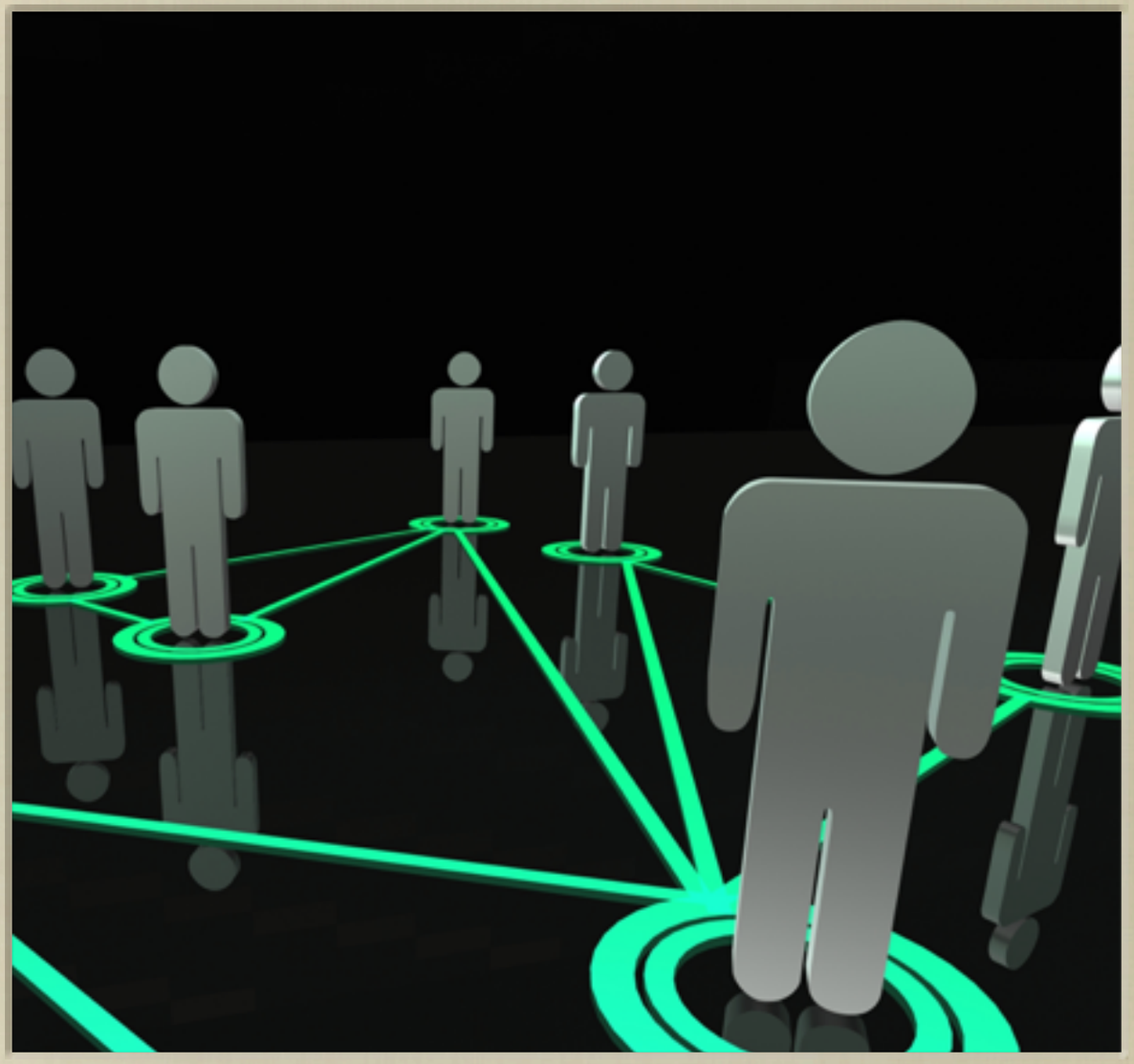
Ability of Collaborative Social Networks to Complete Directed Acyclic Task

Jize Zhang, Rallye Shen, Mäneka Puligandla & Kelly Zaccheo

Smith College
Advisor: Ben Baumer

Overview

- $G = (G_S, G_T, R)$
  - composite graph
- $G_S = (V_S, E_S)$
  - undirected social network ;
  - f assigns V_S (expertise)
- $G_t = (V_t, E_t)$
  - directed acyclic graph of tasks:
  - g assigns V_t (task difficulty)
- R: assignment of researchers to tasks



