Delegate and Conquer: On Minimum Degree Minimum Cost Spanning Tree

R. Ravi

Carnegie Mellon University

Joint Work with Mohit Singh

Minimum Degree MST problem

Given an undirected graph G, cost function c, a bound B on maximum degree

1. Return an MST which satisfies the degree bounds, or
2. Show the degree bounds are infeasible for any MST of G.

Minimum Degree MST problem

Given an undirected graph G, cost function c, a bound B on maximum degree

1. Return an MST which satisfies the degree bounds, or
2. Show the degree bounds are infeasible for any MST of G.

- Problem is NP-complete
- Approximating cost and satisfying the degree bounds exactly is not possible.
- We consider the case for approximating the degree but satisfying the cost exactly, i.e., solution must be a MST.
- Bi-criteria approximations for the problem have been studied.

Previous Work

- Fischer '93 returns a MST with maximum degree O(B+log n) or shows infeasibility for degree bounds B

- Chaudury et al '05 give a quasi-polynomial time algorithm which returns a MST with maximum degree O(B+logn/log logn) or shows infeasibility for degree bounds B

Unweighted Case

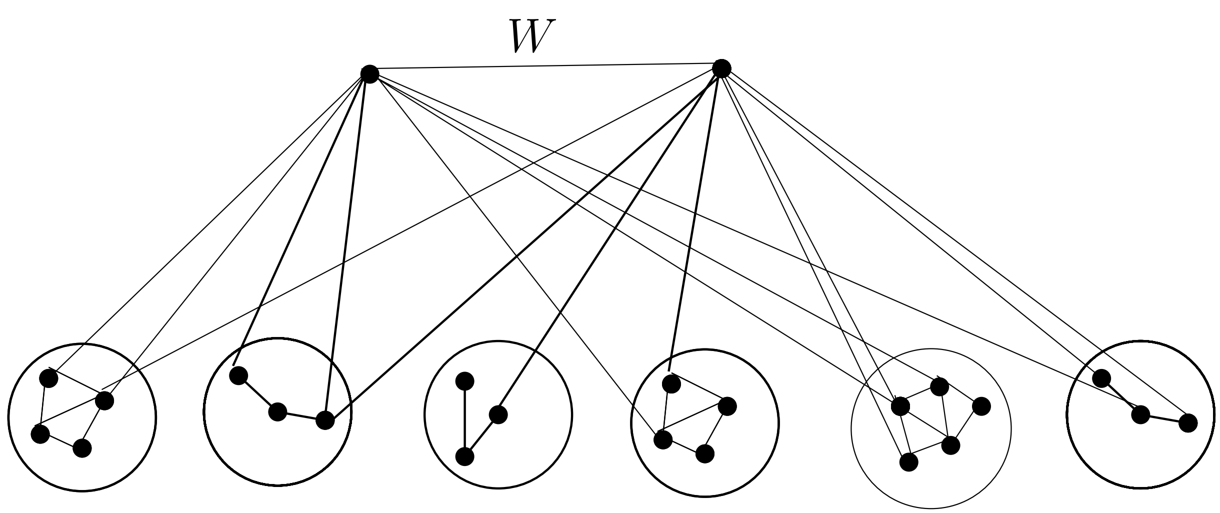
- Theorem [Furer & Raghavachari '92]: Given a unweighted graph and degree bounds B_v for vertex v, a polynomial time algorithm returns a Witness set $W \subset V$ and a tree T such that

Solution	Infeasibility
If $W=\phi$ then $deg_T(v) \leq B_v+1$ for each $v \in V$.	No tree of G satisfies the degree bounds on W.

Cf. Vizing's Theorem for edge-coloring a graph.

Witness Set

- $W \subset V$ such that if $C_1,\ldots,C_r$ are components in $G \backslash W$.
- Total degree of nodes of W in any tree $T \geq r+|W|-1$
- Instance infeasible if $\sum_{w \in W} B_w < r + |W| - 1$



Our Main Result (ICALP '06)

- Theorem: There exists a polynomial time algorithm which given a graph $G=(V,E)$ with cost function c on edges and degree bounds B_v for each vertex v either
  - (Infeasibility) Shows that no MST satisfies the degree bounds.
  - (Solution) Returns a MST such that $deg_T(v) \leq B_v+k$

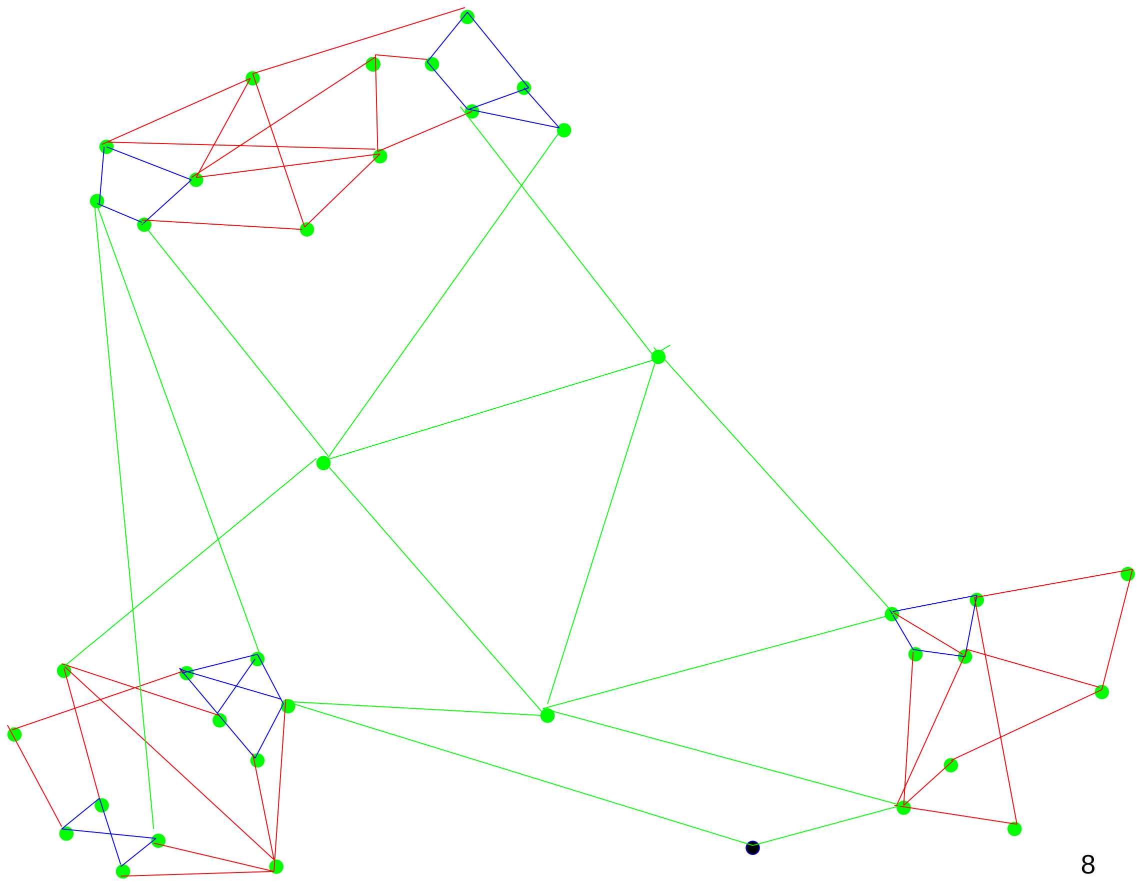
  Infeasibility is via a linear programming relaxation

