Algorithms for Team Formation

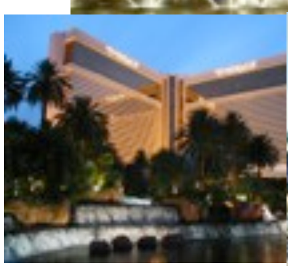
Evimaria Terzi (Boston University)

BOSTON UNIVERSITY

Team-formation problems

- Given a task and a set of experts (organized in a network) find the subset of experts that can effectively perform the task
- Task: set of required skills and potentially a budget
- Expert: has a set of skills and potentially a price
- Network: represents strength of relationships

evimaria@cs.bu.edu



Security expert

Electronics expert

Insider

Mechanic

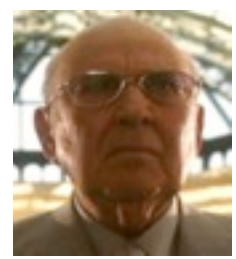
Pick-pocket thief

Organizer

Co-organizer

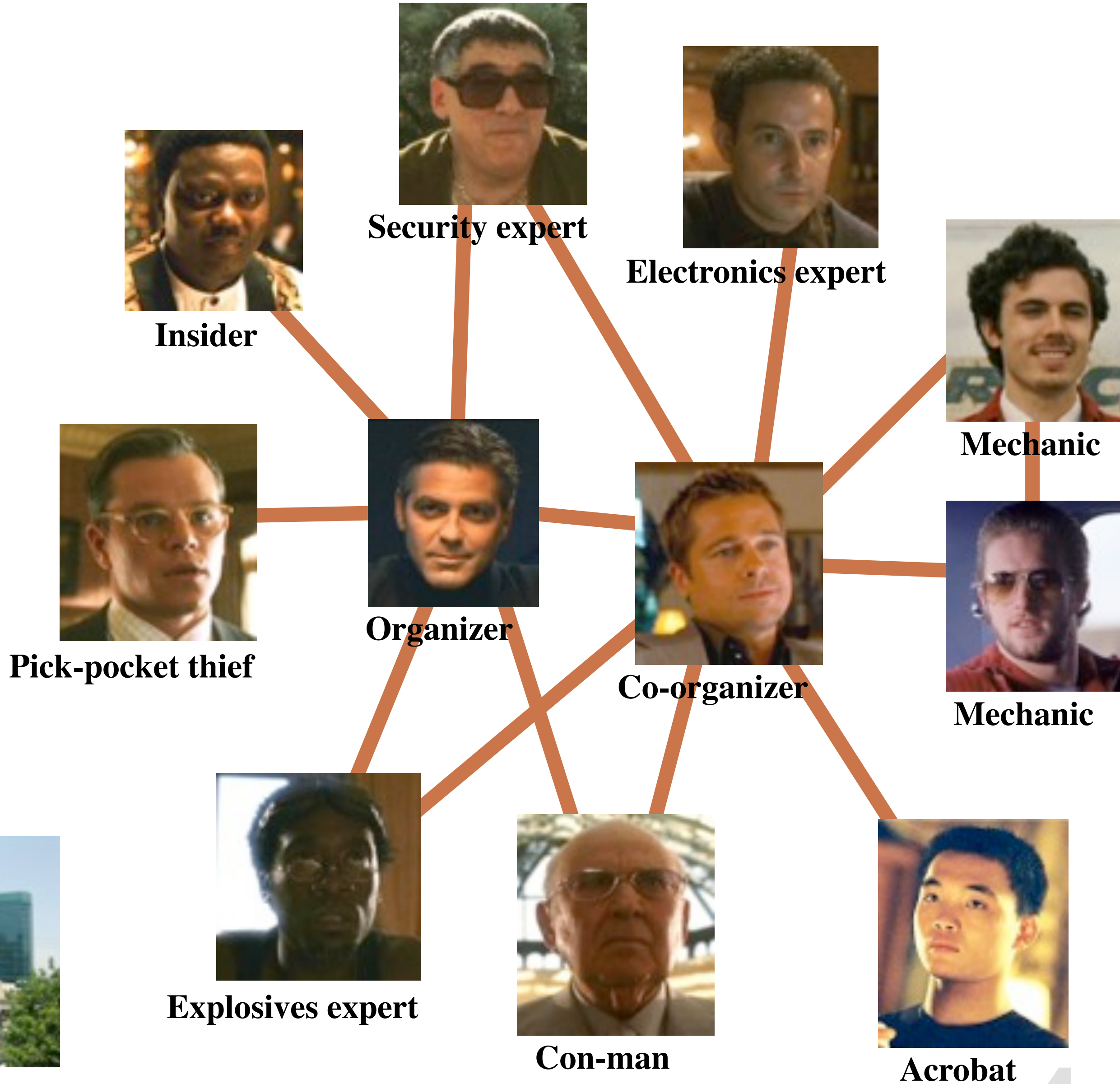
Mechanic

Explosives expert



Con-man

Acrobat



BOSTON UNIVERSITY

Applications

- Collaboration networks (e.g., scientists, actors)
- Organizational structure of companies
- LinkedIn, Odesk, Elance
- Geographical (map) of experts

evimaria@cs.bu.edu

Roadmap

- Background
- Team formation and cluster hires
- Team formation in the presence of a social network
- Inferring abilities of experts
- Team formation in educational settings

evimaria@cs.bu.edu

Roadmap

- **Background**
- Team formation and cluster hires
- Team formation in the presence of a social network
- Inferring abilities of experts
- Team formation in educational settings

evimaria@cs.bu.edu

The SetCover problem

- Setting:
  - Universe of N elements $U = \{U_1,\ldots,U_N\}$
  - A set of n sets $S = \{s_1,\ldots,s_n\}$
  - Find a collection C of sets in S (C subset of S) such that $\cup_{c \in C} c$ contains many elements from U
- Example:
  - U: set of skills required for a task
  - s_i: set of skills of expert i
  - Find a collection of experts that cover the required skills for the task

evimaria@cs.bu.edu

The SetCover problem

- Universe of N elements $U = \{U_1,\ldots,U_N\}$
- A set of n sets $S = \{s_1,\ldots,s_n\}$ such that $U_i s_i = U$

- **Question:** Find the smallest number of sets from S to form collection C (C subset of S) such that $U_{c \in C} c = U$

- The set-cover problem is **NP-hard** (what does this mean?)

evimaria@cs.bu.edu

Trivial algorithm

- Try all subcollections of **S**
- Select the smallest one that covers all the elements in **U**
- The running time of the trivial algorithm is **$O(2^{|S|}|U|)$**
- This is way too slow

evimaria@cs.bu.edu

Greedy algorithm for set cover

- Select first the largest-cardinality set **s** from **S**
- Remove the elements from **s** from **U**
- Recompute the sizes of the remaining sets in **S**
- Go back to the first step

evimaria@cs.bu.edu

As an algorithm

- $X = U$
- $C = \{\}$
- **while** X is not empty **do**
  - For all $s \epsilon S$ let $a_s = |s \text{ intersection } X|$
  - Let s be such that a_s is maximal
  - $C = C \cup \{s\}$
  - $X = X \setminus s$

evimaria@cs.bu.edu

How can this go wrong?

- No global consideration of how good or bad a selected set is going to be

evimaria@cs.bu.edu

How good is the greedy algorithm?

- Consider a minimization problem
  - In our case we want to minimize the **cardinality** of set C

- Consider an instance I, and cost $a^*(I)$ of the optimal solution
  - $a^*(I)$: is the minimum number of sets in C that cover all elements in U

- Let a(I) be the cost of the approximate solution
  - a(I): is the number of sets in C that are picked by the greedy algorithm

- An algorithm for a minimization problem has approximation factor F if for all instances I we have that

$$a(I) \leq F \times a^*(I)$$

- **Can we prove any approximation bounds for the greedy algorithm for set cover ?**

evimaria@cs.bu.edu

How good is the greedy algorithm?

- The greedy algorithm for set cover has approximation factor **$F = O(\log |s_{max}|)$**
- **Proof**: (From CLR "Introduction to Algorithms")

evimaria@cs.bu.edu

Roadmap

- Background
- **Team formation and cluster hires**
- Team formation in the presence of a social network
- Inferring abilities of experts
- Team formation in educational settings

evimaria@cs.bu.edu

What makes a team effective for a task?

- T = {algorithms, java, graphics, python}

Alice	Bob	Cynthia	David	Eleanor
{algorithms}	{python}	{graphics, java}	{graphics}	{graphics,java,python}

Coverage: For every required skill in T there is at least one team member that has it

evimaria@cs.bu.edu

Problem definition (SimpleTeam)

- Given a task and a set of individuals, find the **most efficient** subset (team) of individuals that can perform the given task.

- NP-hard (Set Cover Problem)

evimaria@cs.bu.edu

Setting [GLT'14]

- Experts (defining the set V, with $|V|=n$):
  - Every expert i is associated with **a set of skills** X_i
  - and **a price** p_i
- Tasks
  - Every task T is associated with a set of skills (T) **required** for performing the task

	Team Formation
Experts' skills	Known
Participation of experts in teams	Unknown

evimaria@cs.bu.edu

Expertise systems

- Two main components of a job market

Jobs

JAVA
Node.JS
90$ / hour

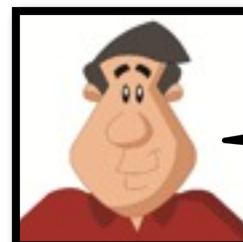
HTML
JAVA
33$ / hour

⋮

Node.JS
SQL
10$ / hour

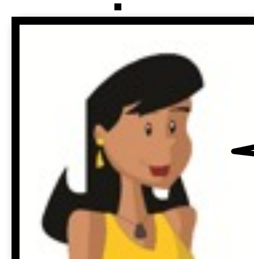
Workers

JAVA, C++, SQL
18$ / hour



JAVA, HTML
7$ / hour

⋮



HTML, Node.JS
40$ / hour

Expertise systems

- Two main components of a job market

Jobs

JAVA
Node.JS
90$ / hour

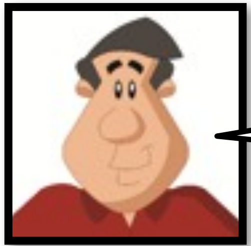
HTML
JAVA
33$ / hour

⋮

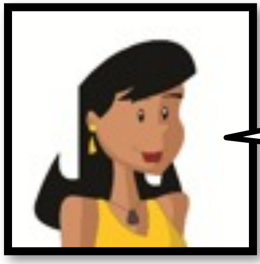
Node.JS
SQL
10$ / hour

Organizations Agencies

Workers



⋮



Expertise systems

- Two main components of a job market

Jobs

JAVA
Node.JS
90$ / hour

HTML
JAVA
33$ / hour

⋮

Node.JS
SQL
10$ / hour

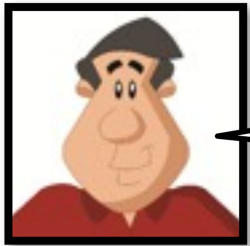
Organizations
Agencies

↓

Who to hire and which jobs to do?

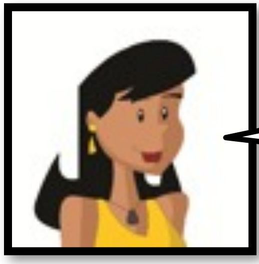
Workers

JAVA, C++, SQL
18$ / hour



JAVA, HTML
7$ / hour

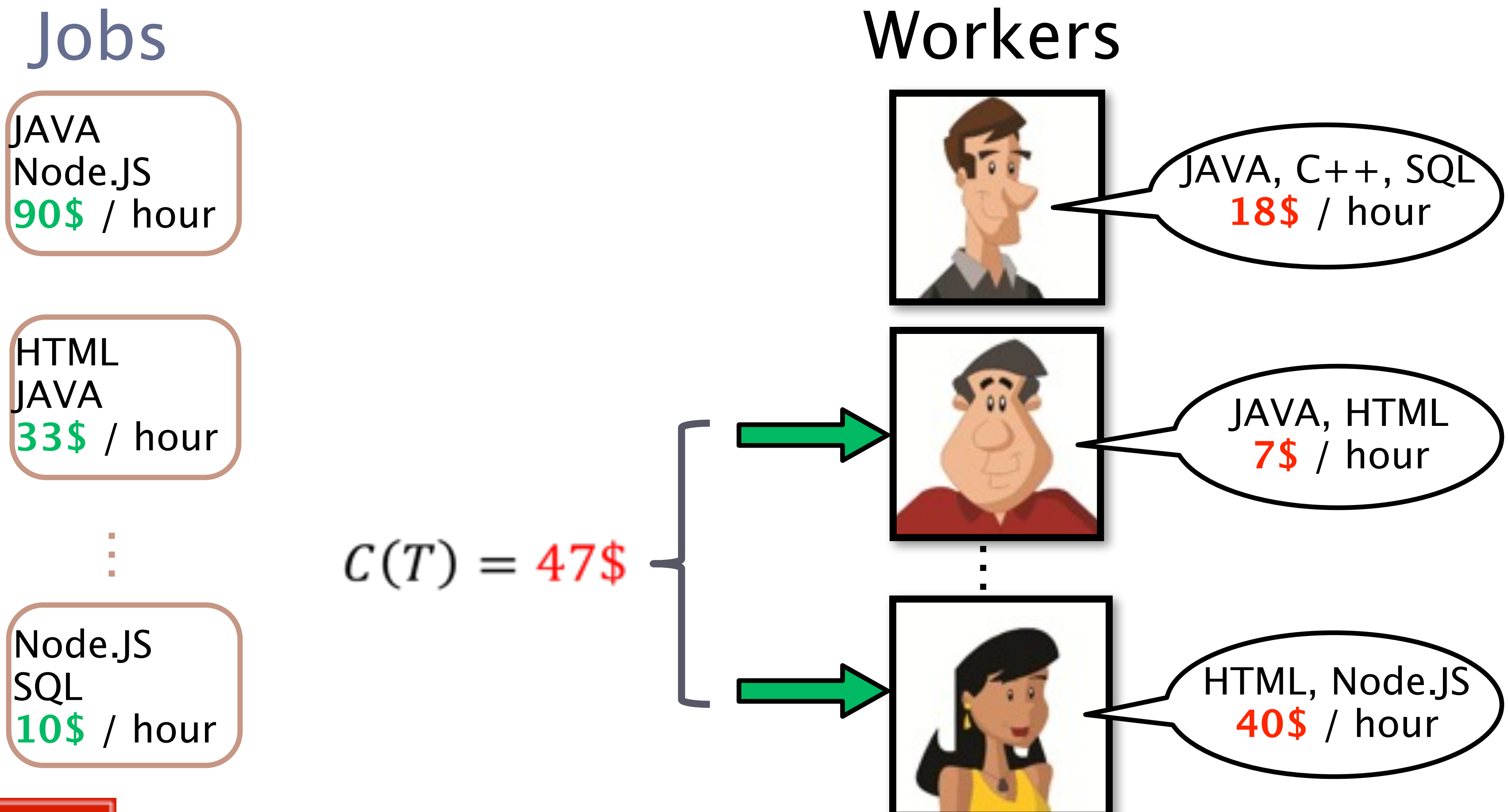
⋮



HTML, Node.JS
40$ / hour

Expertise systems

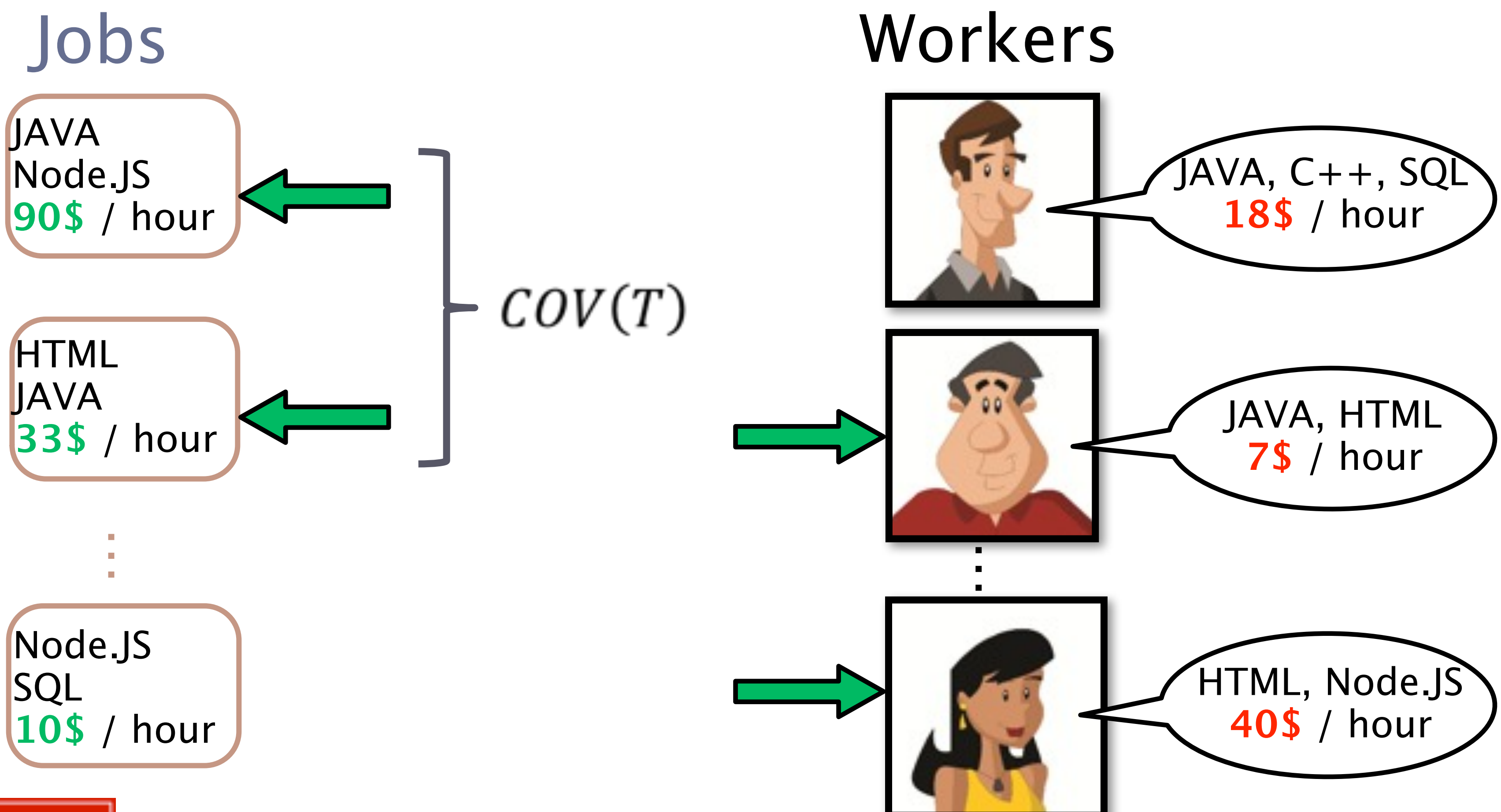
- Cost of hiring a team of experts



evimaria@cs.bu.edu

Expertise systems

- Jobs completed by a team of experts



Expertise systems

- Possible profit models? F($COV(T)$)

Jobs

JAVA
Node.JS
90$ / hour

HTML
JAVA
33$ / hour

⋮

Node.JS
SQL
10$ / hour

- Dollar-based profit model:
  - Value of all project covered by experts
  - Example: **90$ + 33$**
- Competition-based profit model:
  - Probabilistic model
    - $P(\text{Getting Job } J) = \frac{1}{\text{Freq. of the rarest skill in } J}$
    - Expected value of all covered projects
  - Example: **(90$ + 33$) / 10**
    - Assuming that only 10 people know JAVA!

