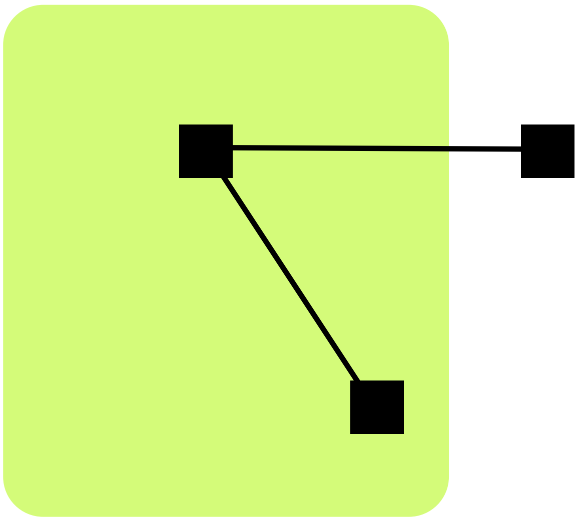
Any greedy strategy has ratio at most n-2

- **greedy:** while merges are possible, the algorithm does some merge
- **key observation:** for 3 vertices connected by 2 edges, at least one of the edges is a non-clique edge



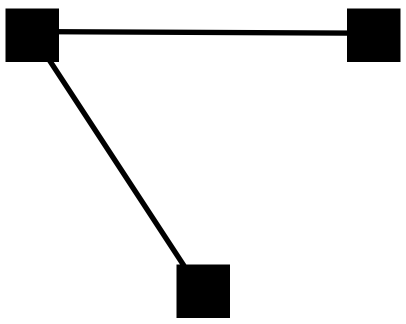
Any greedy strategy has ratio at most n-2

- **greedy:** while merges are possible, the algorithm does some merge
- **key observation:** for 3 vertices connected by 2 edges, at least one of the edges is a non-clique edge



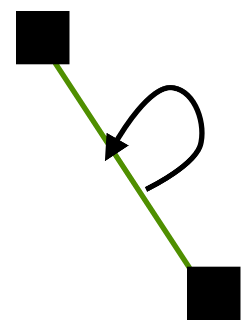
Any greedy strategy has ratio at most n-2

- **greedy:** while merges are possible, the algorithm does some merge
- **key observation:** for 3 vertices connected by 2 edges, at least one of the edges is a non-clique edge

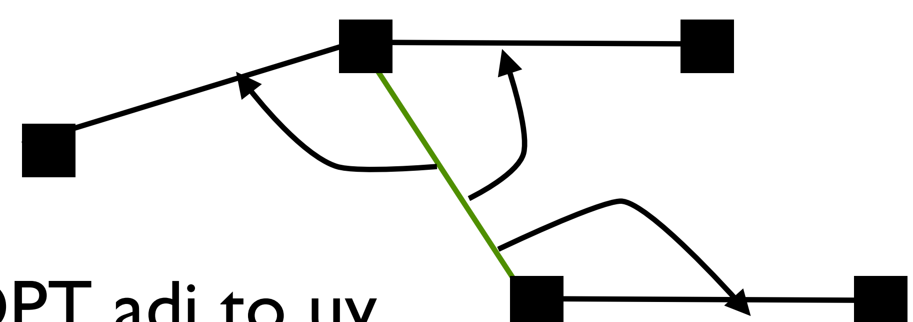


Any greedy strategy has ratio at most n-2

- **compare** a greedy solution ALG with optimal solution OPT
- **charging scheme:** charge every non-clique edge in ALG to non-clique edges by OPT
- **show:** that every non-clique edge obtains at most n-2 charges