Here k = no. of distinct costs in an MST of G

Structure of an MST

- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

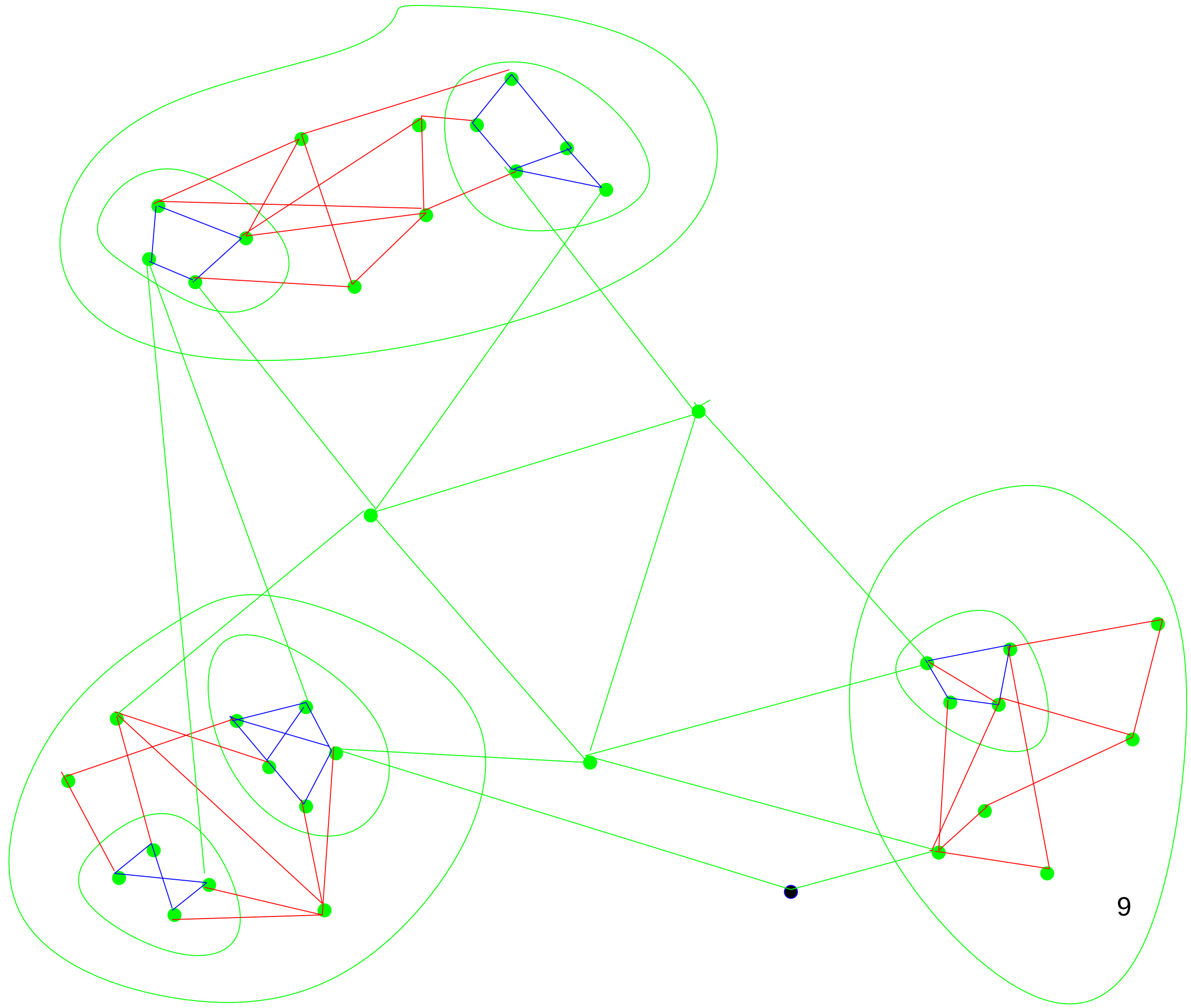
Colors and Costs:
Blue=1
Red=2
Green=3



Structure of an MST

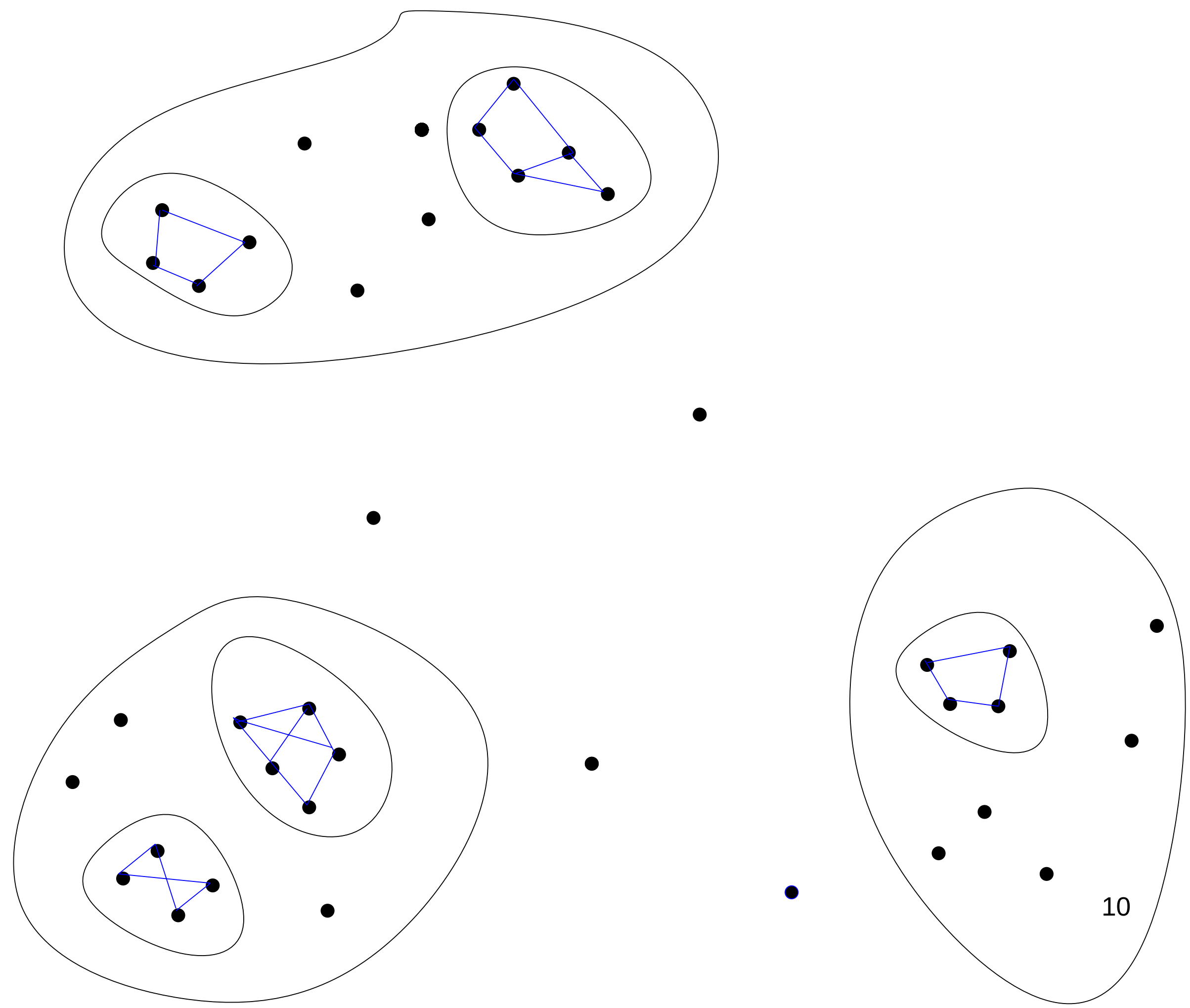
- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

Colors and Costs:
Blue=1
Red=2
Green=3



Structure of an MST

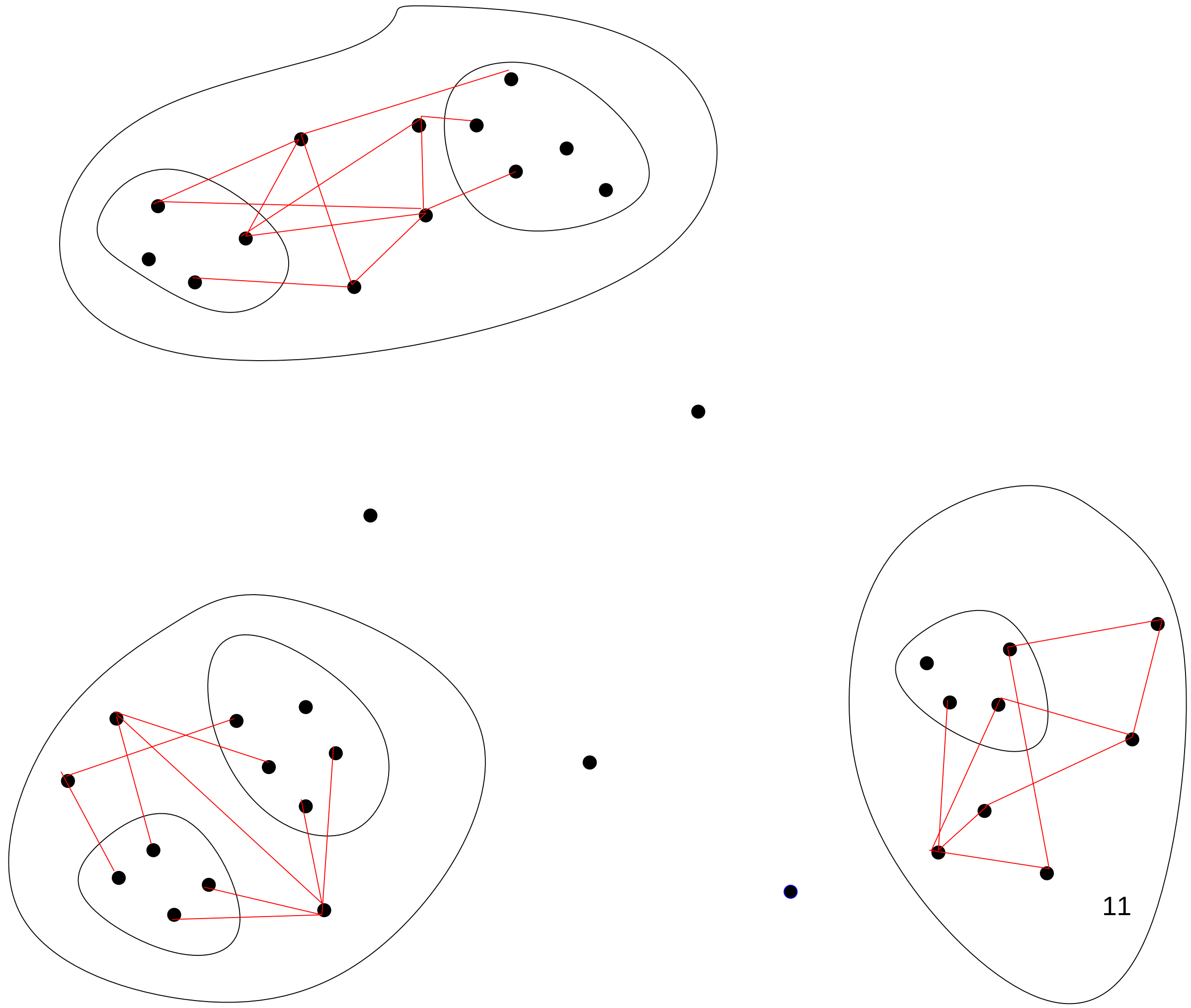
- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.