Collaboration

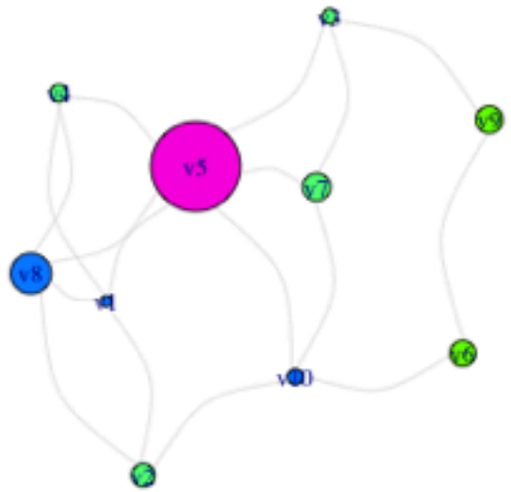
- In theory collaboration should increase productivity
- Desirable Properties:
  - Monotonic, Submodular

$$c(V,E) = \left(1 + \frac{|E|}{\binom{|V|}{2}}\right) \cdot \sum_{v \in V} f(v)$$

$$c(V,E) = \left[\log\left(\sum_{v \in v'} \text{expertise } (v) + 1\right)\right] \cdot \left(1 + \frac{\log(|E|+1)}{\log(\binom{n}{2}+1)}\right)$$

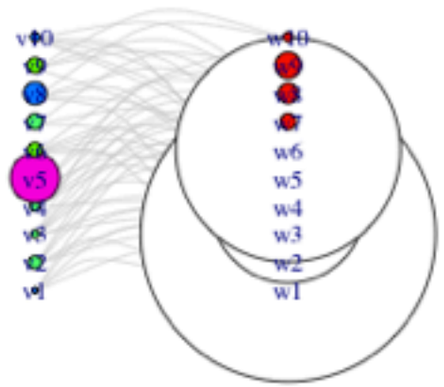
$$c(V,E) = \log \sum_{v \in v'} [\text{ expertise } (v) \times ((\text{ deg}(v) \text{ in } G') + 1)]$$

G1 (Social Network)
Erdos renyi (gnp) graph



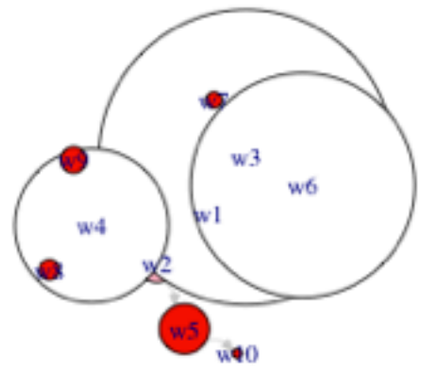
|V| = 10 , |E| = 17 , Edge Density = 0.378
Collective Expertise = 35.26

R (Assignments)
Bipartite Mapping



|E| = 50 , Edge Density = 0.5
Alls Tasks are Not Completed

G2 (Task Graph)
Tree



|V| = 10 , |E| = 9 , Diameter = 3
5 complete, 2 solvable, 3 remaining
Max Difficulty = 46.58

Optimization

- Goals:
  - For each researcher: assume no limit on number of assignments
  - Minimize total number of assignments to complete each task

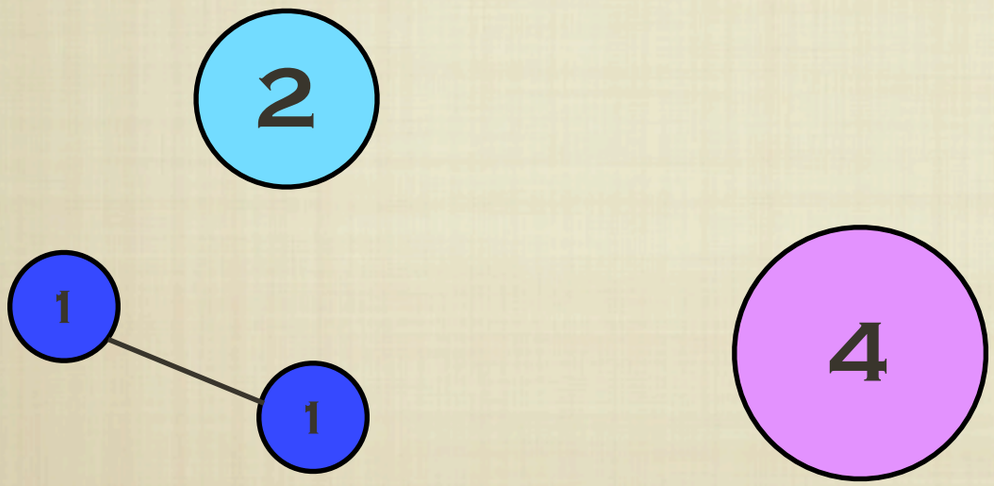
Algorithms

Greedy

- Idea:
  - Assign best researcher to all tasks
- If all tasks are not complete...
  - Assign next best researcher to the remaining incomplete tasks
  - Problem of finding next 'best' researcher
    - Compute collaboration function of best researcher with all remaining researchers and choose the researchers that maximizes the collaboration function

Counter-Examples

- Greedy is not optimal
- At best 3/2 approximation
- Counterexample: for any collaboration function



Demo

- http://rstudio.smith.edu:3838/algorithms/

Future Work

- Compare collaboration functions
- Find algorithms that work better than greedy
- More realistic constraints
  - Ex: Cap number of tasks