The ClusterHire problem

- ClusterHire:
  - Given a budget B, hire a team of experts T such that
    - $C(T) <= B$
    - $F(COV(T))$ is maximized.
  - Complexity: NP-hard to solve and approximate.
    - Reduction from Set Cover to ClusterHire

The ClusterHire problem

- ClusterHire:
  - Given a budget B, hire a team of experts T such that
    - $C(T) <= B$
    - $F(COV(T))$ is maximized.
  - Complexity: NP-hard to solve and approximate.
    - Reduction from Set Cover to ClusterHire
- t-ClusterHire:
  - Each skill of a worker can be used in at most t projects
  - Complexity: NP-hard to evaluate the objective function
    - Reduction from Set Packing to t-ClusterHire

The ExpertGreedy algorithm

- Hires an expert in each iteration
- Expert with the best profit to cost ratio

$$\frac{F(\text{Cov}(\mathcal{T}^i \cup \{X\})) - F(\text{Cov}(\mathcal{T}^i))}{C(X)}$$

- Repeat until the budget is consumed

evimaria@cs.bu.edu

The ProjectGreedy algorithm

- Selects a job in each iteration
- Hire a (not "the") cost-effective experts for the job
  - This is SetCover: Use a greedy method to find a team
- Pick project with the best profit to cost ratio

$$\frac{F(\mathrm{Cov}(\mathcal{T}^i \cup \mathcal{X}_P)) - F(\mathrm{Cov}(\mathcal{T}^i))}{C(\mathcal{X}_P)}$$

- Repeat until the budget is consumed

evimaria@cs.bu.edu

The CliqueGreedy algorithm

- Similar to but faster than ProjectGreedy
- Examines cliques of compatible projects
  - Stand-alone ratio

$$R_i = \frac{F(P_i)}{C(P_i)}$$

  - Combined ratio (for pairs of projects)

$$R = \frac{F(P_1 \cup P_2)}{C(P_1 \cup P_2)}$$

  - Compatibility condition

$$R > (1+\alpha)R_i \text{ for both } i = 1, 2$$

- An edge exists between two projects if condition holds

evimaria@cs.bu.edu

The SmartRandom (baseline) algorithm

- Randomized version of ProjectGreedy
- Somewhat smart
  - Hires a cost-effective team for a project
  - Repeats until the budget is consumed

evimaria@cs.bu.edu

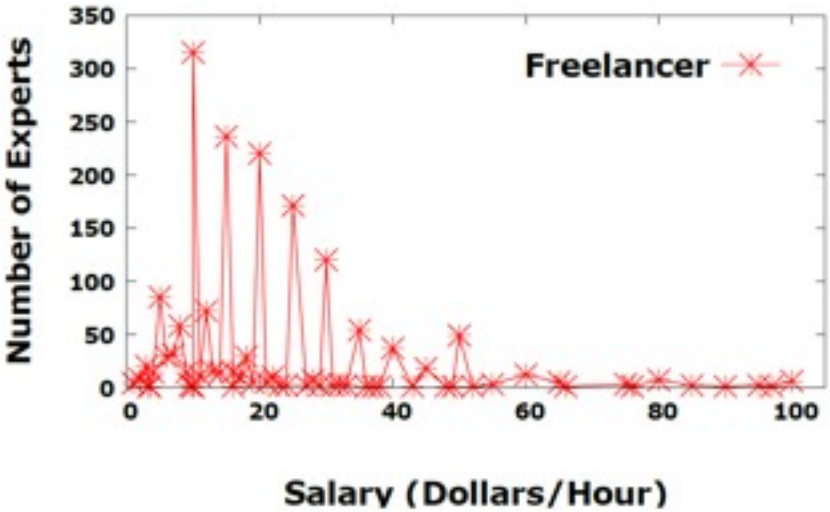
Real-world datasets

- freelancer.com
  - 1,763 experts
  - 721 projects
- guru.com
  - 6,473 experts
  - 1,764 projects

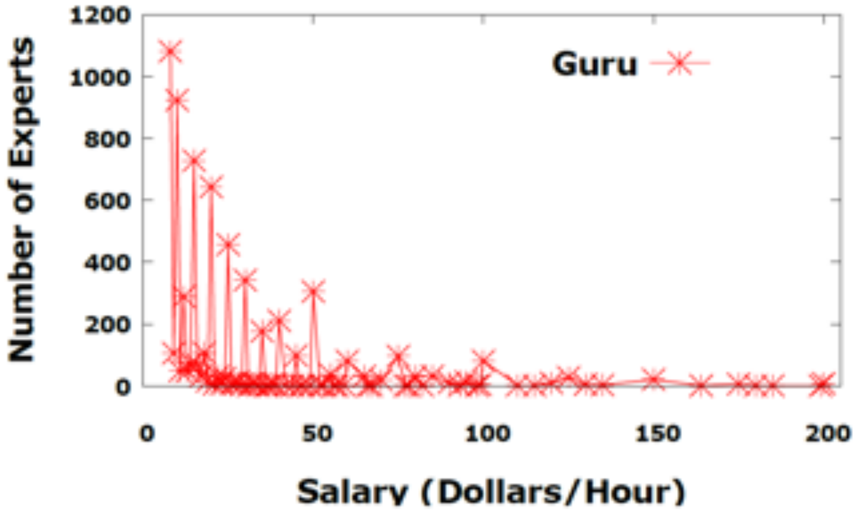
evimaria@cs.bu.edu

Workers data

- Freelancer

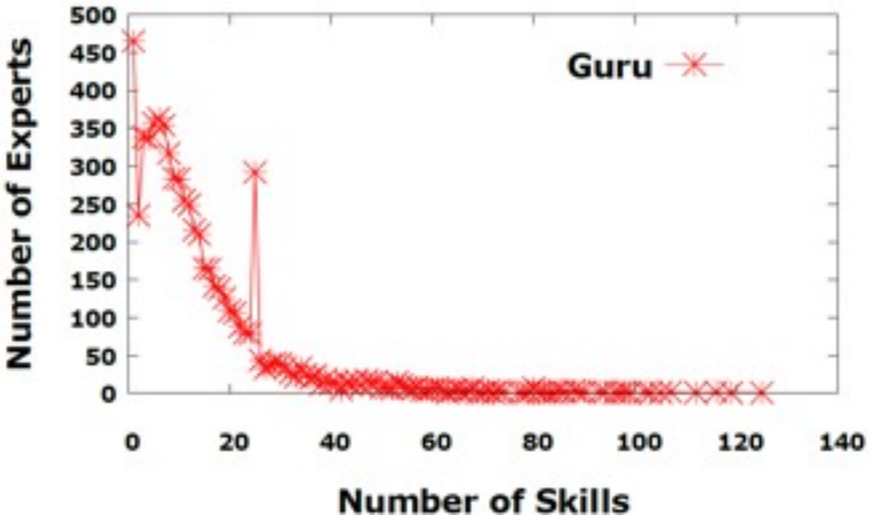


- Guru



Workers data

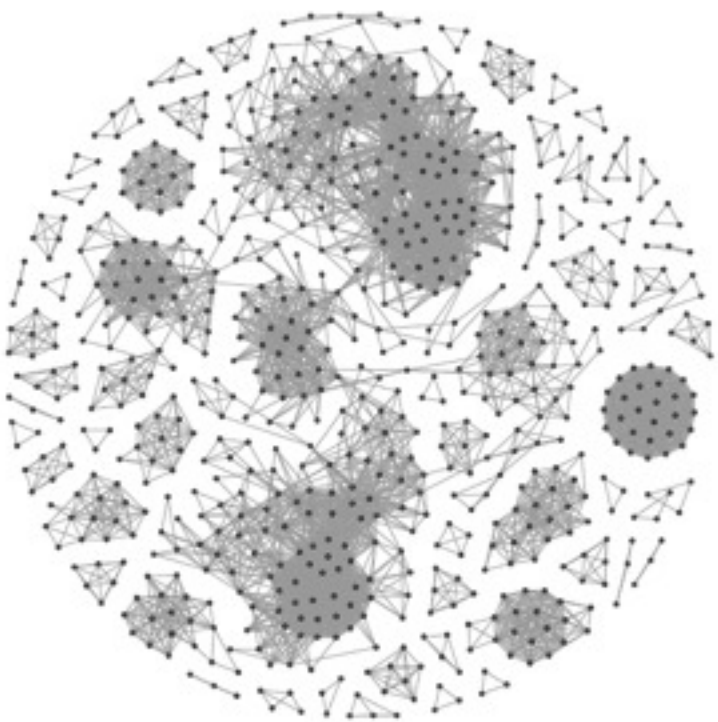
- Freelancer
- Guru



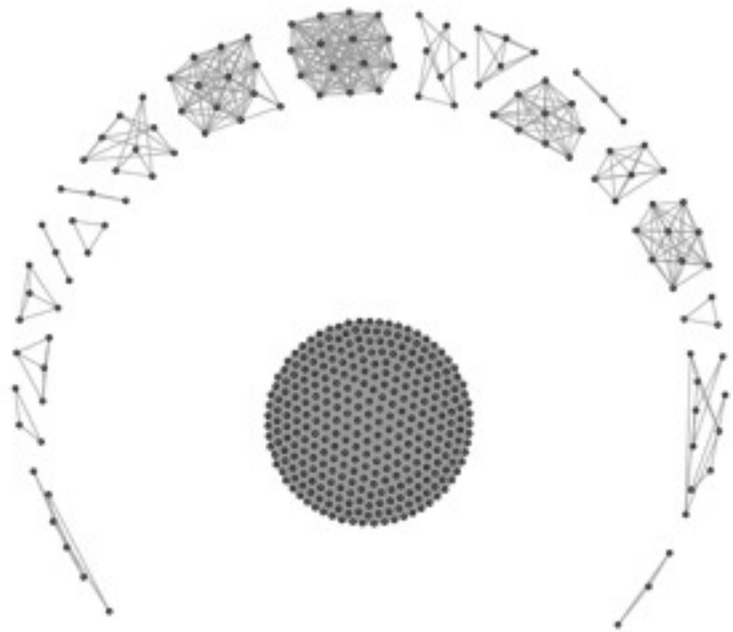
evimaria@cs.bu.edu

Workers data

- Freelancer



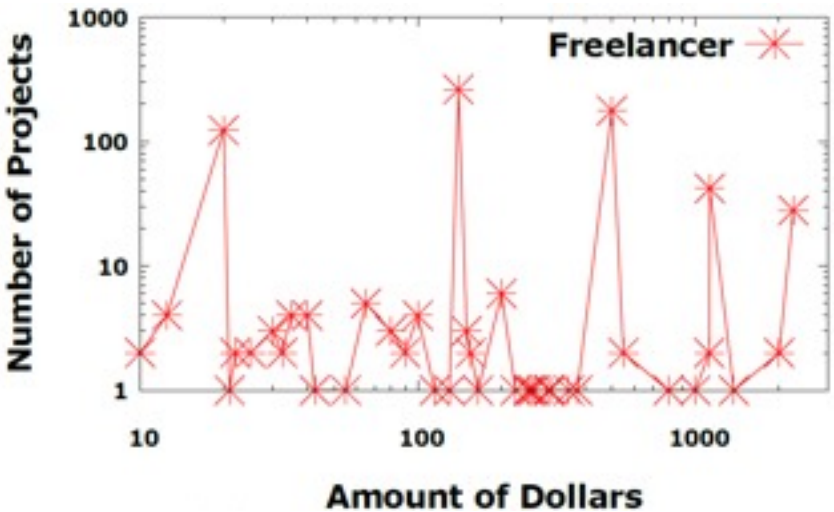
- Guru



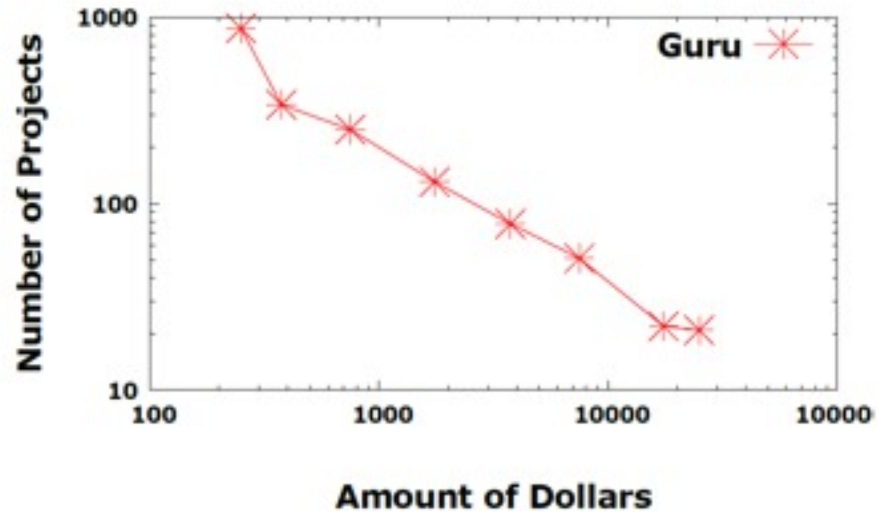
evimaria@cs.bu.edu

Projects data

- Freelancer

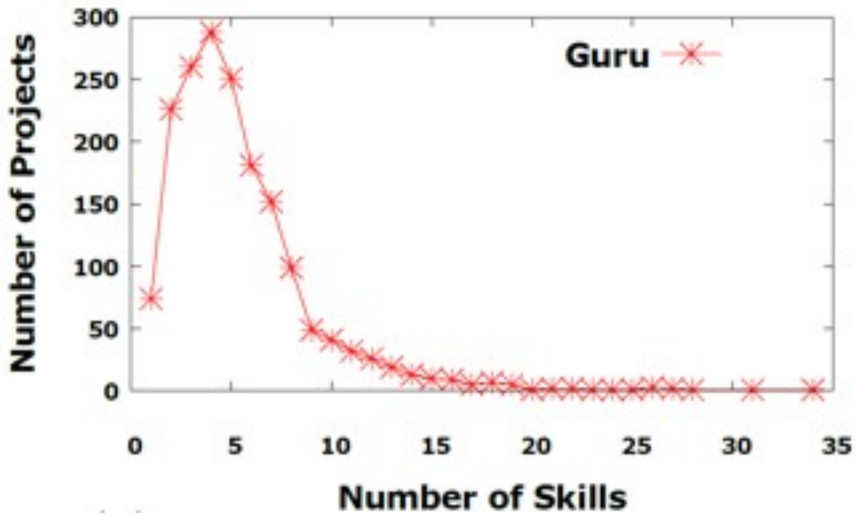


- Guru



Projects data

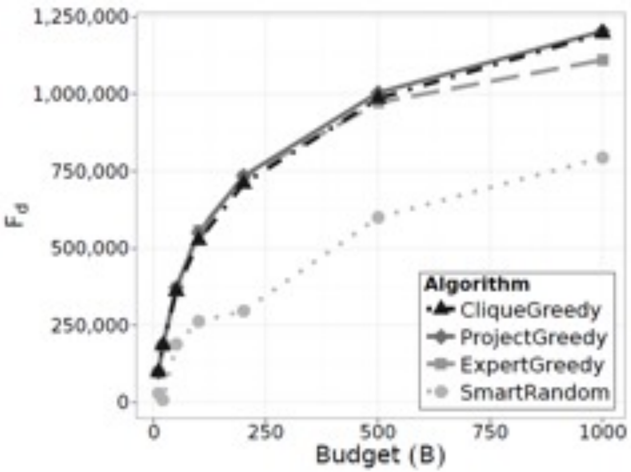
- Freelancer
- Guru

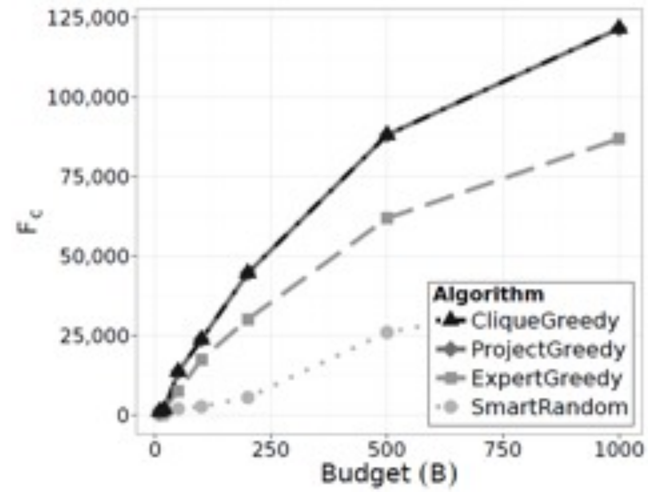


Experiments (Guru)