Colors and Costs:
Blue=1
Red=2
Green=3

Structure of an MST

- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

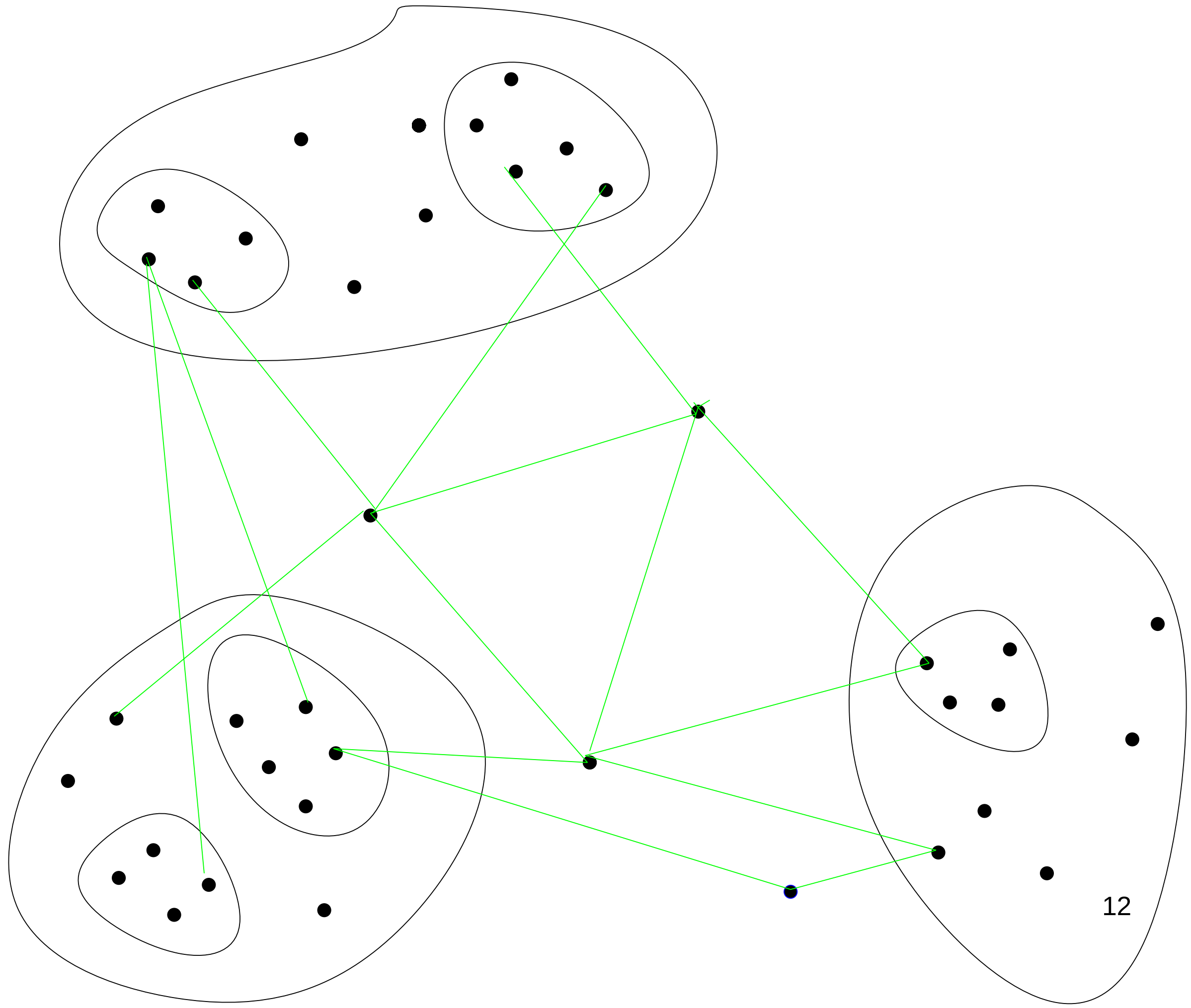


Colors and Costs:
Blue=1
Red=2
Green=3

Structure of an MST

- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

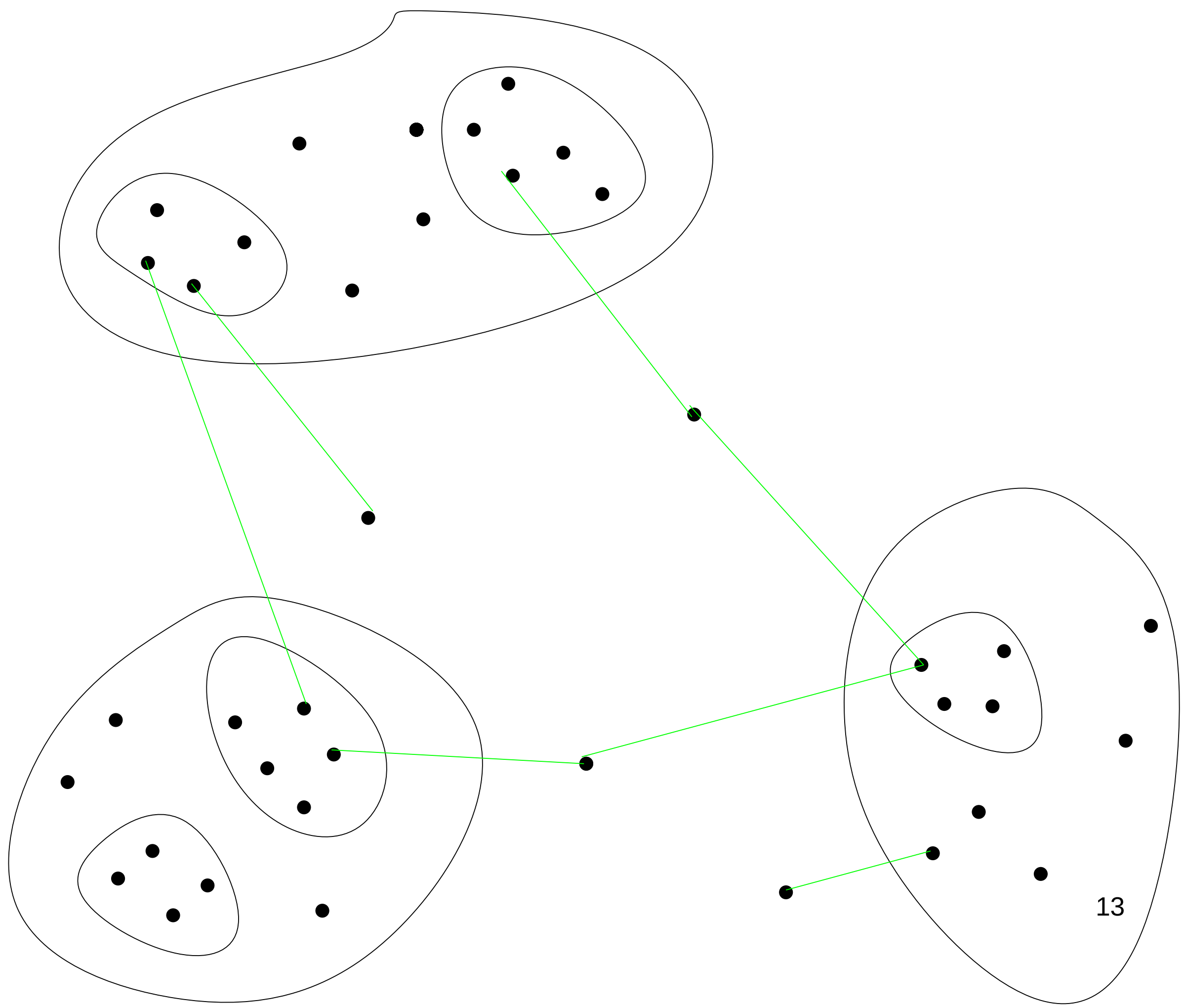
Colors and Costs:
Blue=1
Red=2
Green=3



Structure of an MST

- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

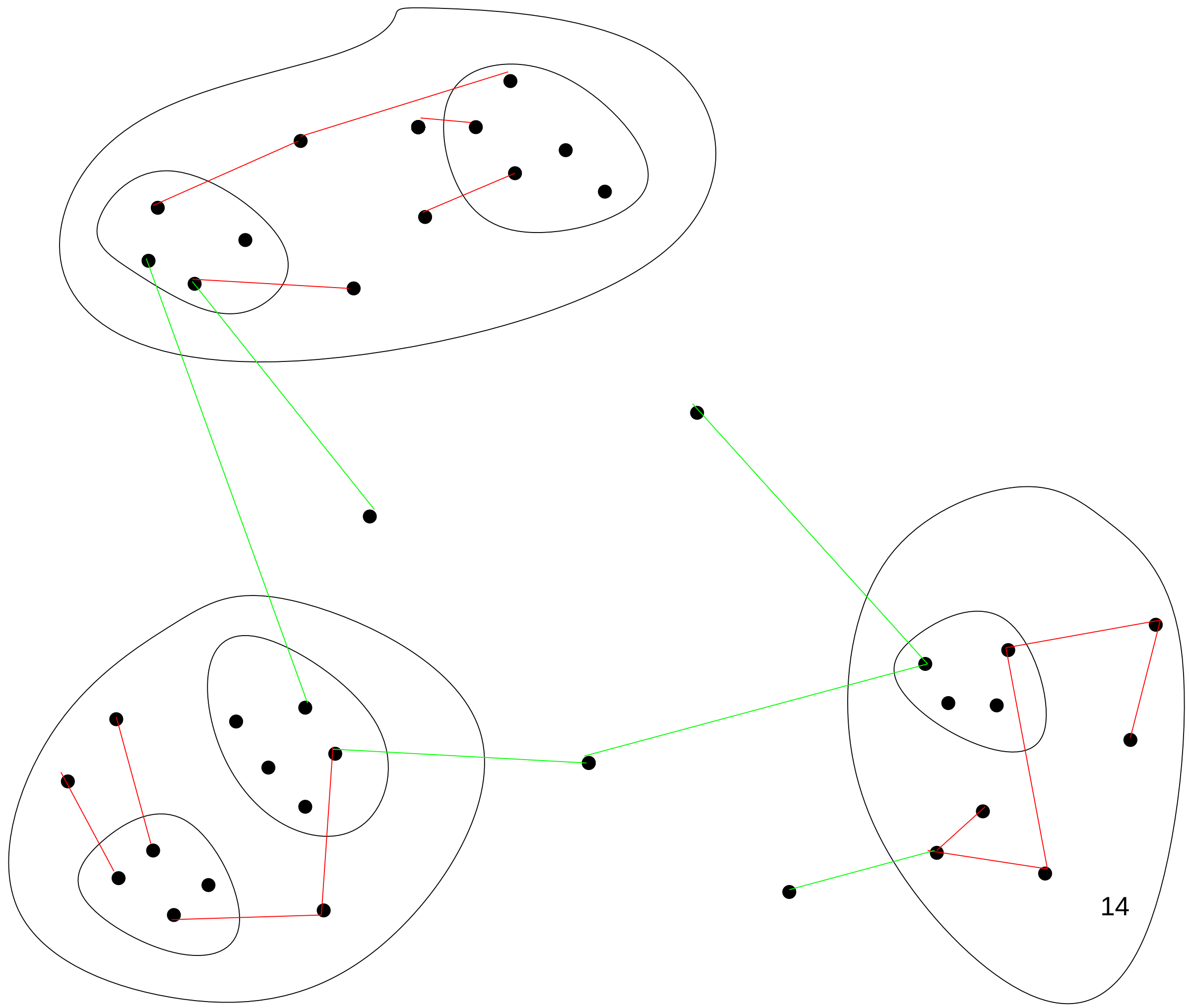
Colors and Costs:
Blue=1
Red=2
Green=3



Structure of an MST

- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

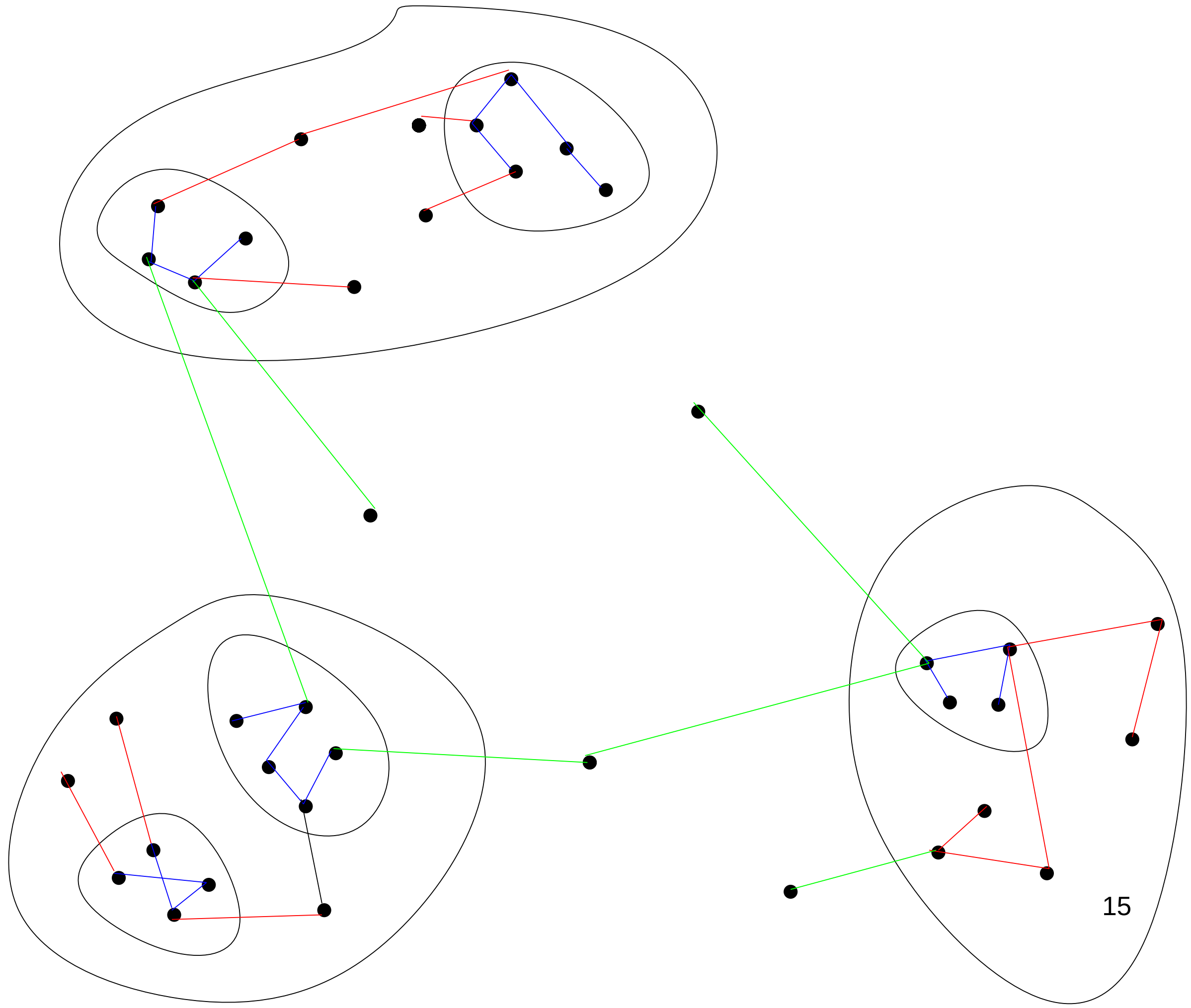
Colors and Costs:
Blue=1
Red=2
Green=3



Structure of an MST

- Any MST can be constructed by decomposing the graph into forests corresponding to each cost class.

Colors and Costs:
Blue=1
Red=2
Green=3



Forests over Forests

- Theorem [Ellingham & Zha '00]: Given a connected unweighted graph, forest F, and degree bounds B_v for vertex v, there is a polynomial time algorithm that returns a witness set W and F-tree T (that connects F) such that either

Infeasibility ($W \neq \phi$) or Solution ($W = \phi$)

W "violates" the degree bounds for any F-tree T'.

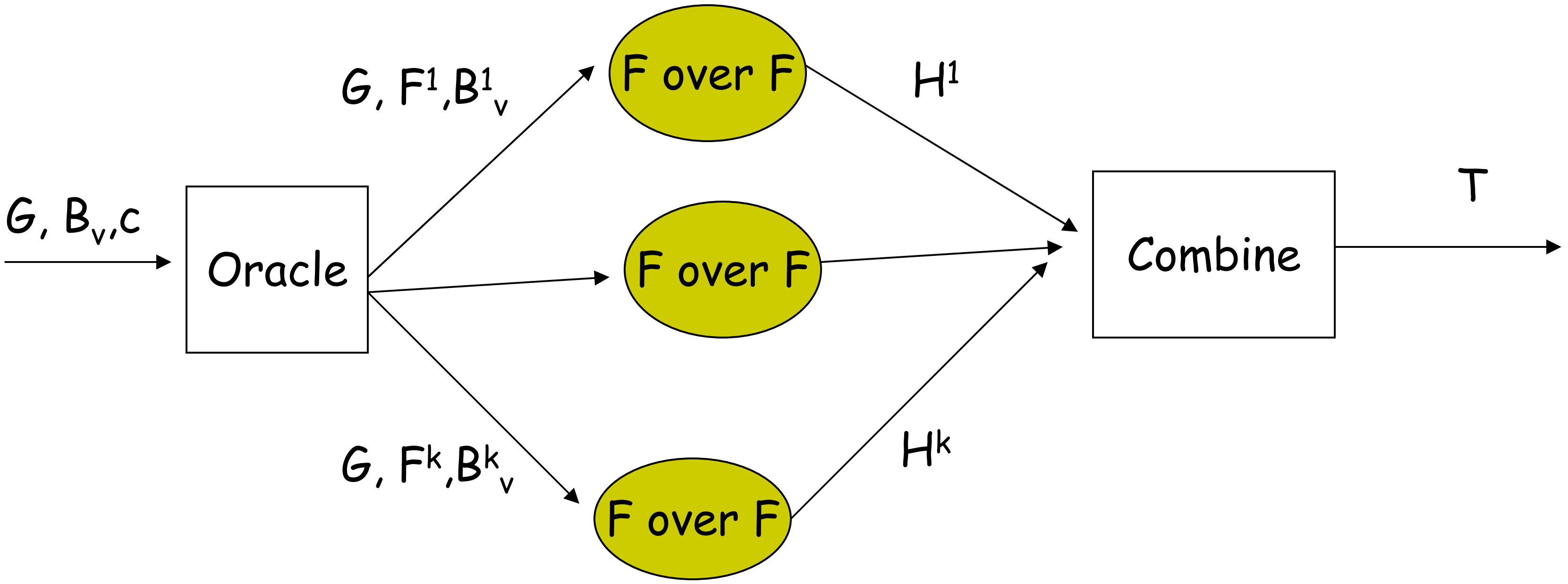
$deg_T(v) \leq B_v + 1$.

Independence (Almost)

- The degree bound needs to be divided among different cost classes.
- Need an Oracle to partition each $B_v=B^1_v+\ldots+B^k_v$ for each v.
- Use B^i_v for constructing the appropriate forests H^i for cost class i.

- If the guesses were correct, $deg_{H^i}(v) \leq B^i_v+1$.
- Return $T= \cup_i H^i$

Big Picture


G, B_v,c
Oracle
G, F^1,B^1_v
F over F
F over F
G, F^k,B^k_v
F over F
H^1
H^k
Combine
T

Two Caveats

- We only get $deg_T(v) = deg_{H^1}(v) + \ldots + deg_{H^k}(v)$

$$\leq (B^1_v+1) + \ldots + (B^k_v+1)$$

$$= B^1_v + \ldots + B^k_v + k$$