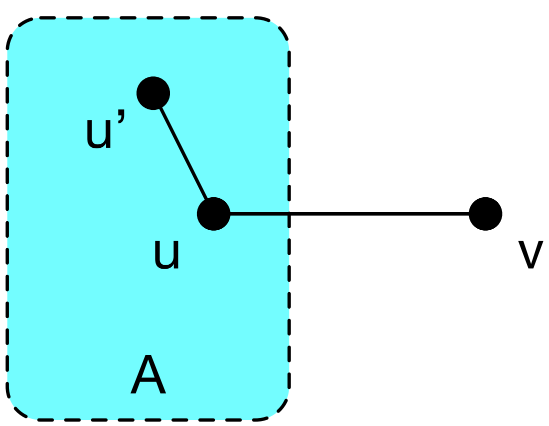


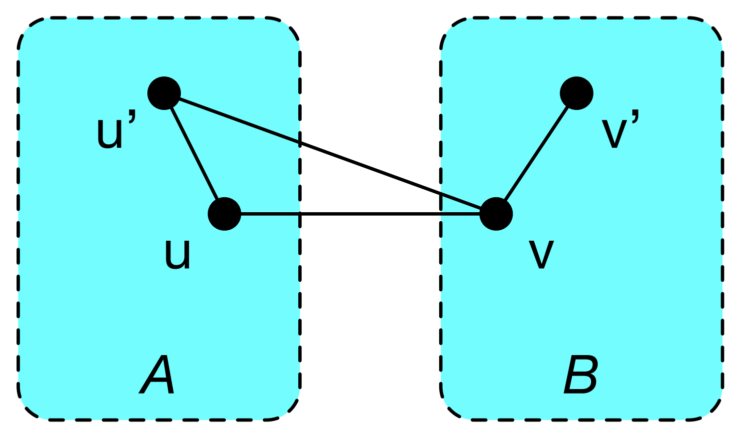
- Given non-clique edge uv in ALG:
  - if uv is also non-clique edge in OPT: **self-charge** uv to uv
  - else **proximate charge** uv in equal fractions to all non-clique edges in OPT adj to uv



Any greedy strategy has ratio at most n-2

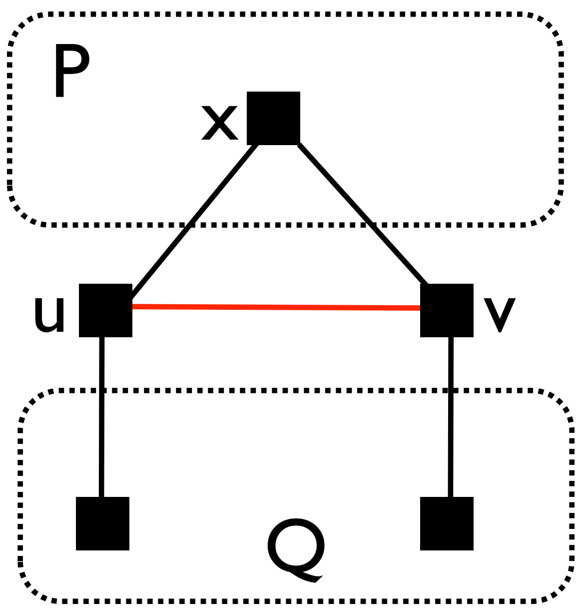
- **Claim** every non-clique edge uv in OPT has an adjacent non-clique edge in OPT
  - Say v arrived after u. Let A be a clique in ALG when v arrived with u∈A.
  - Why did Greedy not add v to A ?
  - Well the clique B containing v could not be merged with A.
  - There must be 3 vertices connected by only 2 edges:
    either uv and some vertex in A
    or v and some vertex in A and some in B





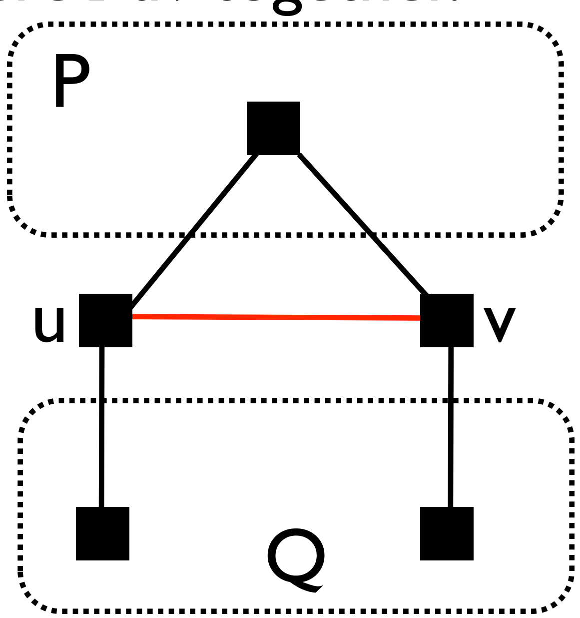
Any greedy strategy has ratio at most n-2

- **Count** how much charge can a non-clique edge uv in OPT obtain
  - P = set of vertices connected to both endpoints
    Q= set of vertices connected to a single endpoint
  - proximate charge from Q is at most |Q|
  - for every x∈P, one of uv or ux must be non-clique edge in OPT
    Hence if ux proximate charges to uv, then with fraction ≤ ½
  - Charge from P is at most |P|/2.



Any greedy strategy has ratio at most n-2

- **We have** $|P|+|Q|\leq n-2$
- **Claim** $|P|/2+|Q|+1 \leq n-2$
  - ok if uv is cluster edge in ALG
  - ok if $|P|\geq 2$
  - ok of Q contains cluster edges in ALG
  - case $|P|\leq 1$ and Q contains only non-cluster edges in ALG cannot happen. Greedy would have clustered uv together.



merci