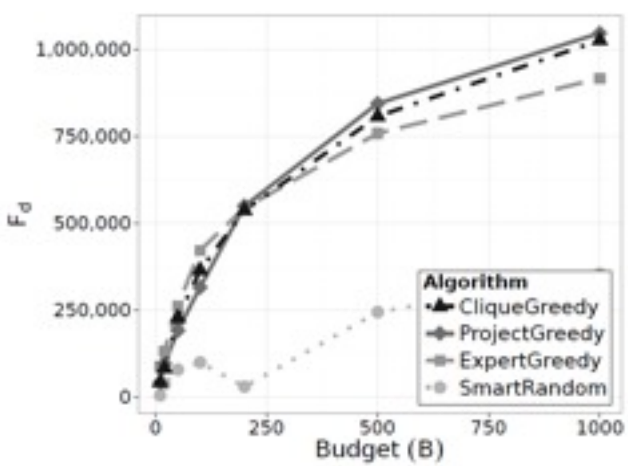
- Dollar-based
- Competition-based

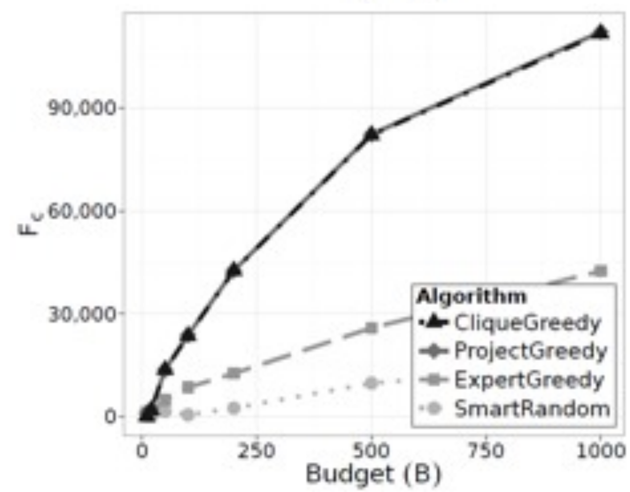
ClusterHire





t-ClusterHire

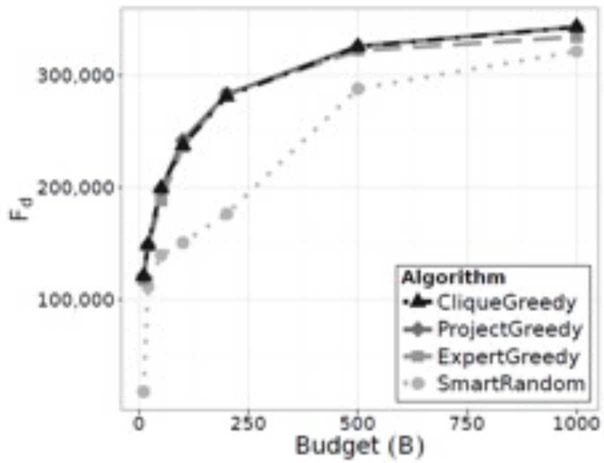


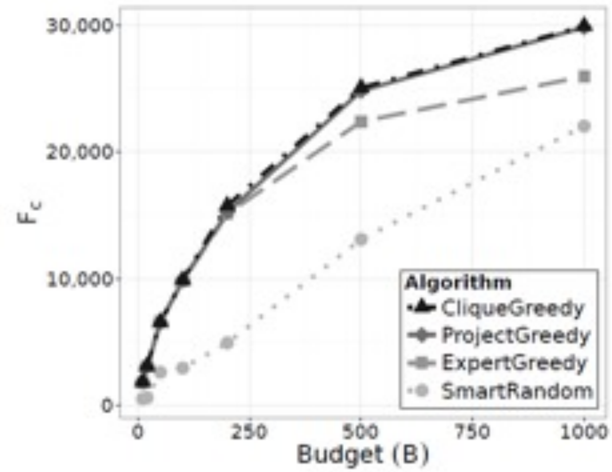


Experiments (Freelancer)

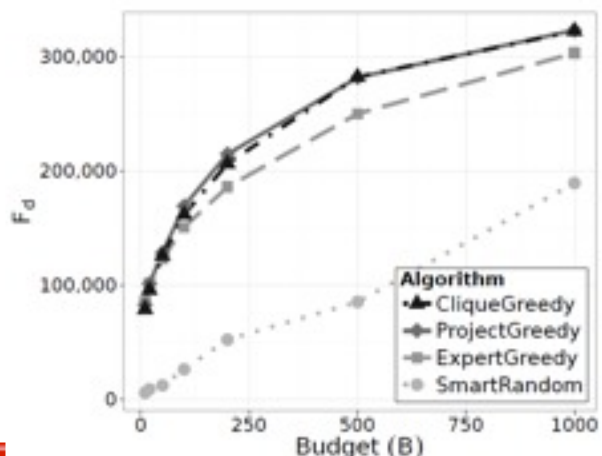
- Dollar-based
- Competition-based

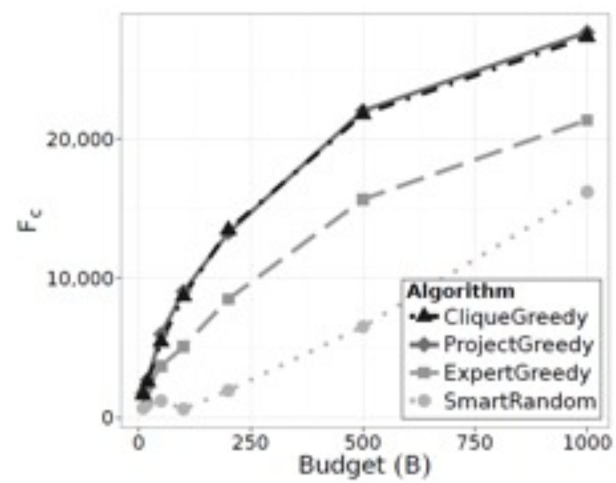
ClusterHire





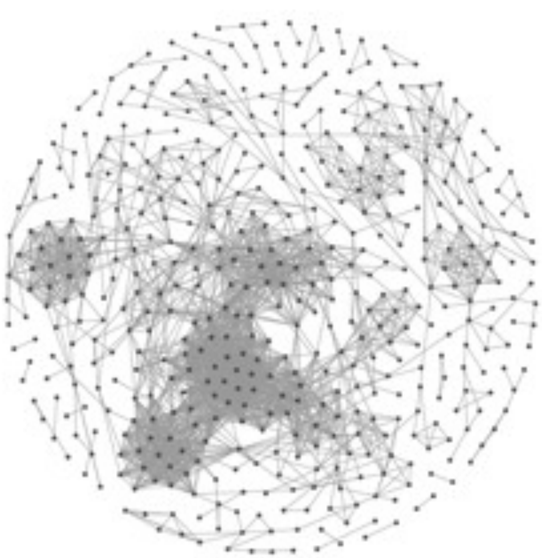
t-ClusterHire



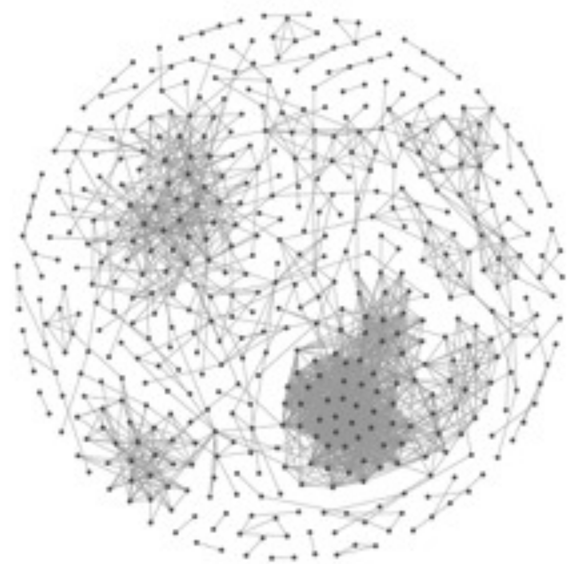


Experiments

- Performance of CliqueGreedy
  - Freelancer
  - Guru



Nodes: 721
Cliques: 520



Nodes: 1764
Cliques: 1660

BOSTON UNIVERSITY
evimaria@cs.bu.edu

Roadmap

- Background
- Team formation and cluster hires
- **Team formation in the presence of a social network**
- Inferring abilities of experts
- Team formation in educational settings

BOSTON UNIVERSITY
evimaria@cs.bu.edu

Setting [LLT'09]

- Experts (defining the set V, with $|V|=n$):
  - Every expert i is associated with **a set of skills** X_i
  - and **a price** p_i
- Tasks
  - Every task T is associated with a set of skills (T) **required** for performing the task
- A social network of experts ($G=(V,E)$)
  - Edges indicate ability to work well together

	Team Formation
Experts' skills	Known
Participation of experts in teams	Unknown
Network structure	Known

BOSTON UNIVERSITY

evimaria@cs.bu.edu

Team formation in the presence of a social network

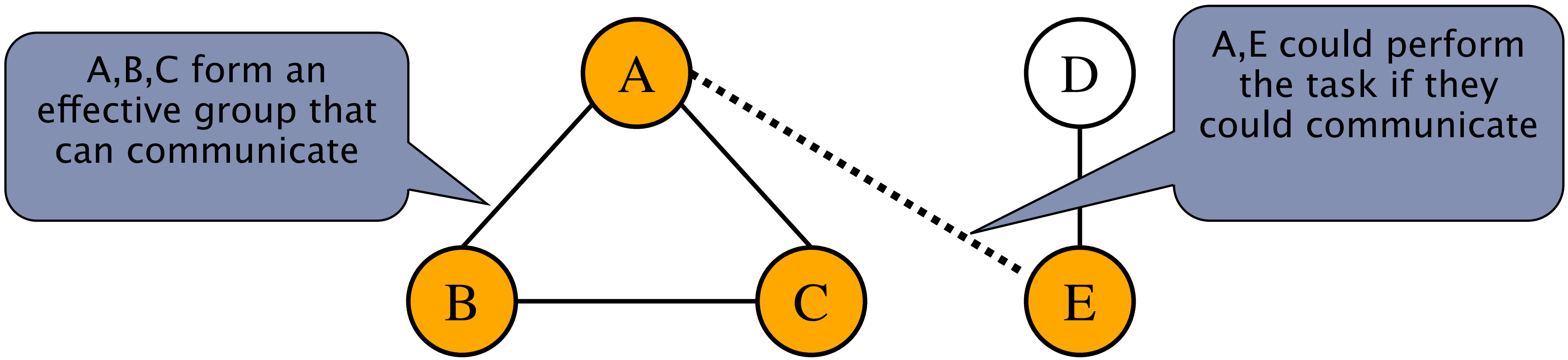
- Given a task and a set of experts organized in a network find the subset of experts that can effectively perform the task
- Task: set of required skills
- Expert: has a set of skills
- Network: represents strength of relationships

evimaria@cs.bu.edu

Coverage is NOT enough

T={algorithms,java,graphics,python}

Alice	Bob	Cynthia	David	Eleanor
{algorithms}	{python}	{graphics, java}	{graphics}	{graphics,java,python}



Communication: the members of the team must be able to efficiently communicate and work together

Problem definition (EffectiveTeam)

- Given a task and a social network of individuals, find the subset (team) of individuals that can effectively perform the given task.

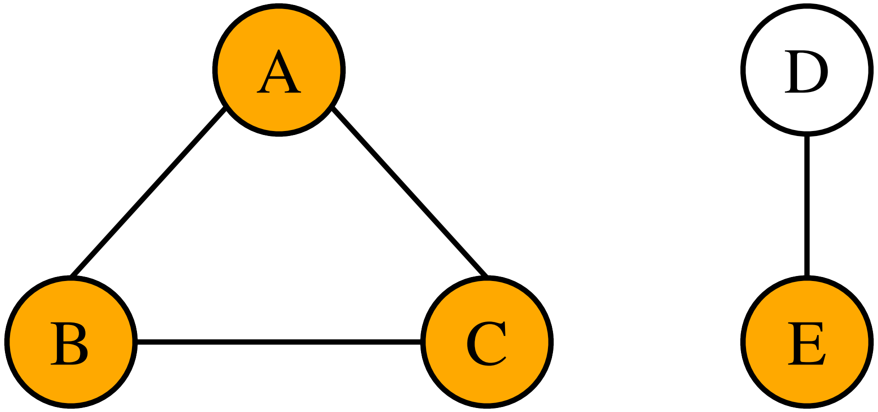
- **Thesis:** Good teams are teams that have the necessary skills and can also communicate effectively

evimaria@cs.bu.edu

How to measure effective communication?

The longest shortest path between any two nodes in the subgraph

- Diameter of the subgraph defined by the group members

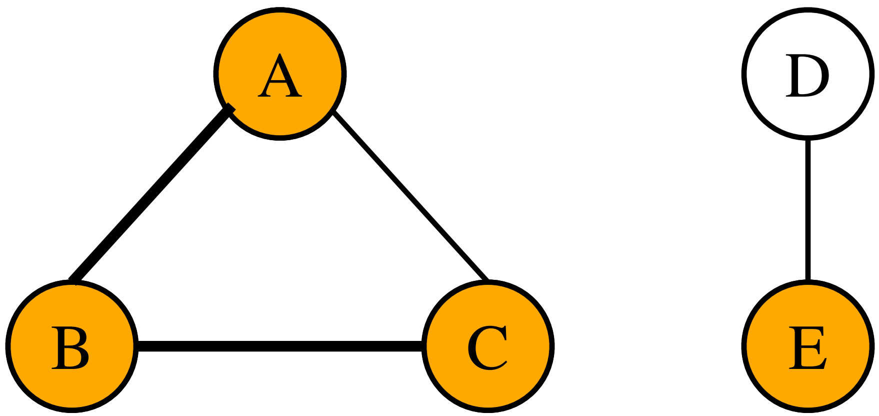


diameter = 1

How to measure effective communication?

The total weight of the edges of a tree that spans all the team nodes

- MST (Minimum spanning tree) of the subgraph defined by the group members



MST = 2

Problem definition (MinDiameter)

- Given a task and a social network G of experts, find the subset (team) of experts that can perform the given task and they define a subgraph in G with the minimum diameter.

- Problem is NP-hard

evimaria@cs.bu.edu

The RarestFirst algorithm

- Find Rarest skill α_{rare} required for a task
- S_{rare} group of people that have α_{rare}
- Evaluate star graphs, centered at individuals from S_{rare}
- Report cheapest star

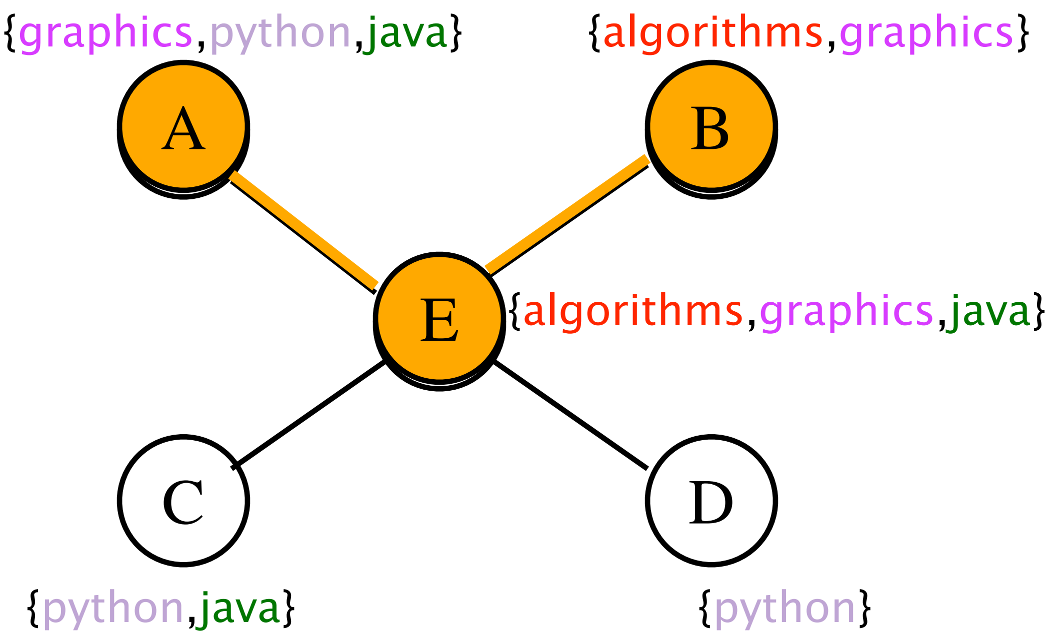
Running time: Quadratic to the number of nodes

Approximation factor: 2xOPT

evimaria@cs.bu.edu

The RarestFirst algorithm

T={algorithms,java,graphics,python}



Skills:

- algorithms
- graphics
- java
- python

α_{rare} = algorithms

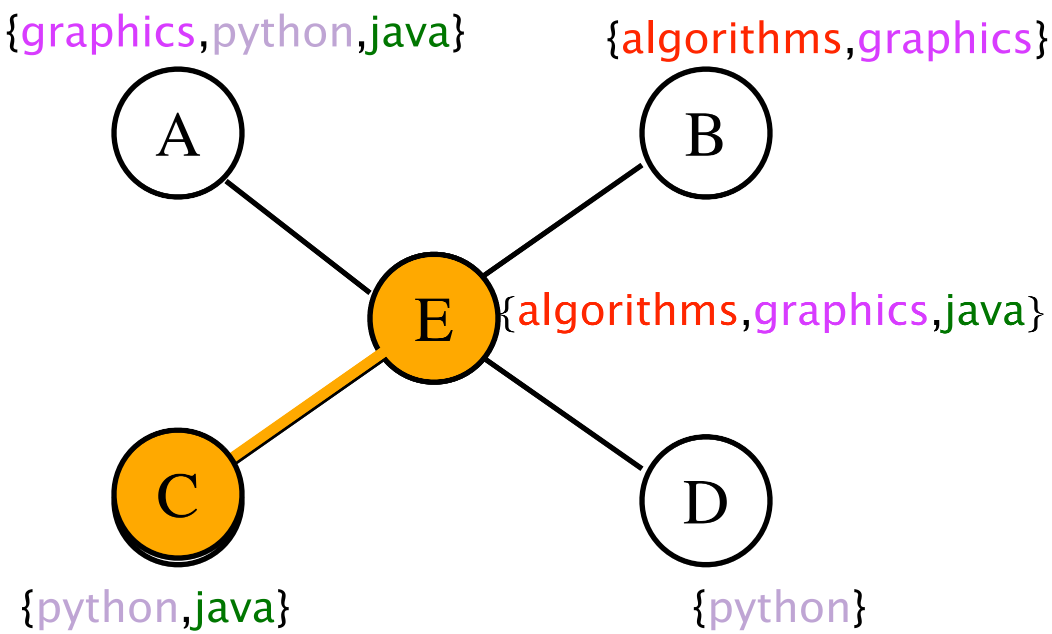
S_{rare} ={Bob, Eleanor}

Diameter = 2

evimaria@cs.bu.edu

The RarestFirst algorithm

T={algorithms,java,graphics,python}



Skills:

- algorithms
- graphics
- java
- python

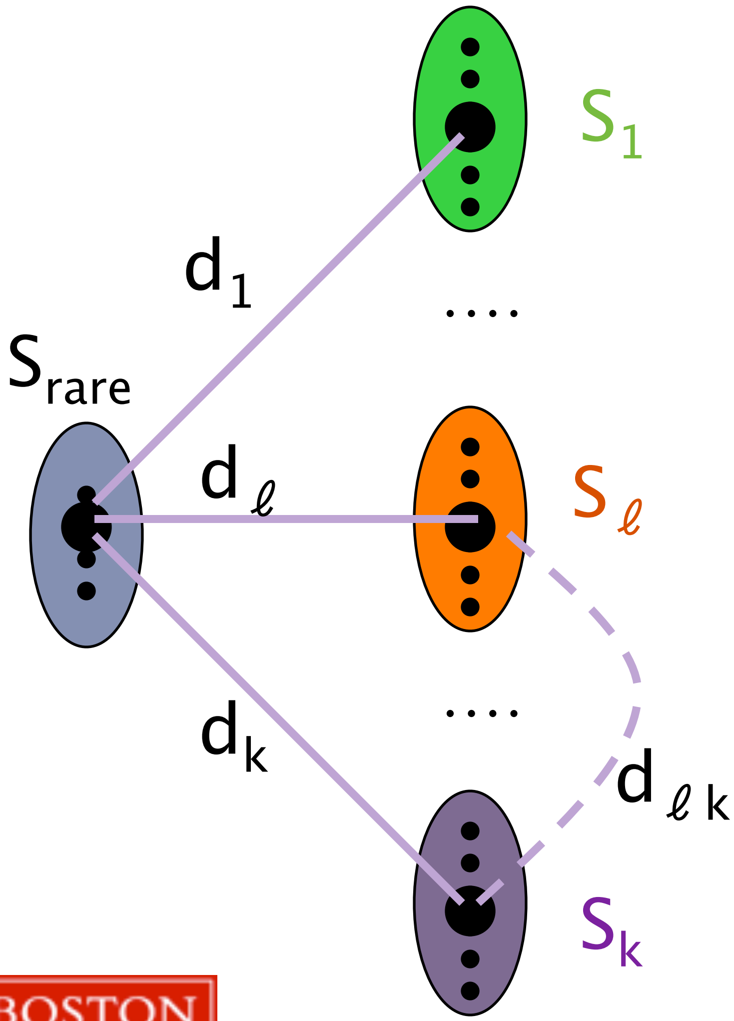
α_{rare} = algorithms

S_{rare} ={Bob, Eleanor}

Diameter = 1

evimaria@cs.bu.edu

Analysis of RarestFirst



- $D = \max\{d_\ell, d_k, d_{\ell k}\}$
- Fact: $OPT \geq d_\ell$
- Fact: $OPT \geq d_k$
- $D \leq d_{\ell k} \leq d_\ell + d_k \leq 2{*}OPT$

evimaria@cs.bu.edu

Problem definition (MinMST)

- Given a task and a social network G of experts, find the subset (team) of experts that can perform the given task and they define a subgraph in G with the minimum MST cost.

- Problem is NP-hard

evimaria@cs.bu.edu

The SteinerTree problem

- Graph G(V,E)

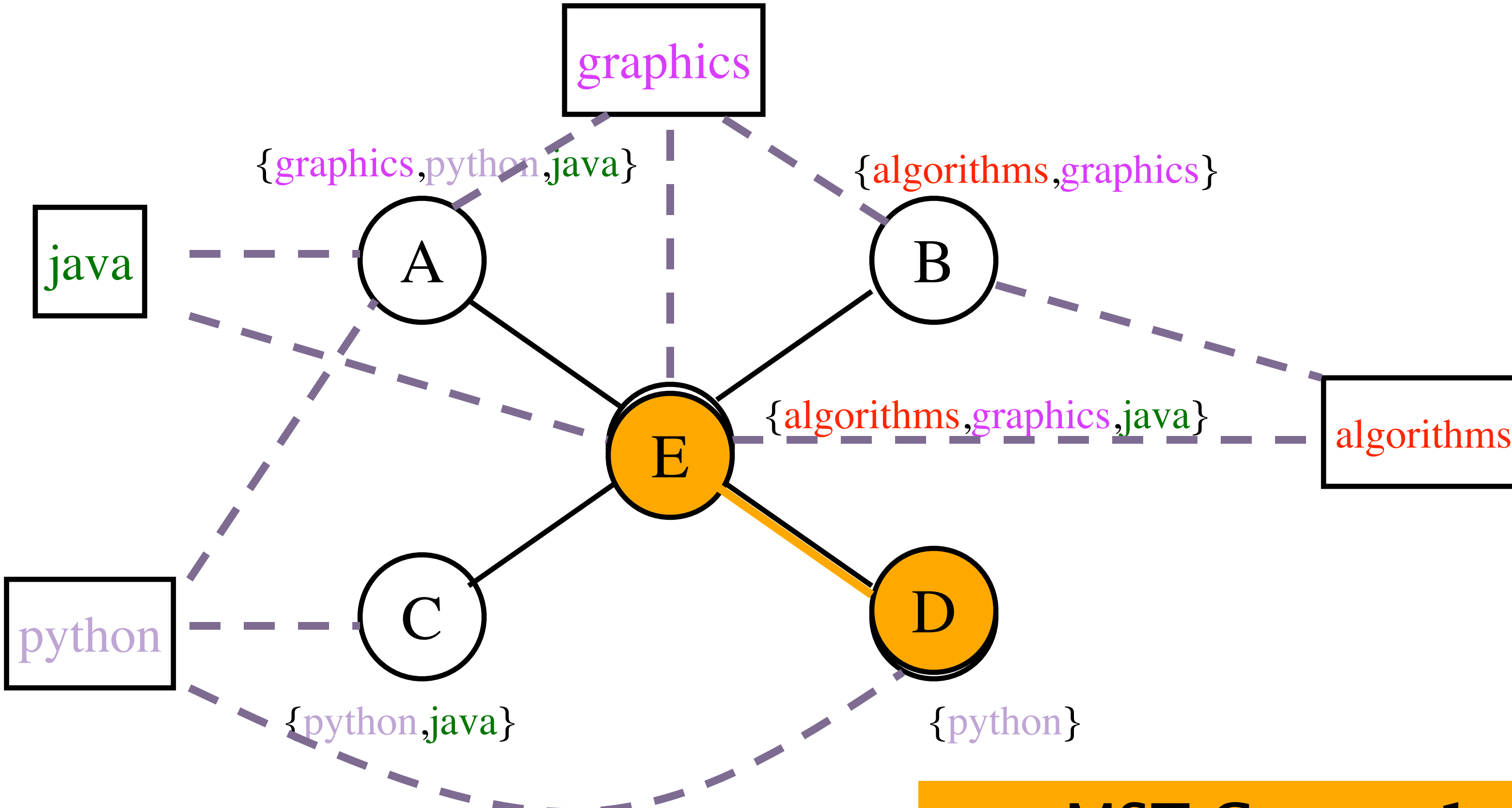
  Required vertices

- Partition of V into V = {R,N}

- Find G' subgraph of G such that G' contains all the required vertices (R) and MST(G') is minimized

The EnhancedSteiner algorithm

T={**algorithms**,**java**,**graphics**,**python**}



MST Cost = 1

Exploiting the SteinerTree problem further

- Graph G(V,E)

Required vertices

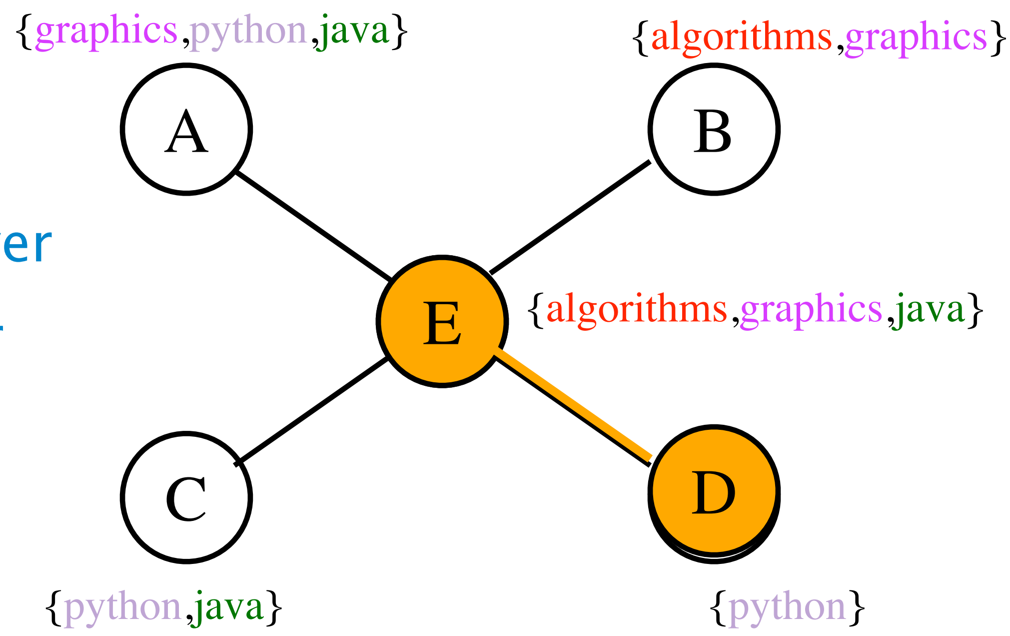
- Partition of V into V = {R,N}
- Find G' subgraph of G such that G' contains all the required vertices (R) and MST(G') is minimized

evimaria@cs.bu.edu

The CoverSteiner algorithm

T={algorithms,java,graphics,python}

1. Solve SetCover
2. Solve Steiner

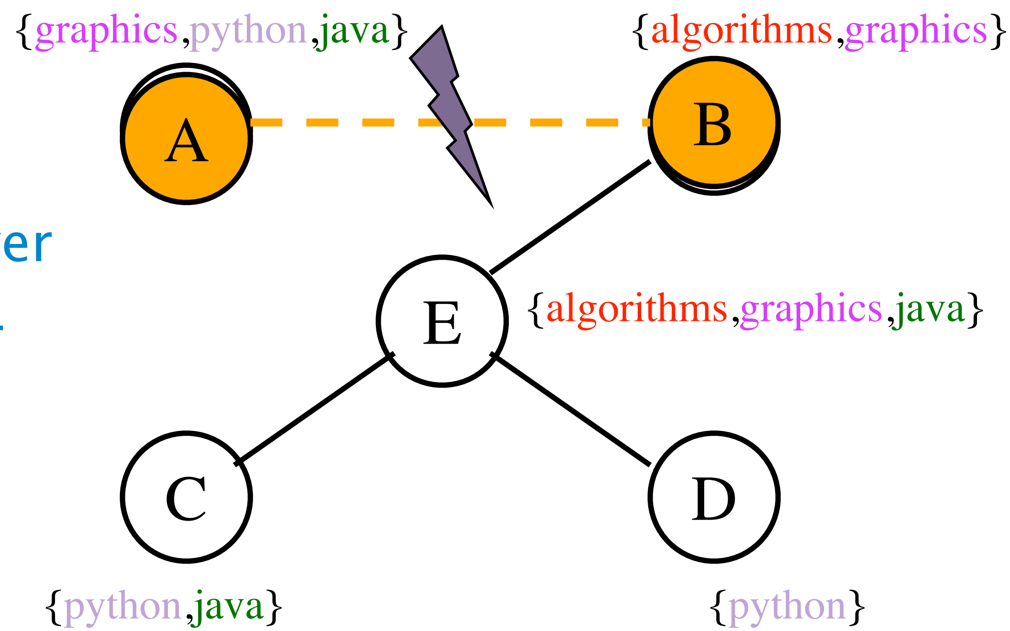


MST Cost = 1

evimaria@cs.bu.edu

How good is CoverSteiner?

T={algorithms,java,graphics,python}

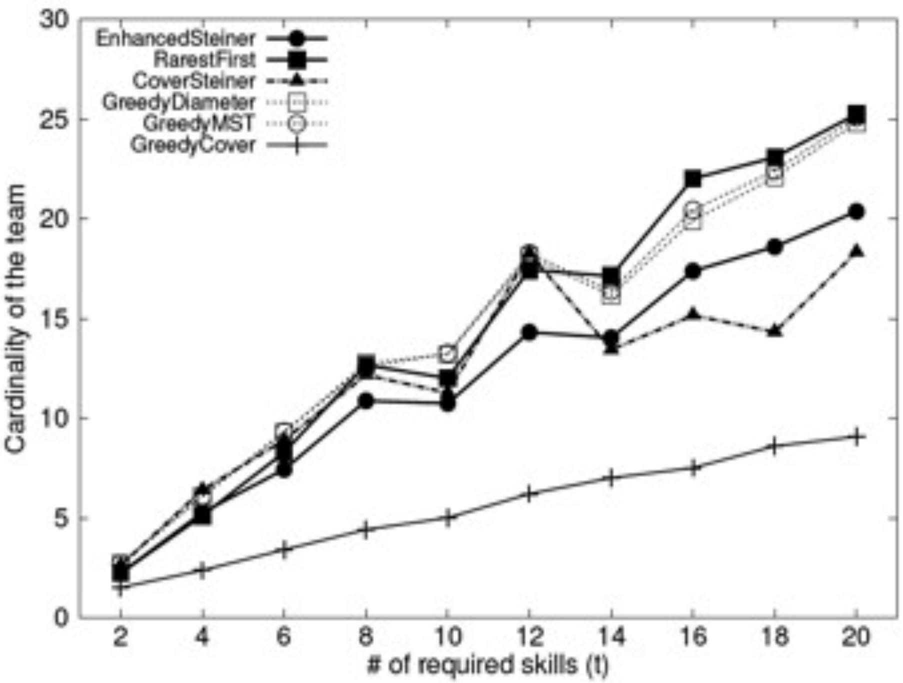


1. Solve SetCover
2. Solve Steiner

MST Cost = Infty

BOSTON UNIVERSITY

Experiments – Cardinality of teams



Dataset

DBLP graph (DB, Theory, ML, DM)

~6000 authors

~2000 features

Features: keywords appearing in papers

Tasks: Subsets of keywords with different cardinality **k**

evimaria@cs.bu.edu

Example teams (I)

- S. Brin, L. Page: The anatomy of a large-scale hypertextual Web search engine
  - **Paolo Ferragina, Patrick Valduriez, H. V. Jagadish, Alon Y. Levy, Daniela Florescu** Divesh Srivastava, S. Muthukrishnan
  - **P. Ferragina ,J. Han, H. V.Jagadish, Kevin Chen-Chuan Chang, A. Gulli**, S. Muthukrishnan, Laks V. S. Lakshmanan

Example teams (II)

- J. Han, J. Pei, Y. Yin: Mining frequent patterns without candidate generation
  - F. Bronchi
  - A. Gionis, H. Mannila, R. Motwani

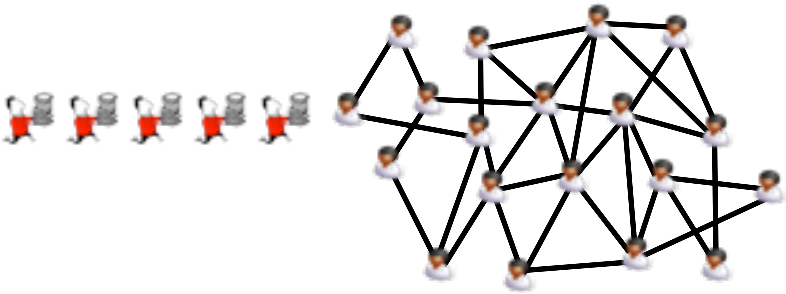
evimaria@cs.bu.edu

Extensions

- Other measures of effective communication
  - density, number of times a team member participates as a mediator, information propagation
- Other practical restrictions
  - Incorporate ability levels
- Online team formation [ABCGL'12]
  -

Setting

- Pool of people/experts with different skills
- People are connected through a social network
- Stream of jobs/tasks arriving online
- Jobs have some skill requirements
- **Goal:** Create teams on-the-fly for each job
  - Select the right team
  - Satisfy various criteria