$$= B_v + k$$

- What is the Oracle?

Oracle: LP relaxation

- We use the LP solution.

$$\min \sum_e c_e x_e$$

s.t.

$$\sum_{e \in \delta(v)} x_e \leq B_v \qquad \forall v \in V$$

$$x \in SP(G)$$

- SP(G) is the convex hull of spanning trees of G.

- $c(x^*) > c(MST)$ then the bounds are infeasible for any MST
- Instead of facing infeasibility at each FoF problem, we decide once.

Algorithm

- Solve LP relaxation to obtain optimal LP solution x^*.
- (Check Feasibility) If $c(x^*)>c(MST)$, then declare the bounds infeasible
- (Divide Bounds) let $B^i_v = \lceil \sum_{e\in \delta(v) \text{ and } cost(e)=i} x_e \rceil$
- (Solve Subproblems) Use Forest over Forest algorithm to obtain F^i with bounds at most B^i_v+1.
- Return $T=\cup_i F^i$

Algorithm

- Solve LP relaxation to obtain optimal LP solution x^*.
- (Check Feasibility) If $c(x^*) > c(MST)$, then declare the bounds infeasible
- (Divide Bounds) let $B^i_v = \lceil \sum_{e \in \delta(v) \text{ and } cost(e)=i} x_e \rceil$
- (Solve Subproblems) Use Forest over Forest algorithm to obtain F^i with bounds at most B^i_v+1.
- Return $T = \cup_i F^i$

Need to make sure algorithm does not return a witness

LP is stronger

- Lemma: If any of the forest-over-forests problem with degree bounds given by the LP solution returns a witness showing infeasibility, then the LP has a value more than c(MST).
- Proof by contradiction: Let x^* denote the optimum LP solution.

$$\min \sum_e c_e x_e$$
s.t.
$$\sum_{e\in \delta(v)} x_e \le B_v \ \forall v\in V$$
$$x \in SP(G)$$

$$x^*=\sum_j \lambda_j T_j,\ \ \sum_j \lambda_j=1,\ \lambda_j > 0\ \forall\ j$$

Claim: $cx^*=c_{MST} \Rightarrow$ each T_j is a MST

Proof: $cx^* = \sum_j \lambda_j\, c(T_j) \ge \sum_j \lambda_j\, c_{MST} = c_{MST} \sum_j \lambda_j = c_{MST}$

Hence, each $c(T_j) \ge c_{MST}$ must hold at equality.