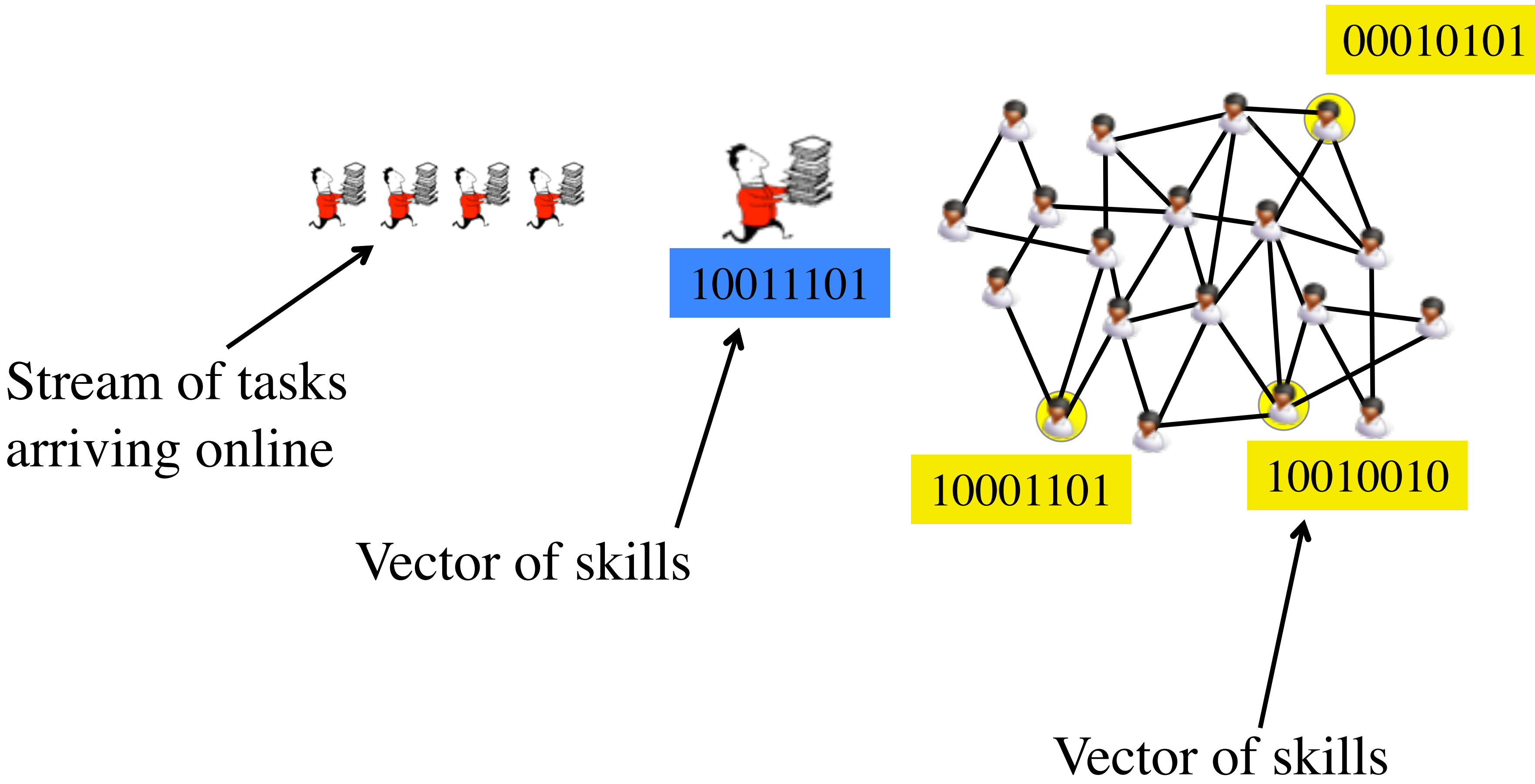


Criteria

- Fitness
  - E.g. if fitness is success rate, maximize expected number of successful jobs
  - Depends on:
    - People skills
    - Ability to coordinate
- Efficiency
  - Do not load people very much
- Fairness
  - Everybody should be involved in roughly the same number of jobs
-

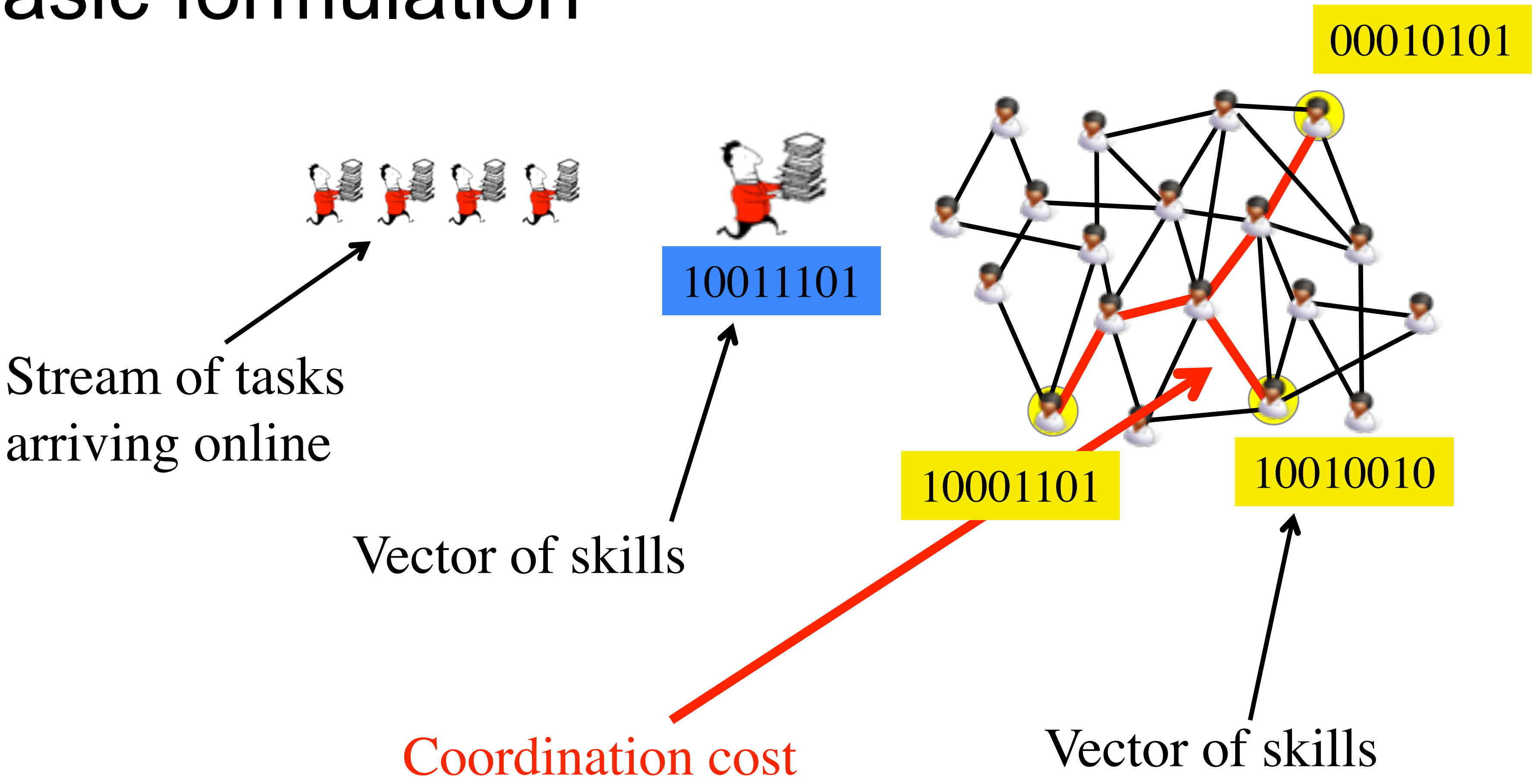
evimaria@cs.bu.edu

Basic formulation



evimaria@cs.bu.edu

Basic formulation

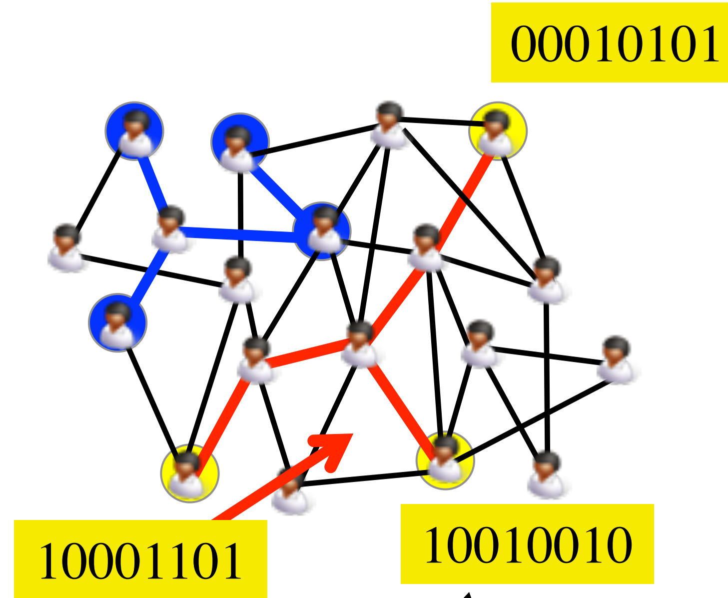


Basic formulation

Stream of tasks arriving online

10011101

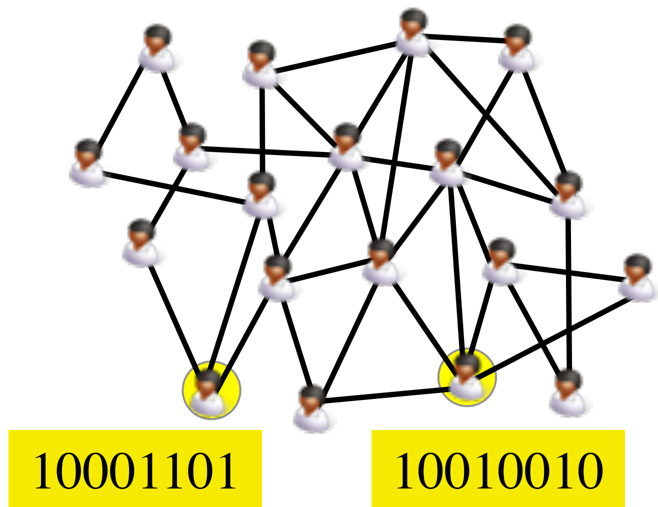
Vector of skills



Coordination cost

Vector of skills

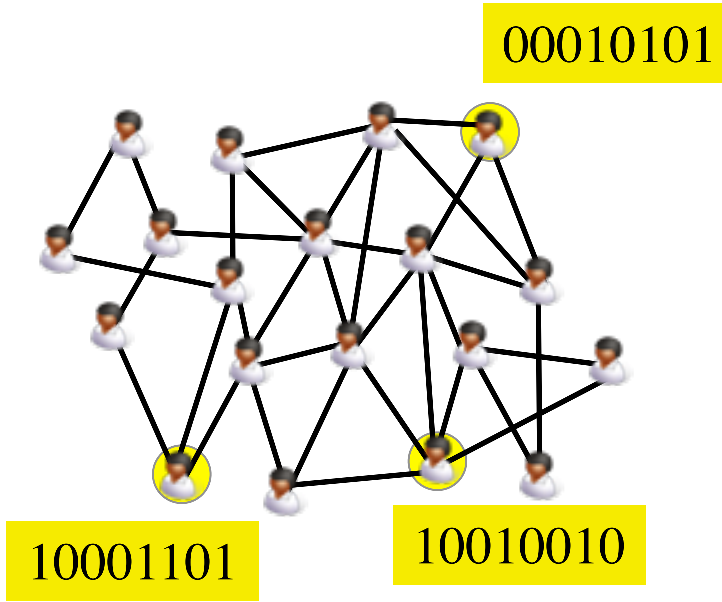
Basic formulation: Skills and people



- n people/experts $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n$
- m skills $\mathcal{S} = \{0,1\}^m$
- Each person has some skills $\mathbf{p}^i \in \mathcal{S}$

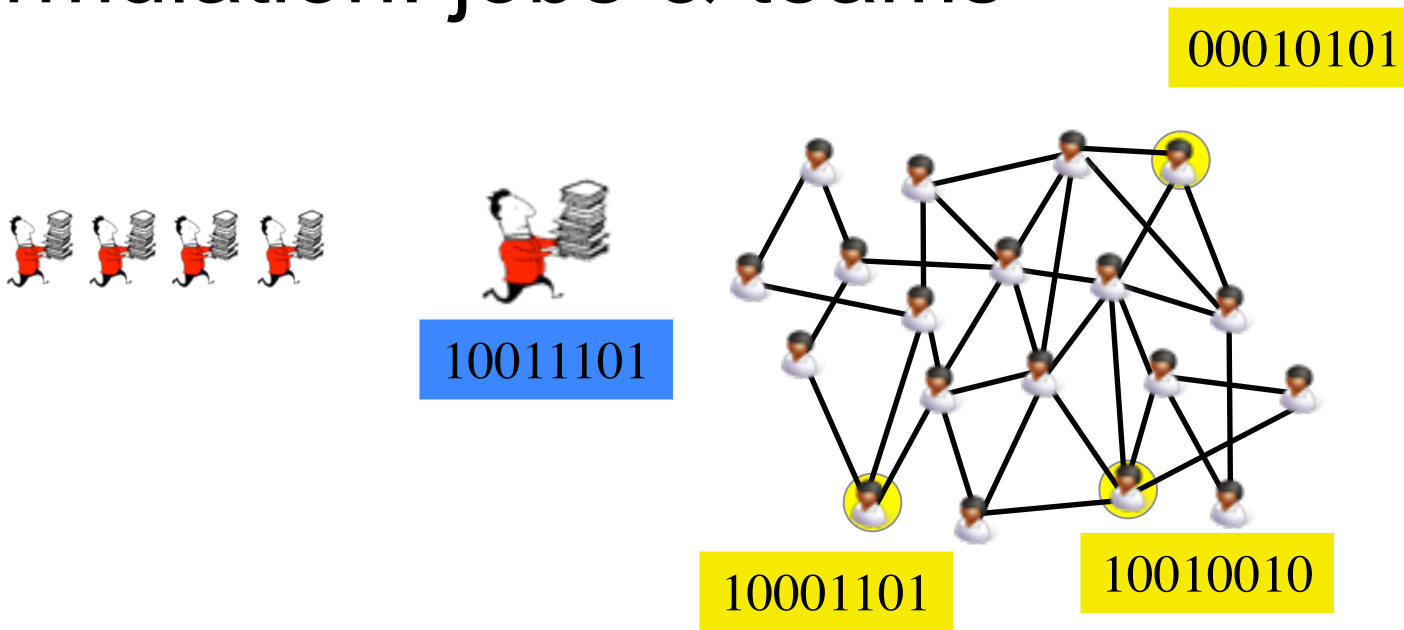
evimaria@cs.bu.edu

Basic formulation: jobs & teams



- Stream of k Jobs/Tasks $\mathbf{J}^1, \mathbf{J}^2, \ldots, \mathbf{J}^k$
- A job requires some skills $\mathbf{J}^j \in \mathcal{S}$
- k Teams are created online $Q^j \subseteq \{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n\}$
- A team must **cover** all job skills

Basic formulation: jobs & teams



- Stream of k Jobs/Tasks $\mathbf{J}^1, \mathbf{J}^2, \ldots, \mathbf{J}^k$
- A job requires some skills $\mathbf{J}^j \in \mathcal{S}$
- k Teams are created online $Q^j \subseteq \{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n\}$
- A team must **cover** all job skills

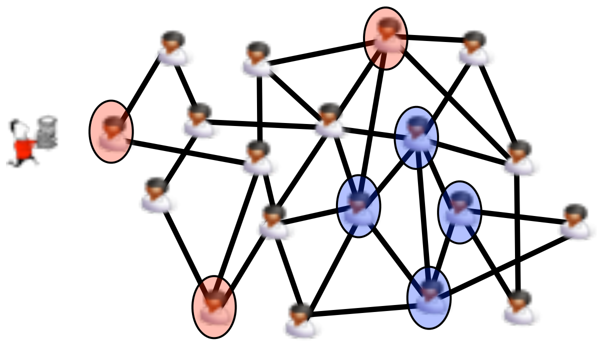
- **Load** of p: L(p) = total # of teams having p

BOSTON UNIVERSITY

evimaria@cs.bu.edu

Coordination cost

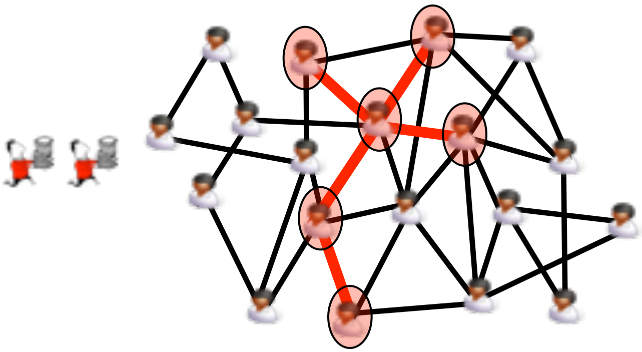
- Coordination cost measures the compatibility of the team members
- Example of $d(\mathbf{p}^i, \mathbf{p}^j)$:
  - Degree of knowledge
  - Time-zone difference
  - Past collaboration



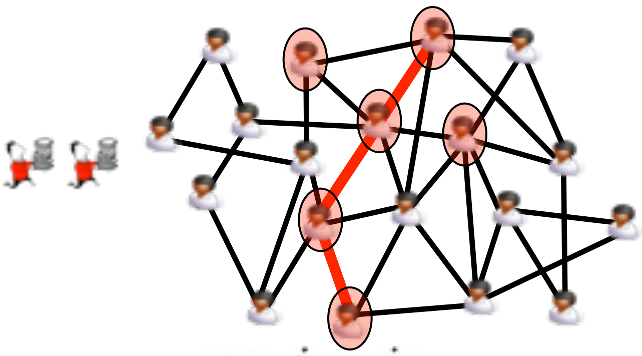
- Select teams that minimizes coordination cost $c(Q)$:
  - Steiner-tree cost
  - Diameter
  - Sum of distances

Coordination cost

- Steiner-tree cost



- Diameter



$$\max_{\mathbf{p}^i,\mathbf{p}^j \in Q} d(\mathbf{p}^i,\mathbf{p}^j)$$

- Sum of distances

$$\sum_{\mathbf{p}^i,\mathbf{p}^j \in Q} d(\mathbf{p}^i,\mathbf{p}^j)$$

Conflicting goals

- We want to create teams **online** that **minimize**
  - **Load**
  - **Unfairness**
  - **Coordination cost**

  and **cover** each job.

- **How can we model all these requirements?**

evimaria@cs.bu.edu

Our modeling approach

- Set a desirable coordination cost upper bound B
- Online solve

$$\min \max_i L(\mathbf{p}^i)$$
$$Q^j \text{ covers } \mathbf{J}^j \quad \forall j$$
$$c(Q^j) \leq B \quad \forall j.$$

Load of person i

Team j covers job j

Bounded coordination cost

- Must concurrently solve various combinatorial problems:
  - Set cover
  - Steiner tree
  - Online makespan minimization

BOSTON UNIVERSITY

evimaria@cs.bu.edu

Our modeling approach

Job	p_1	p_2	p_3	p_4	p_5	p_6	p_7	Q_j
1		✓		✓	✓			$Q_1 = \{p_2, p_4, p_5\}$
2	✓			✓		✓		$Q_2 = \{p_1, p_4, p_6\}$
3			✓	✓				$Q_3 = \{p_3, p_4\}$
4	✓				✓		✓	$Q_4 = \{p_1, p_5, p_7\}$
5		✓	✓	✓	✓			$Q_5 = \{p_2, p_3. p_4, p_5\}$
6			✓		✓	✓		$Q_6 = \{p_3, p_5, p_6\}$
7	✓	✓						$Q_7 = \{p_1, p_2\}$
8	✓	✓	✓	✓			✓	$Q_8 = \{p_1, p_2, p_3, p_4, p_7\}$
9			✓	✓	✓			$Q_9 = \{p_3, p_4, p_5\}$
Load	4	4	5	6	5	2	2	

evimaria@cs.bu.edu

Algorithm ExpLoad

Load of **p** at time *t*

At each time step t, when a task arrives:

- Weight each person **p** by $w(\mathbf{p}) = (2n)^{L_t(\mathbf{p})}$
- Select team Q that
  - Covers all required skills
  - Satisfies $c(Q) \leq B$
  - Minimizes $\sum_{\mathbf{p} \in Q} w(\mathbf{p})$
- **Theorem.** If we can solve this problem optimally, then Competitive ratio = $O(\log k)$. This is the best possible.

evimaria@cs.bu.edu

The ExpLoad algorithm

Load of **p** at time *t*

At each time step t, when a task arrives:

- Weight each person **p** by $w(\mathbf{p}) = (2n)^{L_t(\mathbf{p})}$

$$\text{Competitive ratio} = \max_I \frac{\text{cost of alg's online solution on instance } I}{\text{best offline solution on instance } I}$$

- Select team Q that
  - Covers all required skills
  - Satisfies $c(Q) \leq B$
  - Minimizes $\sum_{\mathbf{p} \in Q} w(\mathbf{p})$
- **Theorem.** If we can solve this problem optimally, then Competitive ratio $= O(\log k)$. This is the best possible.

evimaria@cs.bu.edu

The ExpLoad algorithm

Load of **p** at time *t*

At each time step t, when a task arrives:

- Weight each person **p** by $w(\mathbf{p}) = (2n)^{L_t(\mathbf{p})}$
- Select team Q that
  - Covers all required skills
  - Satisfies $c(Q) \leq B$
  - Minimizes $\sum_{\mathbf{p} \in Q} w(\mathbf{p})$

We can solve this problem only approximately.

- **Theorem.** If we can solve this problem optimally, then Competitive ratio = $O(\log k)$. This is the best possible.

evimaria@cs.bu.edu

Roadmap

- Background
- Team formation and cluster hires
- Team formation in the presence of a social network
- **Inferring abilities of experts**
- Team formation in educational settings

Setting [GLT'12]

- Experts (defining the set V, with |V|=n):
  - Every expert i is associated with **a set of skills** X_i
  - and **a price** p_i
- Tasks
  - Every task T is associated with a set of skills (T) **required** for performing the task
- A social network of experts (G=(V,E))
  - Edges indicate ability to work well together

	Team Formation	Skill Attribution
Experts' skills	Known	Unknown
Participation of experts in teams	Unknown	Known
Network structure	Known	Irrelevant

evimaria@cs.bu.edu

The Skill-Attribution problem

- Input: a set of teams and the tasks they performed
  - Team T_1={A,B} performed task S_1={algorithms, databases}
  - Team T_2={B,C,D} performed task S_2={algorithms, system, programming}
  - Team T_3={A,B,C} performed task S_3={databases, algorithms, systems}
- Question: What are the contributions of each team member?
  - Team {A,B} appear to know algorithms and databases but who knows algorithms and who knows databases?
- Assumptions:
  - **Complementarity:** A team has a skill if at least one of its members has that skill
  - **Parsimony:** It is hard to imagine a world where all individuals have all skills

evimaria@cs.bu.edu

The Skill-Attribution problem

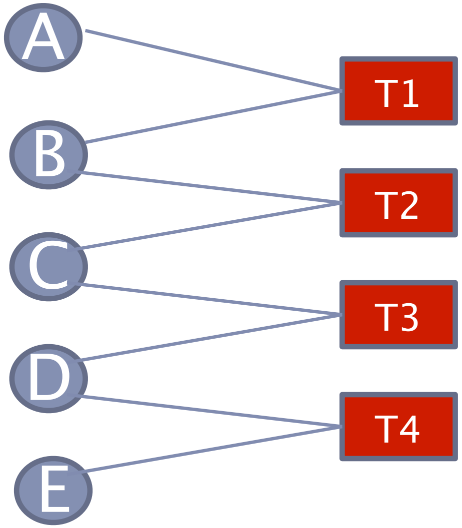
- The input introduces a set of constraints
  - Team T_1={A,B} performed task S_1={algorithms, databases}
  - Team T_2={B,C,D} performed task S_2={algorithms, system, programming}
  - Team T_3={A,B,C} performed task S_3={databases, algorithms, systems}
- A skill assignment is consistent if for every task T_i and every skill in $s \in S_i$ there exist at least one expert in T_i who has s.
  - A skill assignment is consistent if and only if it is consistent for every skill separately

Focus on the single-skill attribution problem

evimaria@cs.bu.edu

Skill vectors and hitting sets

- s = algorithms
- Team T_1={A,B}
- Team T_2={B,C}
- Team T_3={C,D}
- Team T_4={D,E}



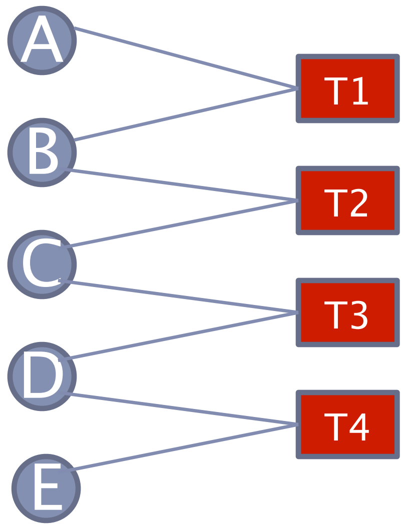
- A skill vector assigns skill s to individuals from V
- Any consistent skill vector is a hitting set for the set system $(\underbrace{T_1,T_2,\ldots,T_m}_{\text{Teams: subsets of individuals}}, \underbrace{V}_{\text{Universe of individuals}})$

BOSTON UNIVERSITY
evimaria@cs.bu.edu

Minimum skill attribution (v 0.0)

- For a single skill s, and input teams $T_1, T_2, \ldots, T_m$ find a consistent skill attribution with the minimum number of individuals possessing s.

- s = algorithms
- Team T_1={A,B}
- Team T_2={B,C}
- Team T_3={C,D}
- Team T_4={D,E}



- Minimum skill attribution: X^* = {B,D}
- Minimum skill attribution is as hard as the minimum hitting set problem
- X^* is a strictly parsimonious solution
- One solution is not enough:
  - Near-optimal attributions are ignored X'={A,C,D}, X''={A,C,E}, X'''={B,C,D}, X''''={B,C,E}

evimaria@cs.bu.edu

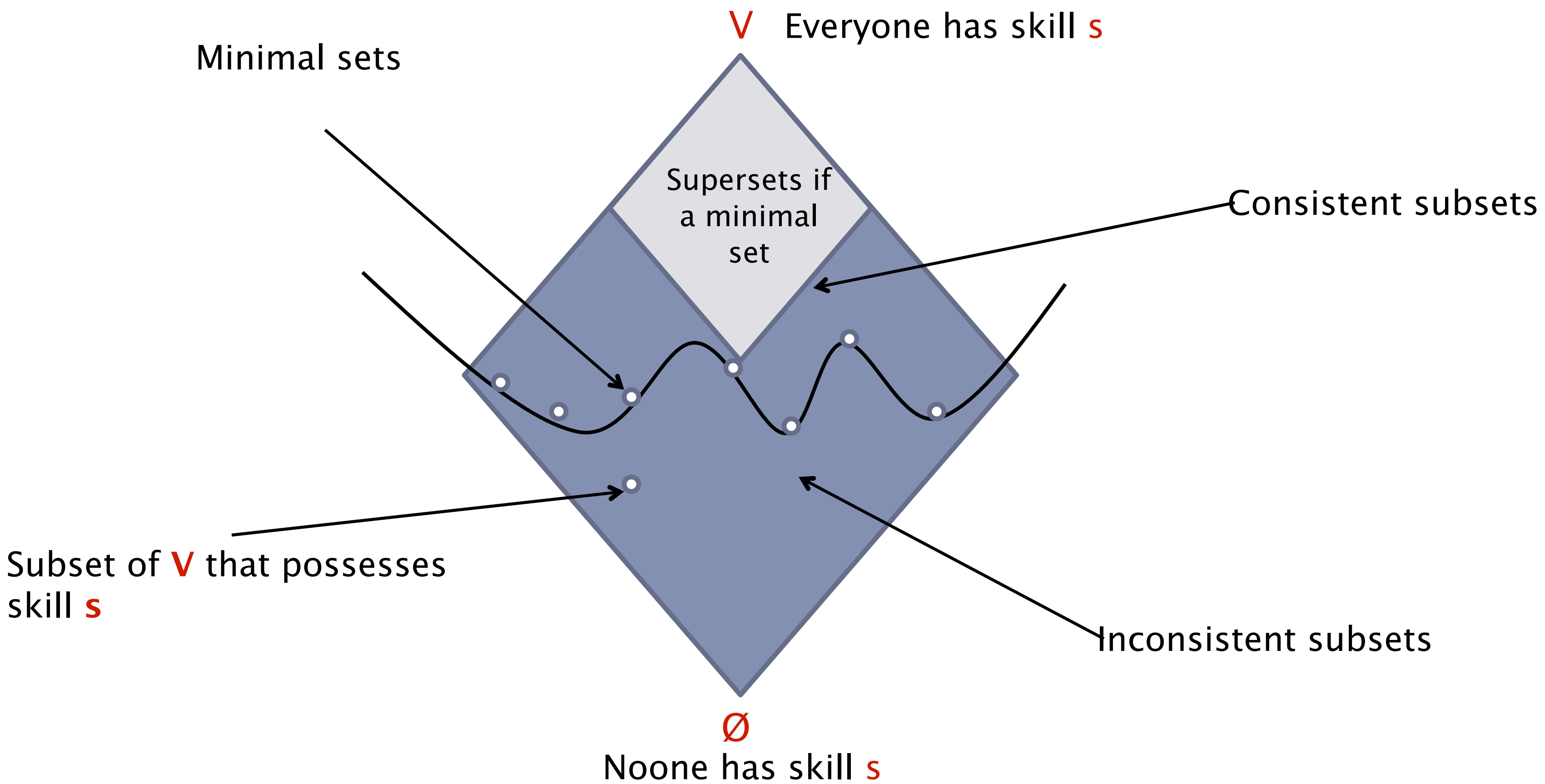
Counting all consistent skill vectors

- For a single skill s, and input teams $T_1,T_2,\ldots,T_m$ count for every individual in V the number of consistent skill vectors he participates in.

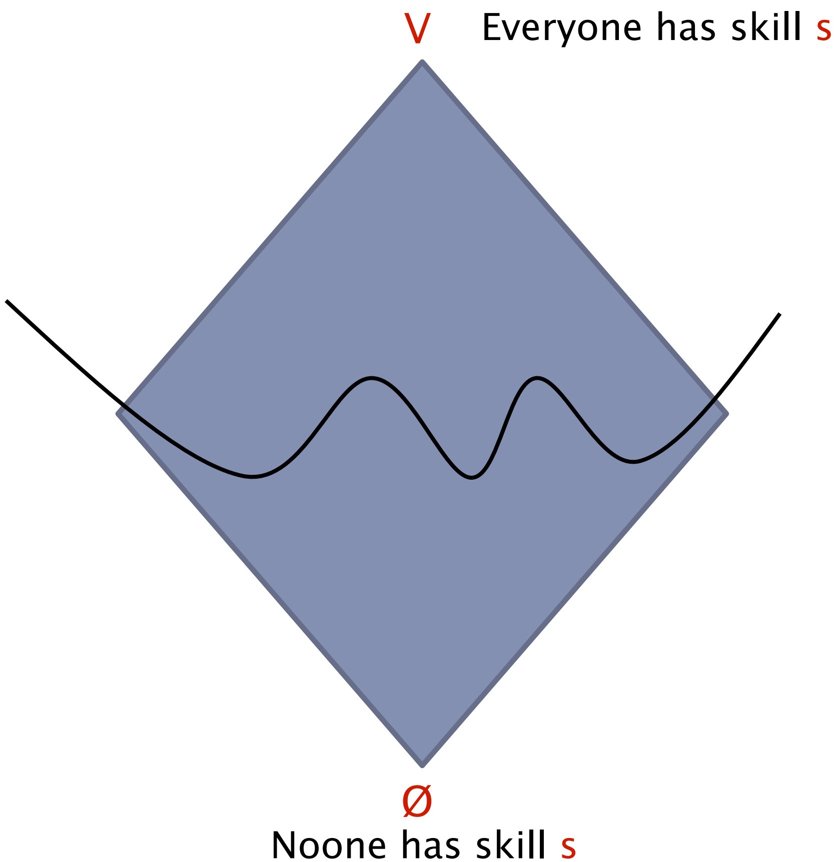
  - Equivalent to counting hitting sets for input $(T_1,T_2,\ldots,T_m ,V)$
  - #P-complete problem

evimaria@cs.bu.edu

The lattice of skill vectors

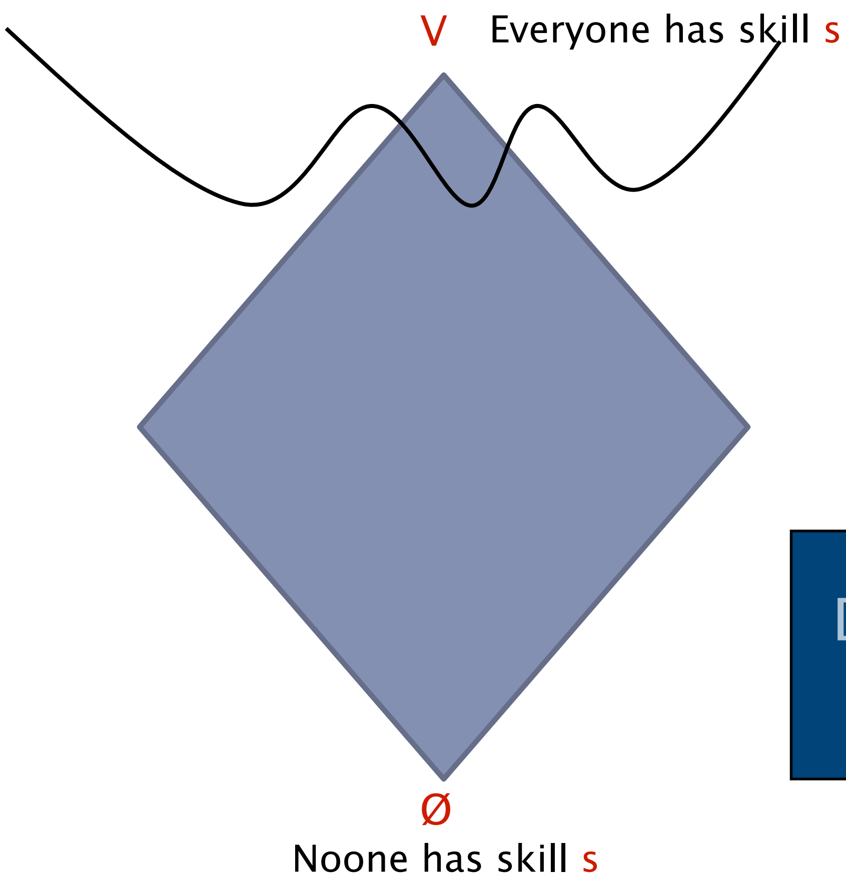


Counting all consistent skill vectors



- Naïve Monte-Carlo sampling
  - C=0
  - for i=1…N
    - Sample an element from the lattice; if it is consistent C++
  - return $(C/N)x2^n$