LP is stronger: Proof Contd

Let the i^{th} forest-over-forest problem be infeasible.

Let $F_j \subset T_j$ be the i^{th} forest-over-forest solution. Each F_j is exactly the cost i edges of T_j.

Claim: Let $y=\sum_j \lambda^j F_j$. Then $deg_v(y) \leq B^i_v \; \forall v \in V$

Proof: y is the exactly the cost i edges of x*.

There is a convex combination of forest-over-forests which satisfies the degree bounds.

Let W be the witness for i^{th} forest-over-forest problem.

Then $\sum_{w\in W} deg_{F_j}(w) \geq \sum_{w\in W} B^i_w + 1$ for each j.

Hence, for the convex combination y, $\sum_{w\in W} deg_y(w) \geq \sum_{w\in W} B^i_w + 1$

Contradiction.

Two Caveats

- What is the Oracle : Linear Program
- We still get $deg_T(v) = deg_{H^1}(v) + \dots + deg_{H^k}(v)$

$$\leq (B^1_v+1) + \dots + (B^k_v+1)$$

$$= B^1_v + \dots + B^k_v + k$$

Cost Class Error

$$\leq B_v + k-1+k = B_v+2k-1$$

Rounding Error

Strengthening of FR

- Theorem: Given a graph $G=(V,E)$, degree bounds B_v for each vertex v, $\exists$ polynomial time algorithm that returns a Witness set W and tree T such that

1. $W \neq \phi$ (Infeasible, as earlier…)
2. $W = \phi$

(Solution)

$deg_T(v) \leq B_v+1$ for each v

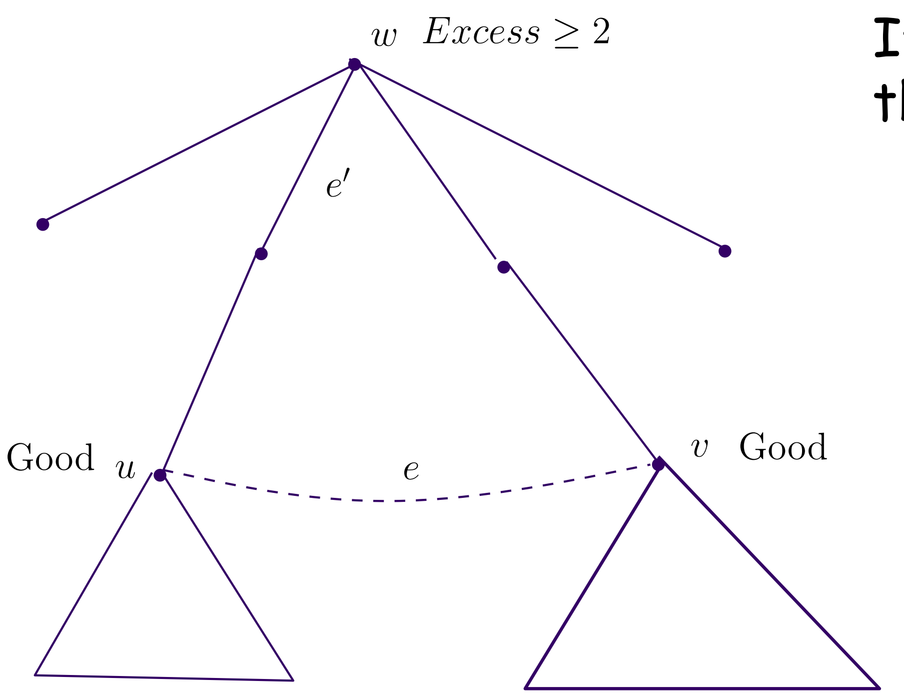
(Strong Solution)

For each $u \in V$, there exists a tree T_u such that $deg_{T_u}(u) \leq B_u$ and $\forall v \neq u$: $deg_{T_u}(v) \leq B_v+1$.

FR Algorithm

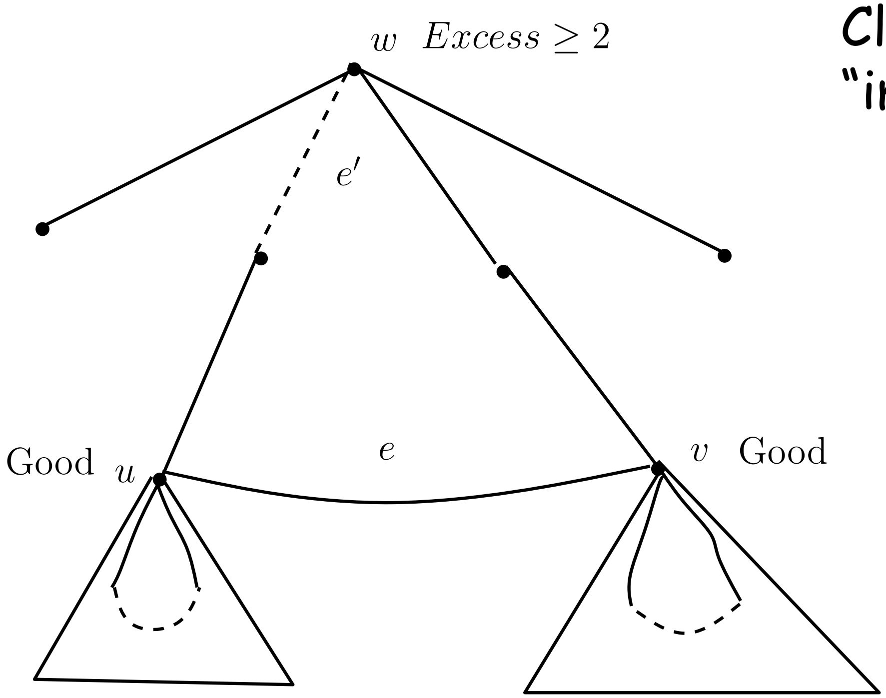
1. Initialize with any tree T
2. Define Ugly:={v| $deg_T(v) \geq B_v+2$}, Bad:={v| $deg_T(v)=B_v+1$} Good:={v|$deg_T(v) \leq B_v$}. If Ugly(T)=ϕ then return T
3. While there exists $e=(u,v) \in E \setminus T$ such that $u,v \in$ Good
   - mark all vertices in the cycle in $T \cup e$ as good.
4. If some Ugly vertex w is marked good, swap e for an edge incident at w and recursively improve u and v. Return to Step 2.
5. Return W=Ugly $\cup$ Bad

FR algorithm



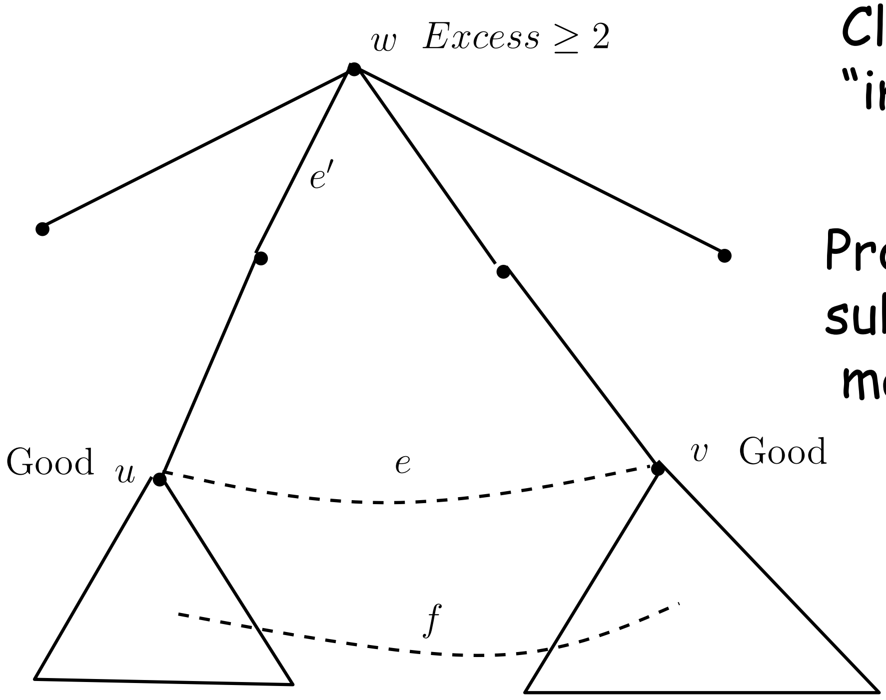
If $deg_T(u) \leq B_u$ and $deg_T(u) \leq B_v$ then swap e and e'.

FR algorithm



Claim: Both u and v can be "improved" in their own subtrees.

FR algorithm



Claim: Both u and v can be "improved" in their own subtrees.

Proof: If there exists a edge f across subtrees then w would have been marked good earlier!

FR algorithm

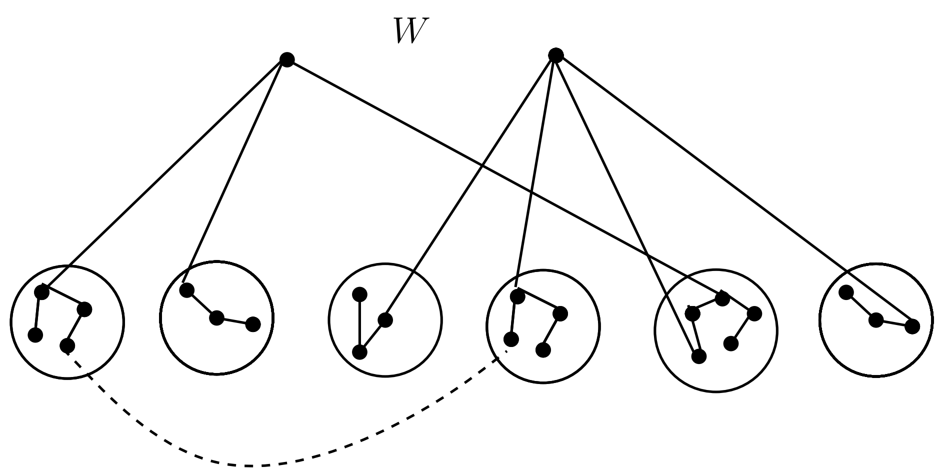
Claim: If the algorithm returns a witness $W = \text{Bad} \cup \text{Ugly}$ then the degree bounds are infeasible.