Counting all consistent skill vectors

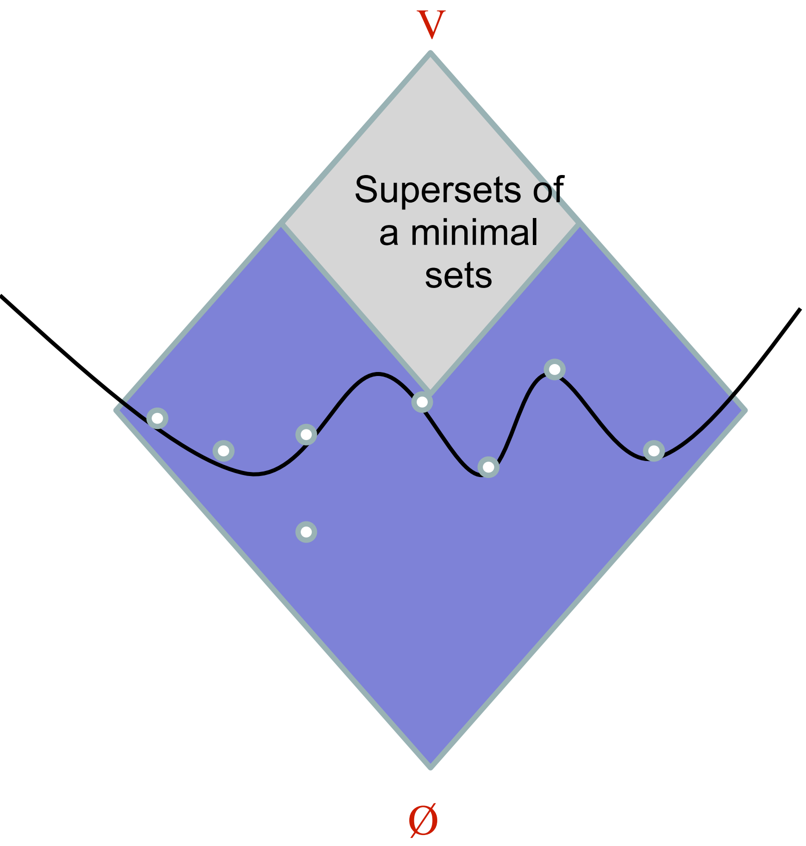


- Naïve Monte-Carlo sampling
  - C=0
  - for i=1…N
    - Sample an element from the lattice; if it is consistent C++
  - return $(C/N)\text{x}2^n$

Does not work when there are few consistent vectors

evimaria@cs.bu.edu

The ImportanceSampling algorithm



- Assume we know the set of minimal sets that contain **r**

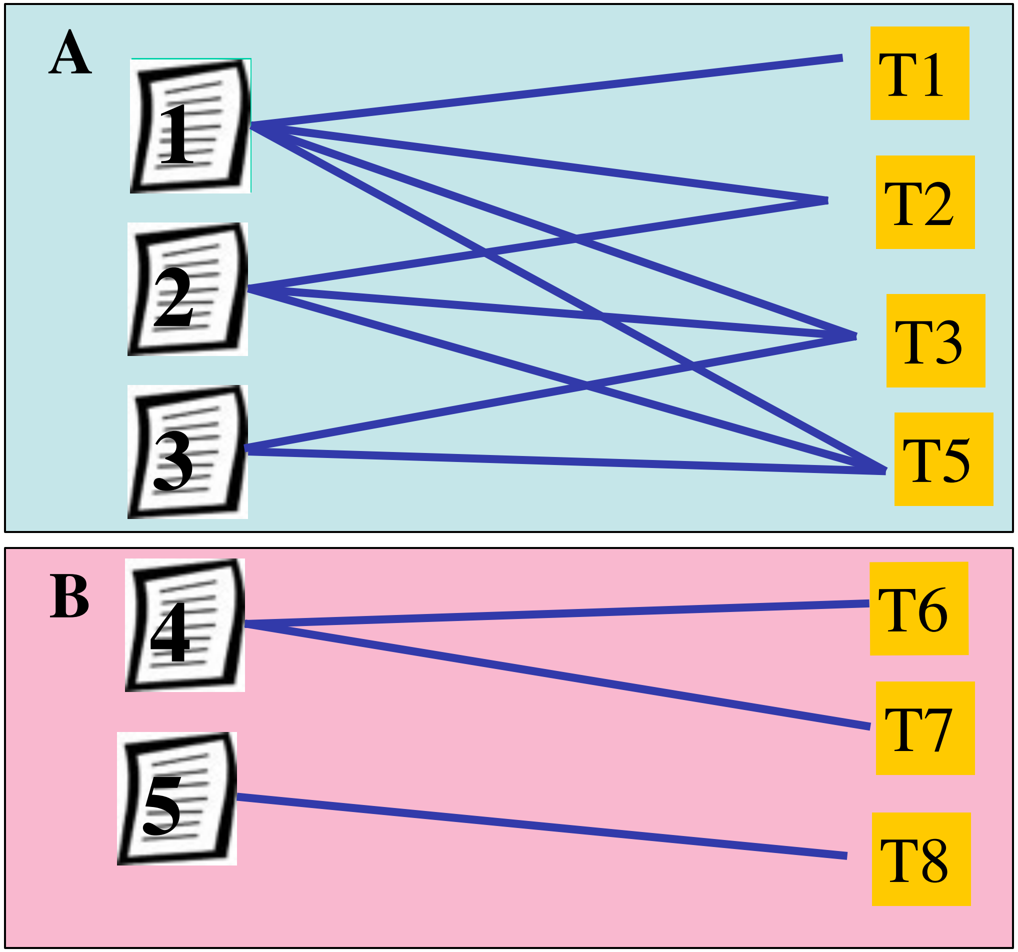
$$\mathbf{M(r)} = \{M_1, \ldots, M_k\}$$

- Sample consistent vectors from the space of hitting sets only

- Running time: polynomial in **k**

ImportanceSampling Speedups

- Run ImportanceSampling for all experts ***simultaneously***
- View the input as a bipartite graph and partition it into (almost) independent components
- Cluster together experts that participate in identical sets of teams into **super-experts**



ConsistentVectors(1) = ConsistentVectors(1,A)xConsistentVectors(B)

Ranking of experts

social networks	privacy	graphs
P. Mika (1)	A. Acquisti (1)	C. Faloutsos (1)
J. Golbeck (5)	M. S. Ackerman (3)	J. Kleinberg (2)
M. Richardson (5)	L. Faith Cranor (3)	J. Leskovec (2)
P. Singla (19)	B. Berendt (5)	R. Kumar (3)
L. Zhou (7)	S. Spiekermann (5)	A. Tomkins (3)
A. Java (19)	O. Gunther (19)	L. A. Adamic (3)
L. Ding (2)	J. Grossklags (5)	E. Vee (4)
T. Finin (2)	G. Hsieh (19)	P. Ginsparg (4)
A. Joshi (2)	K. Vaniea (19)	J. Gehrke (4)
R. Agrawal (19)	N. Sadeh (19)	B. A. Huberman (3)

Roadmap

- Background
- Team formation and cluster hires
- Team formation in the presence of a social network
- Inferring abilities of experts
- **Team formation in educational settings**

evimaria@cs.bu.edu

Team formation in educational settings [AGT'14]

- Consider a class of students
  - Different ability levels (single scores)
    - Example: GRE, TOEFL, SAT, …

How to form study groups?

evimaria@cs.bu.edu

Team formation in educational settings [AGT'14]

- Classical methods
  - Ability-Based Grouping
    - Grouping students with similar abilities together
  - Pseudo-Random Grouping
    - Grouping students based on some arbitrary ordering
    - Alphabetically, FCFS, …

evimaria@cs.bu.edu

Team formation in educational settings [AGT'14]

- Classical methods
  - Ability-Based Grouping
    - Grouping students with similar abilities together
  - Pseudo-Random Grouping
    - Grouping students based on some arbitrary ordering
    - Alphabetically, FCFS, …

Which method to use?

Inconclusive verdict from empirical studies

(Kulik 92, Loveless 13, McPartland 87)

Let's take a computational approach

Framework

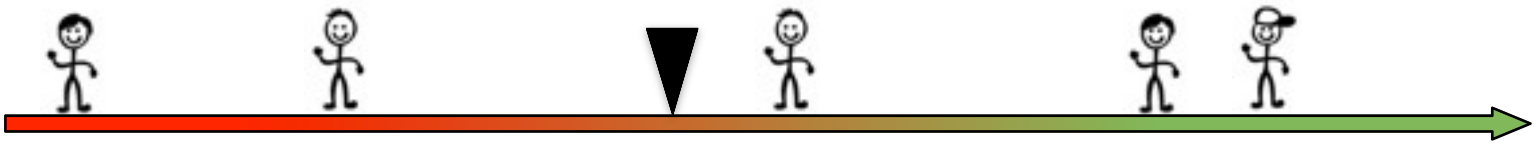
- Set of n students with abilities $\theta_1, \theta_2, \dots, \theta_n$
  - Ability scores are real number ($\theta_i \in R$)

- Collective Ability of a team T
  - Represents the group ability
  - Expected Ability $\widehat{\Theta}_T = 1/|T| \sum_{i \in T} \theta_i$
    - Choose a random student and ask him

evimaria@cs.bu.edu

Framework

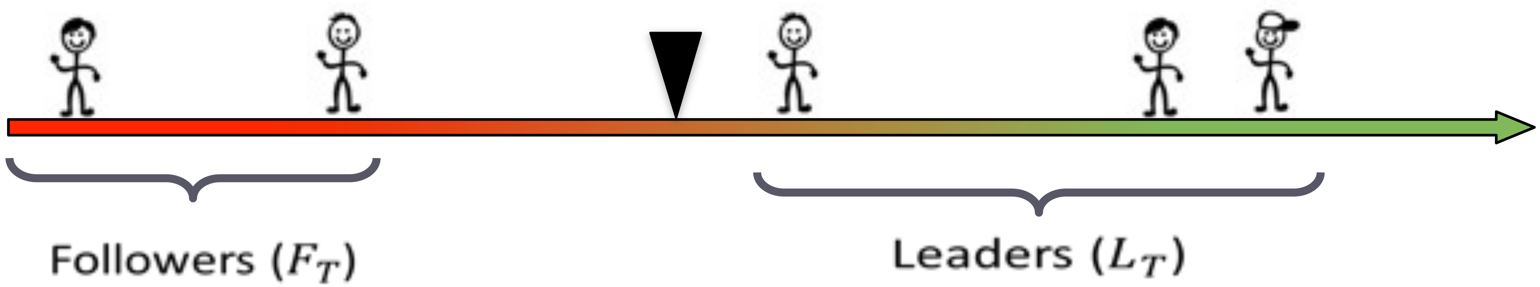
- Two groups of students in a study group
  - Students below the collective ability
  - Students above the collective ability



evimaria@cs.bu.edu

Framework

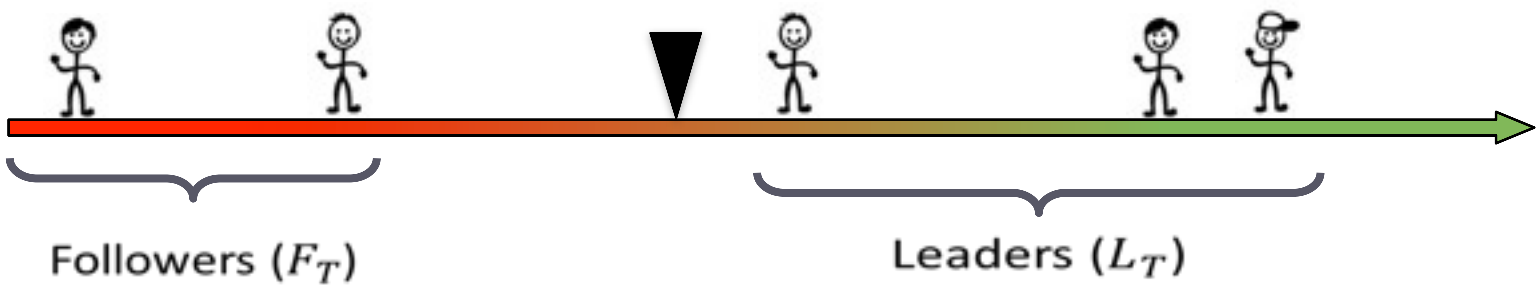
- Two groups of students in a study group
  - Students below the collective ability
  - Students above the collective ability



- Mostly learn from other members of the group
- Mostly improve by teaching others

Framework

- Two groups of students in a study group
  - Students below the collective ability
  - Students above the collective ability



- Followers: Mostly learn from other members of the group
- Leaders: Mostly improve by teaching others

Our Focus

  - Maximize the number of such students

evimaria@cs.bu.edu

Problem

- Partitioning students into study groups
  - Partition students into l groups of size k to maximize the gain
    - Gain = sum of the number of followers in each group
- Theorem:
  - NP-hard to solve
  - PARTITION problem reduces this problem

evimaria@cs.bu.edu

Algorithm

- Partitioning students into study groups
  - Partition students into l groups of size k to maximize the gain
- Algorithm:
  - Find the best team of size k from the pool of students
  - Remove the team from the pool
  - Repeat until all groups are formed

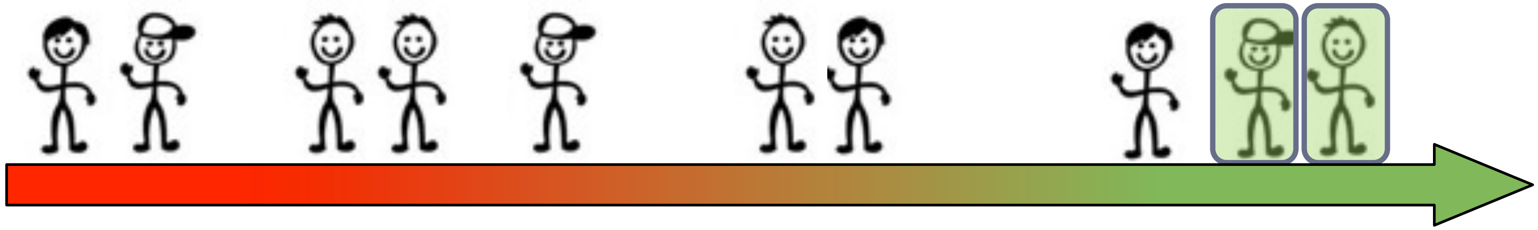
evimaria@cs.bu.edu

Algorithm

- Partitioning students into study groups
  - Partition students into l groups of size k to maximize the gain
- Algorithm:
  - **Find the best team** of size k from the pool of students
  - Remove the team from the pool
  - Repeat until all groups are formed
- Best Team
  - Team with the maximum gain (i.e., number of followers)
  - How to find the best team?

Finding the **best** team

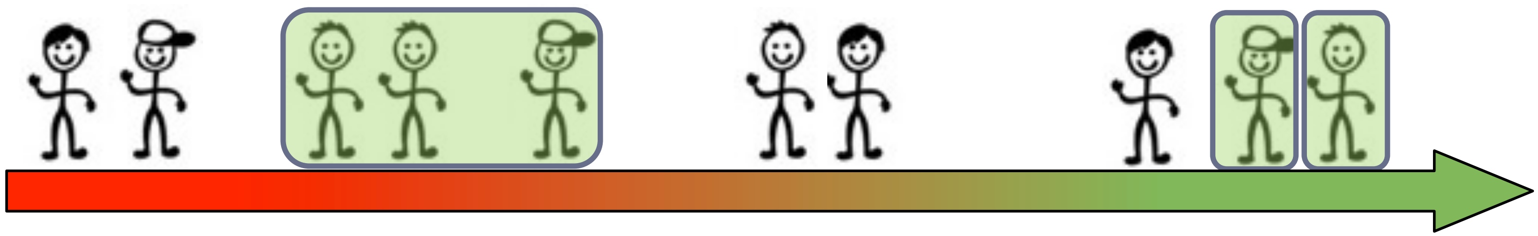
- Observation 1
  - Pick the best students



evimaria@cs.bu.edu

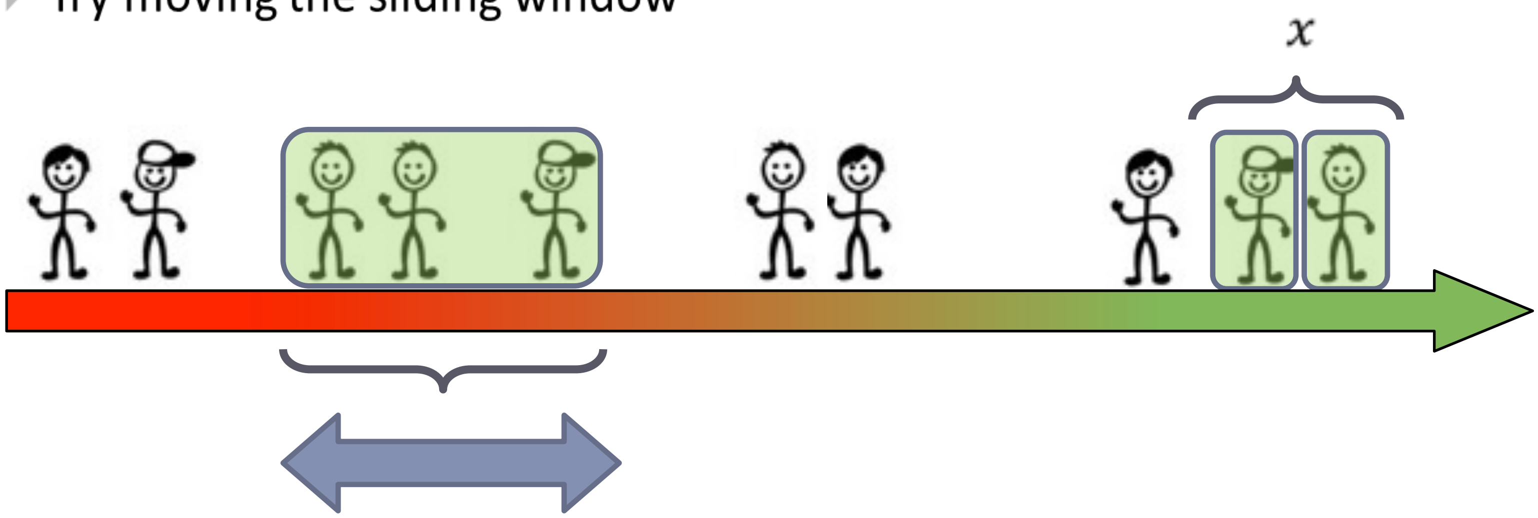
Finding the **best** team

- Observation 2
  - The followers are consecutive



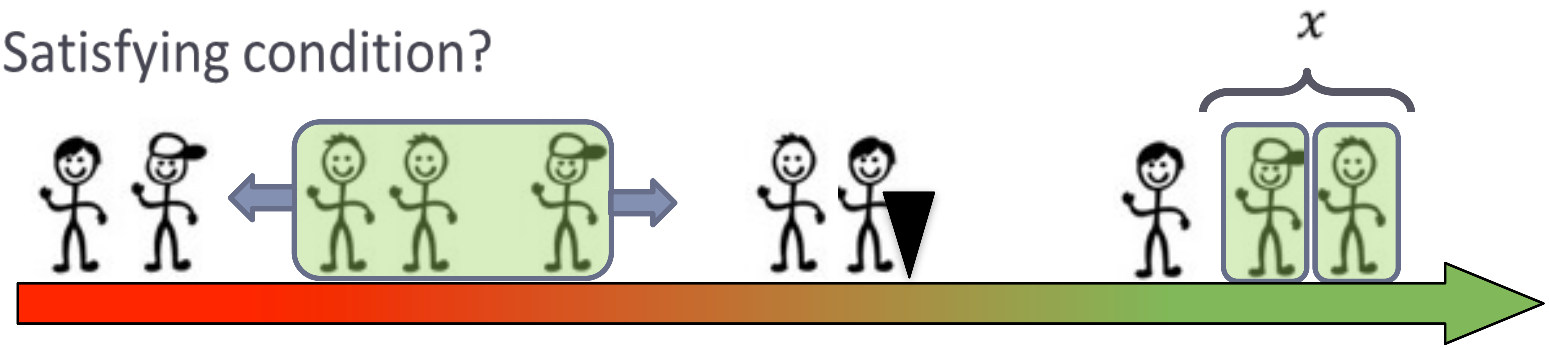
Finding the **best** team

- Algorithm
  - How many leaders?
    - Try all values of x (i.e., number of leaders)
  - Who are the followers?
    - Try moving the sliding window



Finding the **best** team

- Algorithm
  - How many leaders?
    - Try all values of x (i.e., number of leaders)
  - Who are the followers?
    - Try moving the sliding window
  - Satisfying condition?