Proof: Consider the components of $T \backslash W$. We claim that components of $T \backslash W$ are also components of $G \backslash W$.

$deg_T(w) \geq B_w + 1$ for each $w \in$ Bad and $deg_T(w) \geq B_w + 2$ for each $w \in$ Ugly

$\Rightarrow |C| \geq \sum_{w \in W} B_w + |W| + 1 - 2(|W|-1) \geq \sum_{w \in W} B_w - |W| + 3.$

$\Rightarrow \sum_{w \in W} deg_{T'}(w) \geq |C| + |W| - 1 \geq (\sum_{w \in W} B_w) + 2$ for any tree T'



Strengthened FR

1. Initialize with any tree T
2. Define Ugly:={v| $deg_T(v) \geq B_v+2$}, Bad:={v| $deg_T(v)=B_v+1$} Good:={v| $deg_T(v) \leq B_v$}. If (Ugly $\cup$ Bad) $=\phi$ then return T
3. While there exists $e=(u,v) \in E \setminus T$ such that $u,v \in$ Good
   - mark all vertices in the cycle in $T \cup e$ as good.
4. If some Ugly vertex w is marked good, swap e for an edge incident at w and recursively improve u and v. Return to Step 2.
5. Return W=Ugly $\cup$ Bad

Strengthening of FR

- Theorem: Given a graph $G=(V,E)$, degree bounds B_v for each vertex v, $\exists$ polynomial time algorithm that returns a Witness set W and tree T such that

1. $W \neq \phi$ (Infeasible, as earlier…)
2. $W = \phi$

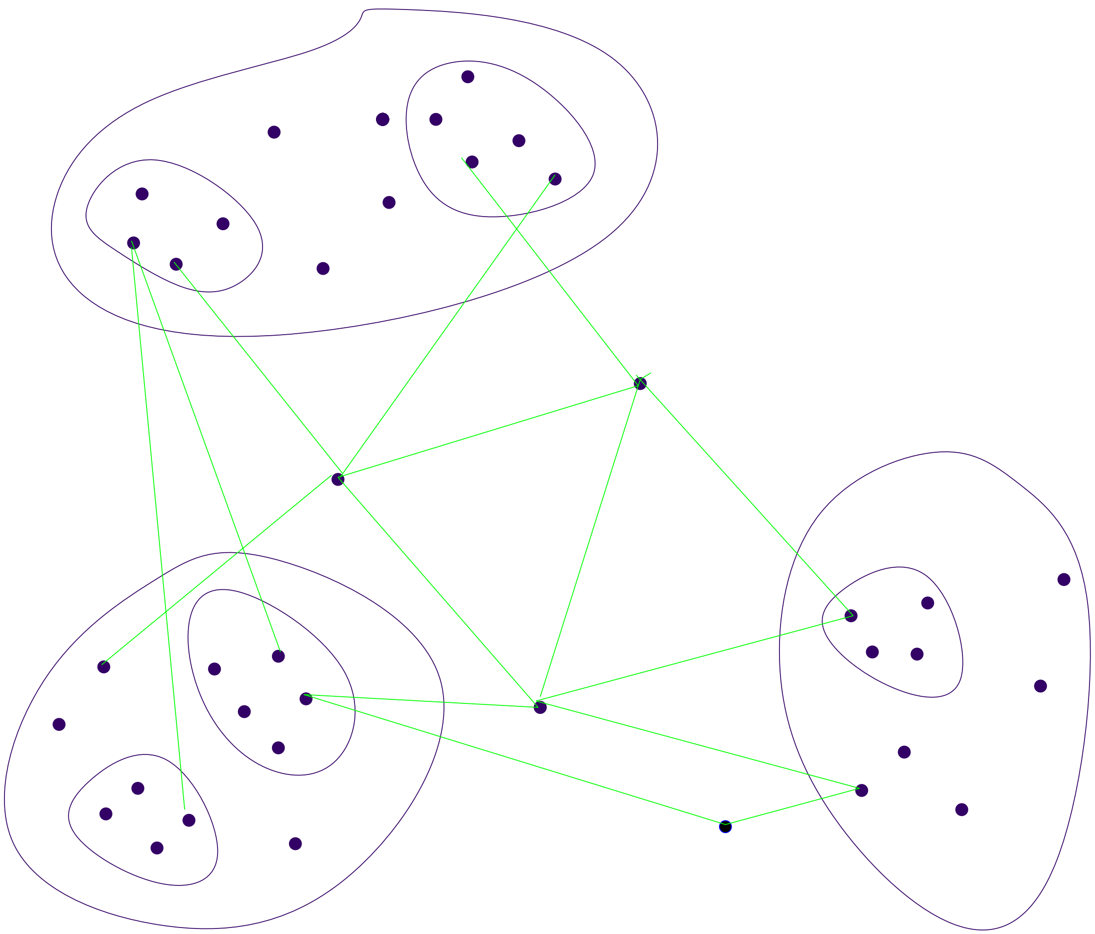
(Solution)

$deg_T(v) \leq B_v+1$ for each v

(Strong Solution)

For each $u \in V$, there exists a tree T_u such that $deg_{T_u}(u) \leq B_u$ and $\forall v \neq u$: $deg_{T_u}(v) \leq B_v+1$.

Forest over Forest Problem



Strengthening of Forest-over-Forest

- Theorem: A polynomial time algorithm returns a Witness set W and tree T such that

1. $W \neq \phi$ (Infeasible, as earlier…)
2. $W = \phi$

(Solution)

$deg_T(v) \leq B_v+1$ for each $v \in V$

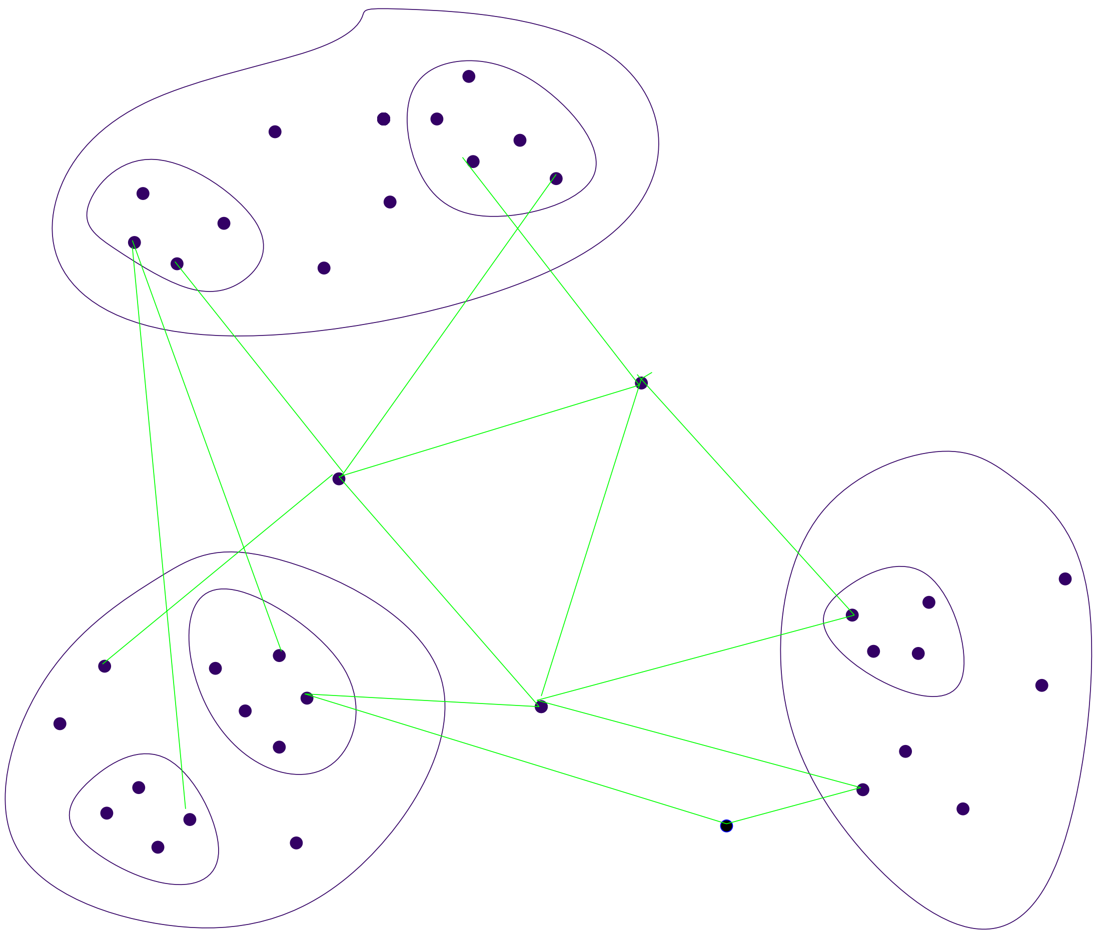
(Strong Solution)

In each "supernode", there is at most 1 vertex at B_v+1 and one can choose a supernode such that every vertex satisfies the degree bound in that supernode.

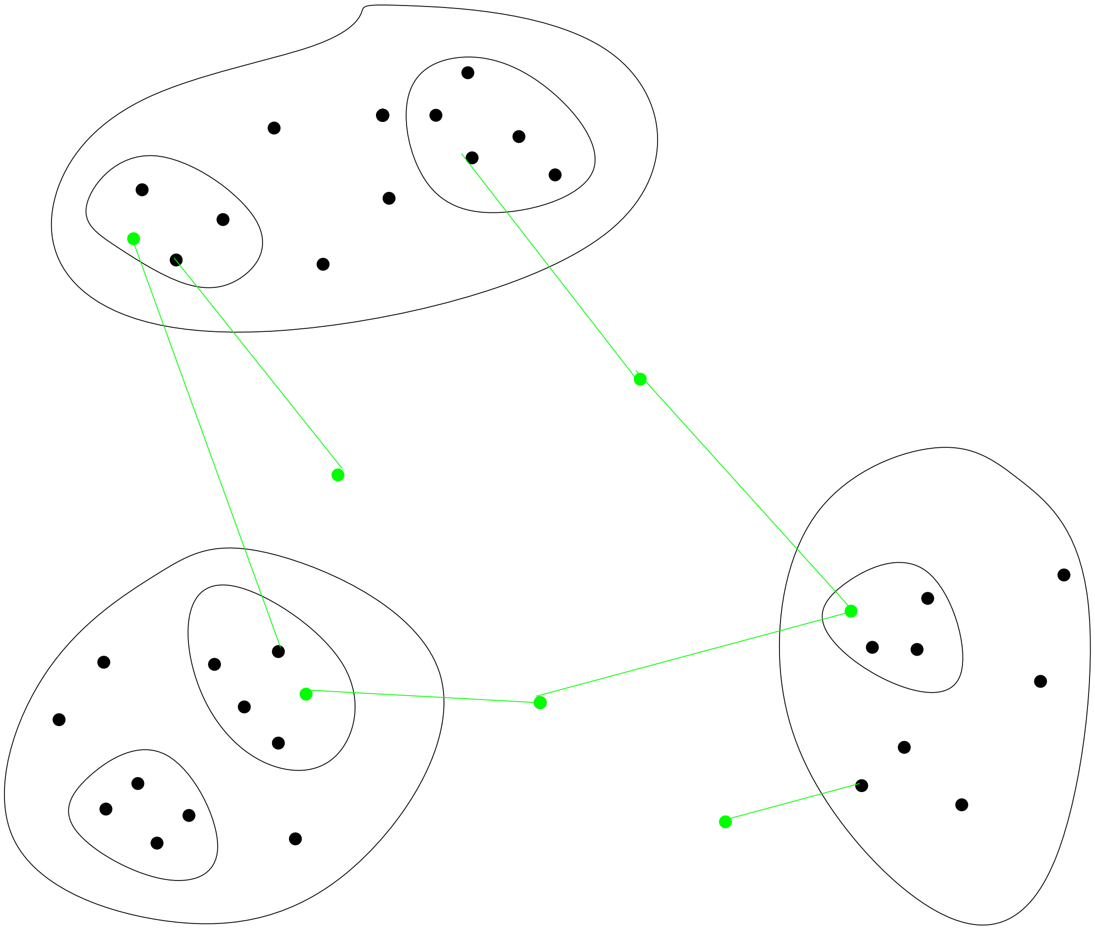
Delegate and Conquer

- Solve LP relaxation to obtain optimal LP solution x*.
- (Check Feasibility) If c(x*)>c(MST), then declare the bounds infeasible
- (Divide Bounds) let $B^i_v = \lceil \sum_{e \in \delta(v) \text{ and } cost(e)=i} x_e \rceil$
- (Solve Subproblems) In a top down manner, solve the FoF problem using the Strong guarantee to ensure that the degree of any vertex exceeds its bound in at most 1 cost class.
- Return $T = \cup_i F^i$

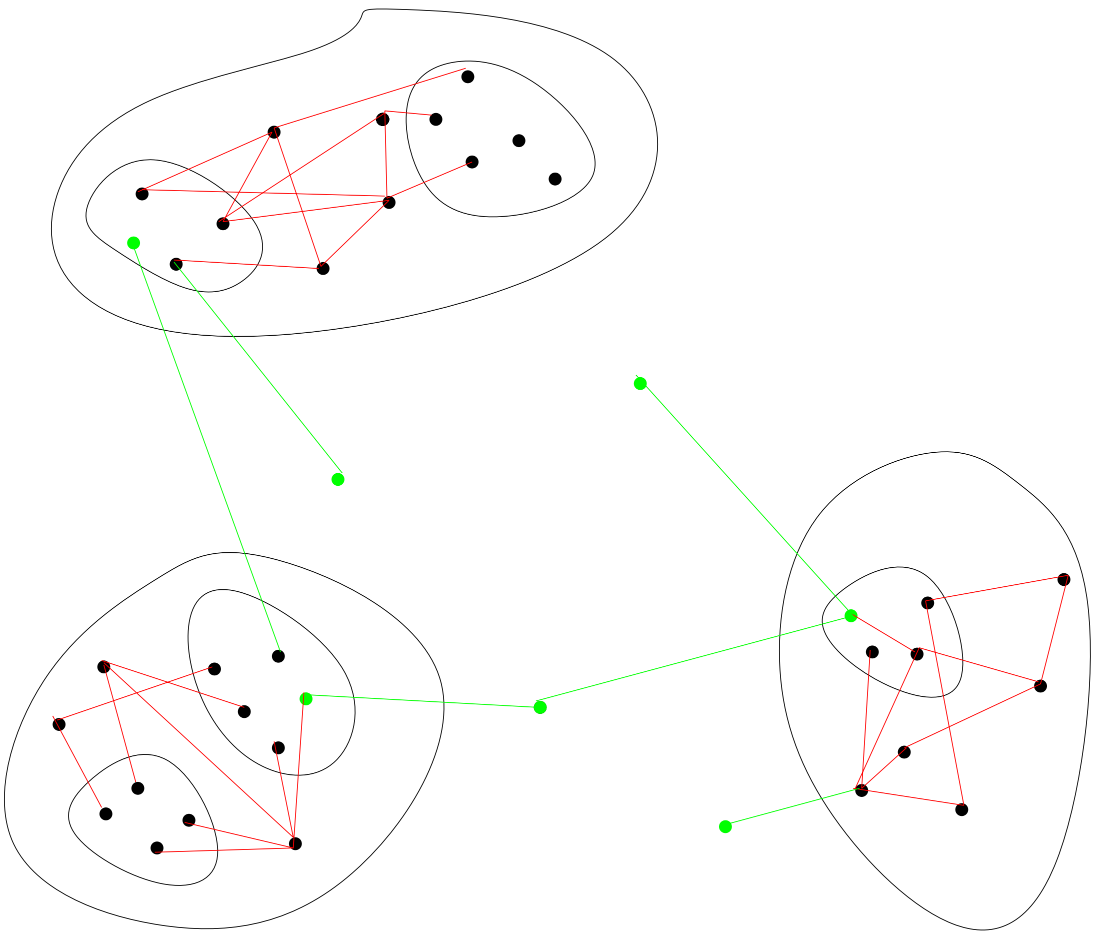
Delegate and Conquer



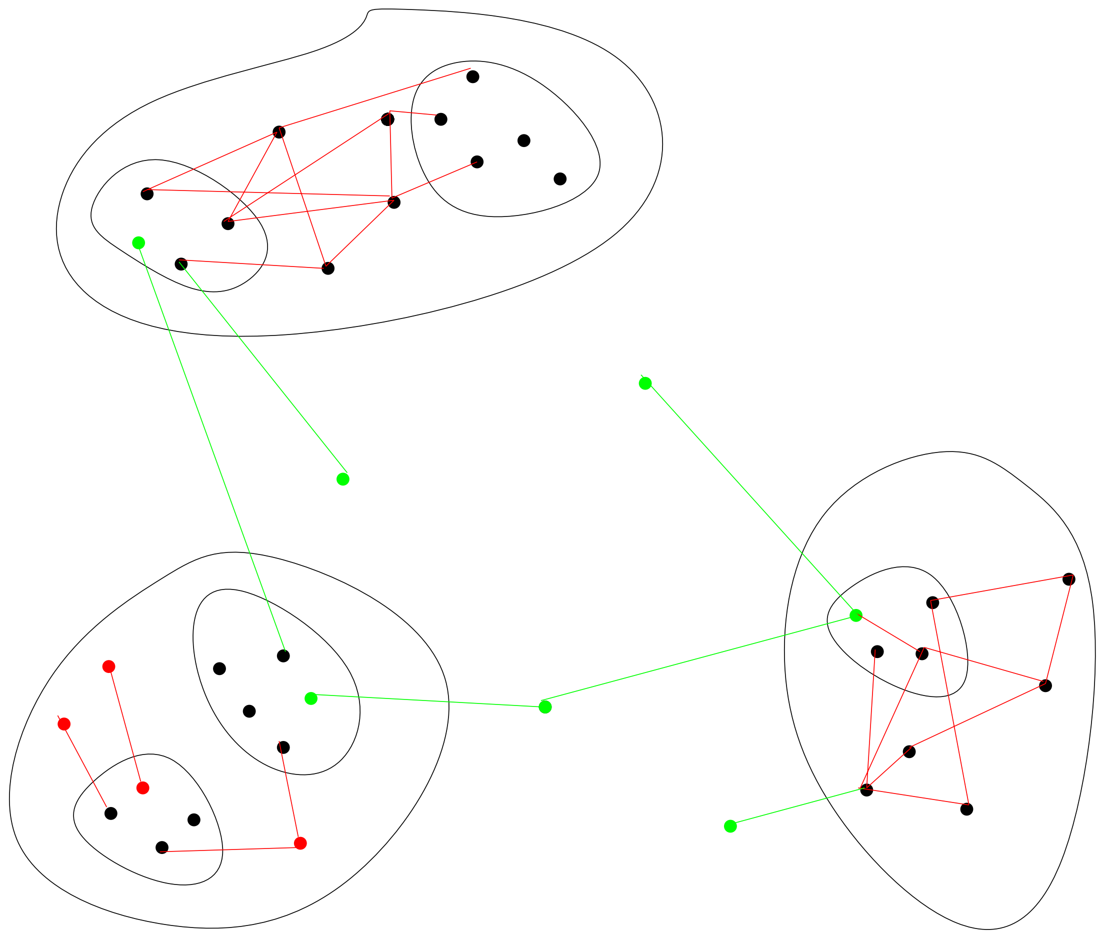
Delegate and Conquer



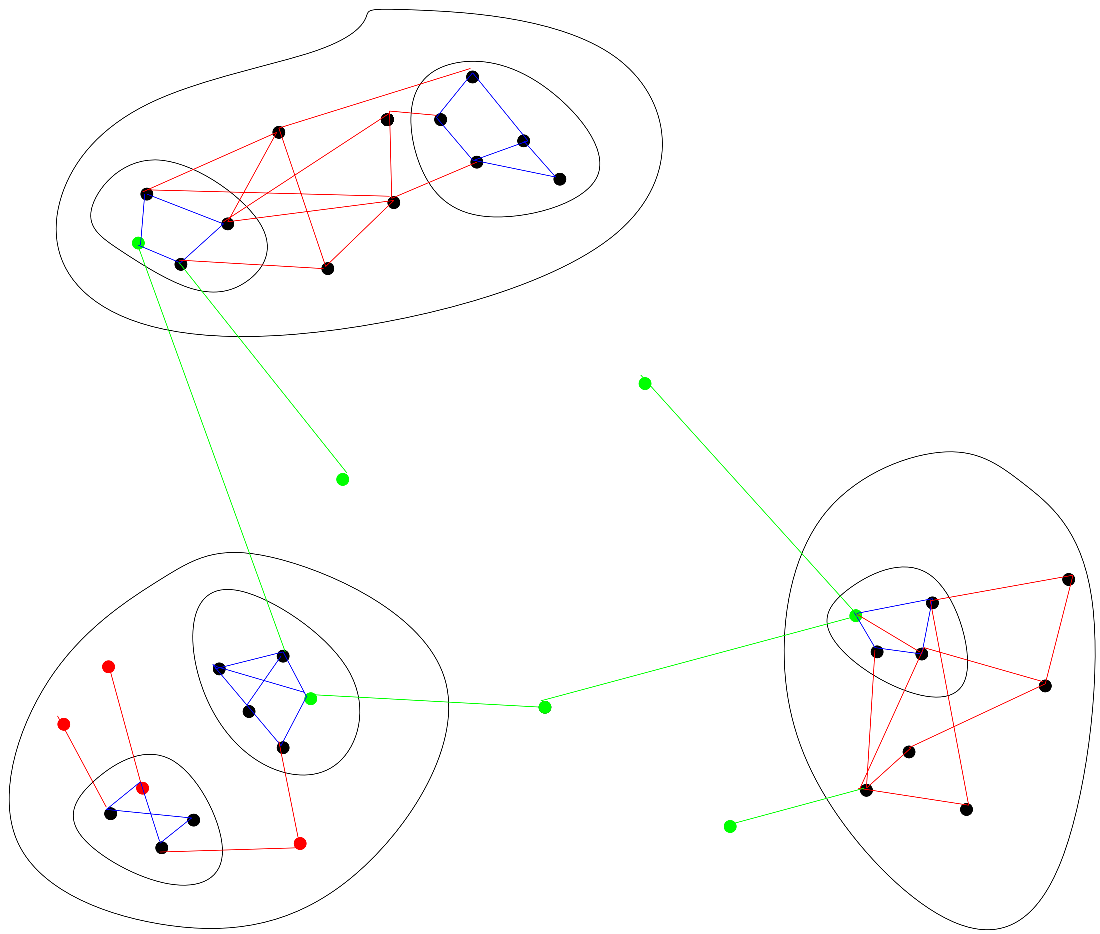
Delegate and Conquer



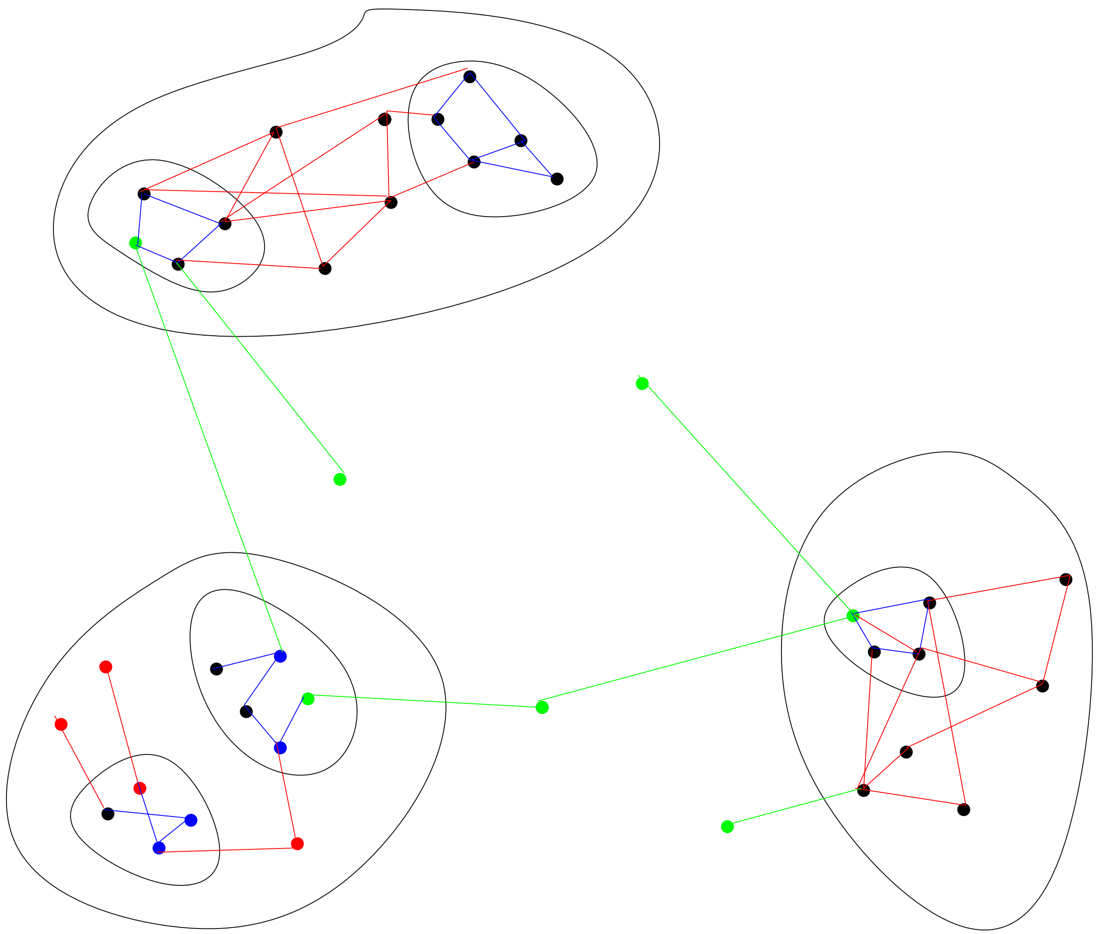
Delegate and Conquer



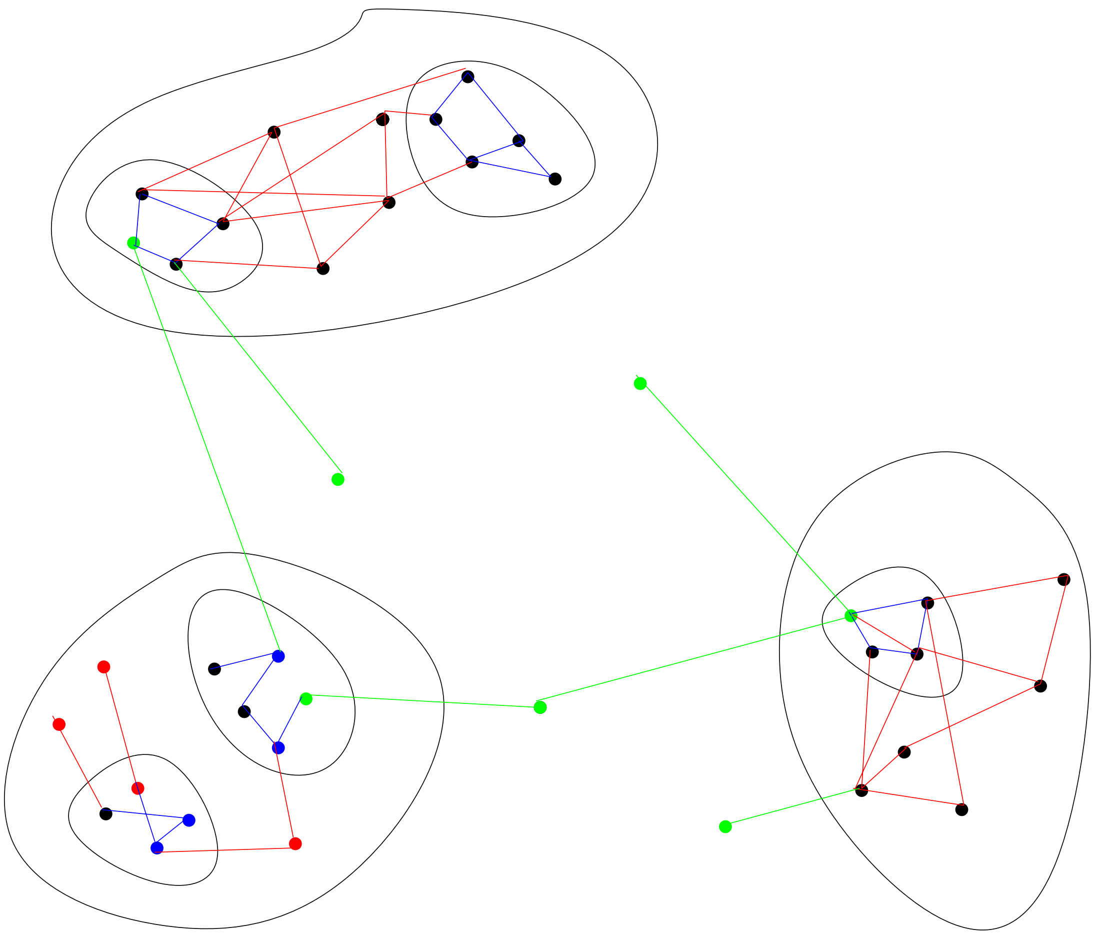
Delegate and Conquer



Delegate and Conquer



Delegate and Conquer



QED

BDMST problem

- Given an undirected graph G, cost function c, a bound B on maximum degree
  1. Return the cheapest tree which satisfies the degree bounds, or
  2. Show the degree bounds are infeasible for any tree of G

Konemann and Ravi '00,'02 gave a general procedure using Lagrangian relaxation for obtaining bicriteria approximation for BDMST problem. Using Fischer's algorithm they return a tree of cost $O(c_{opt})$ and degree $O(\Delta^*+\log n)$.

Using similar ideas, Chaudhuri et al'05, give a tree of cost at most c_{opt} and degree $O(\Delta^*+\log n)$

Open Problems

- Obtain a MST of max degree OPT+1 similar to unweighted case?
- Recently, Goemans announced an OPT+2 algorithm.

Questions?

A Swap Theorem

- Theorem: Given any T, there exists a sequence of trees

$$T=T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_l$$

such that $deg(T_i) \leq deg(T_{i-1})$ and $deg(T_l)=\Delta^*$.

and $\rightarrow$ is a single edge swap of equicost edges.

Proof: We will fix T_{opt} and make progress towards T_{opt} by edge swaps.