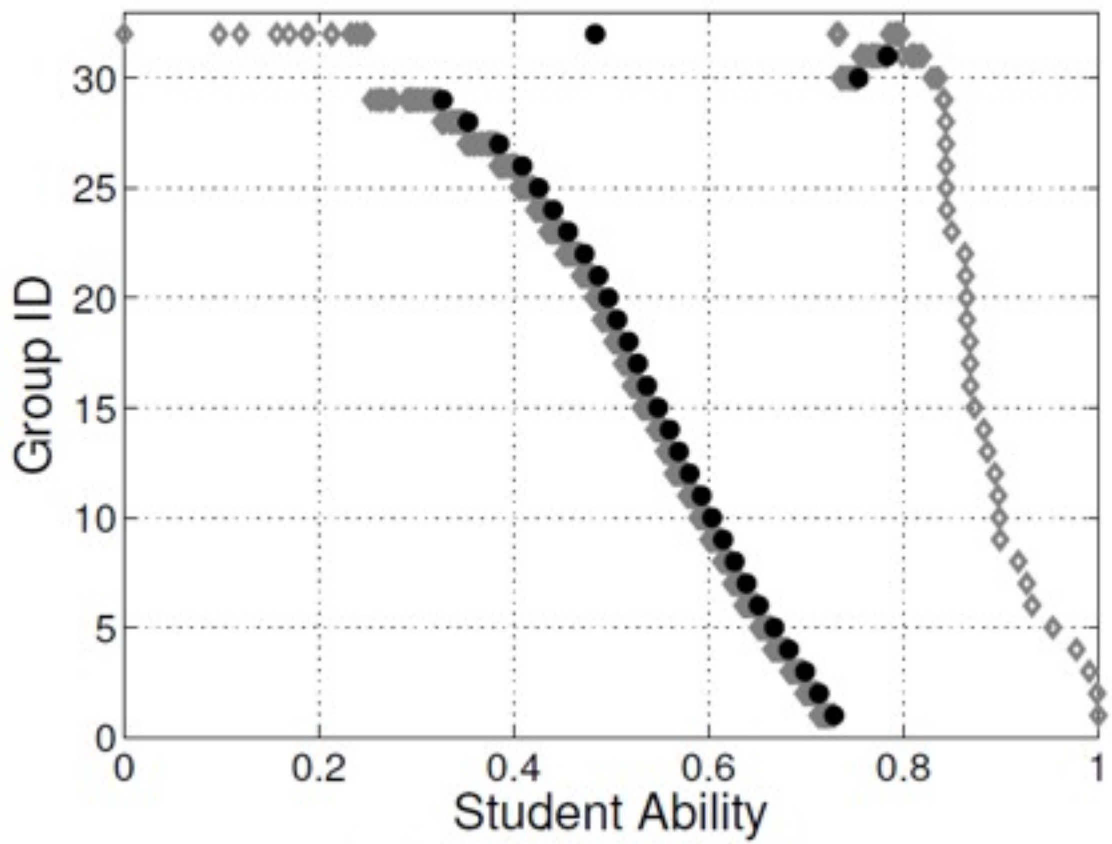


  - Test $O(n\log(k))$ groupings

evimaria@cs.bu.edu

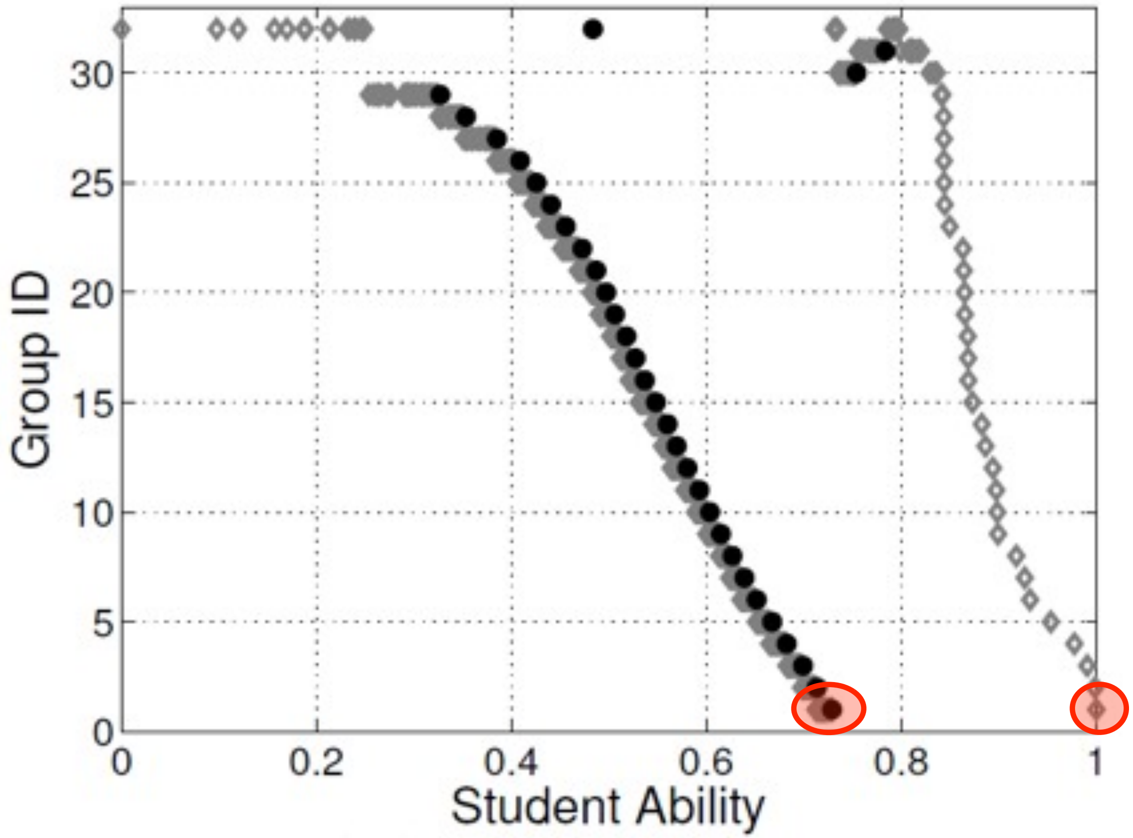
Results

- Grouping strong students with not much weaker students



Results

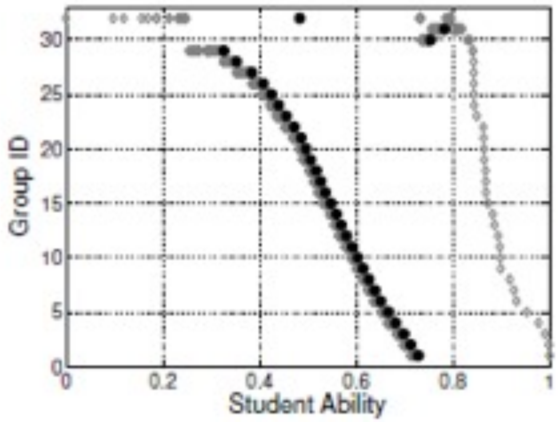
- Grouping strong students with not much weaker students



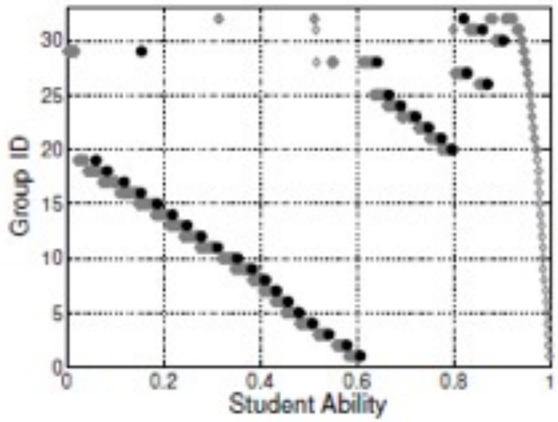
evimaria@cs.bu.edu

Results

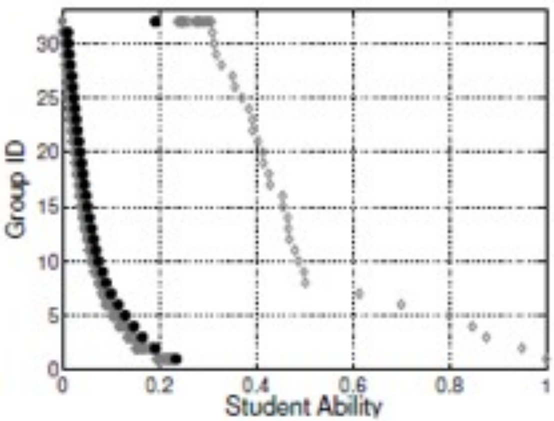
- Similar structure with different distributions of abilities



Normal Distribution



Uniform Distribution

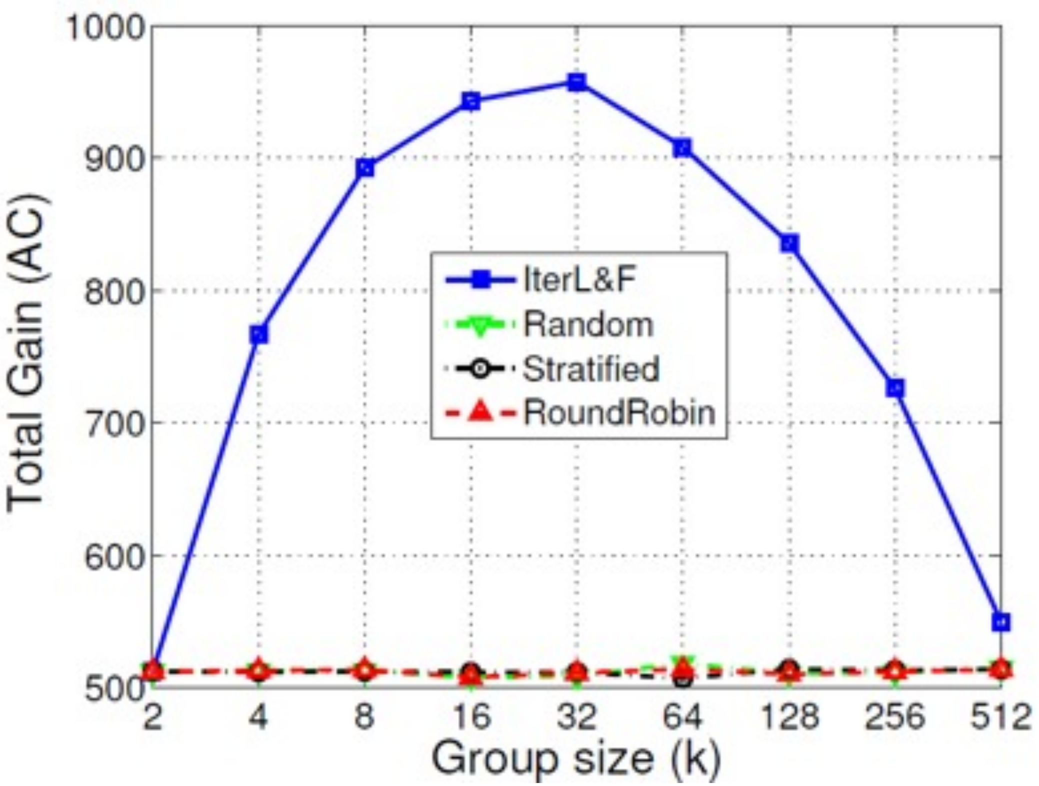


Pareto Distribution

BOSTON UNIVERSITY

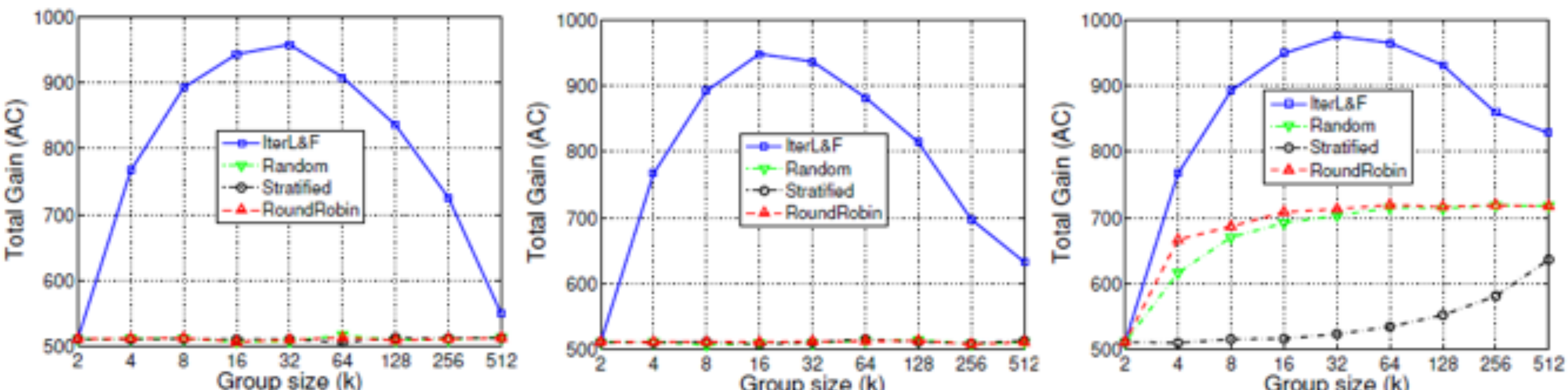
Results

- Classical methods are not optimal
  - With respect to our objective



Results

- Different distribution of student abilities



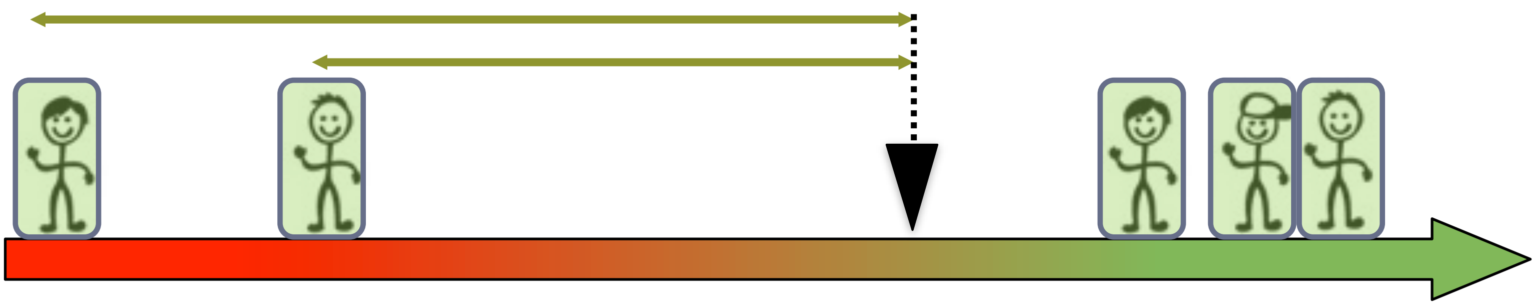
(a) *normal* dataset (b) *uniform* dataset (c) *pareto* dataset

General Framework

$$\mathcal{A}(T) = \sum_{i \in F_T} A_f(i,T) + \sum_{i \in L_T} A_\ell(i,T)$$

Gain Function — Gain (follower) — Gain (leader)

- Other Gain functions
  - How much do followers learn?
  - See the paper for more details



Summary of this part

- Traditional methods are not optimal
- Different objectives leads to different team structures
- Computation approaches can reveal such optimal structures
- Future Work
  - Richer gain functions
    - Gain for the leaders
    - Non-linear gain functions
  - Incorporating constraints due to socio-emotional factors

evimaria@cs.bu.edu

Overall summary

- Finding teams from a set of exerts
  - Organized in a network
  - Set Cover + Graph problems + Other online problems
- Inferring abilities from team performance
  - How about the chemistry of the team?
- Applications
  - Human resource management
  - (Online) educational settings (coursera, EdX, etc)

evimaria@cs.bu.edu

References

- [AGT'14] R. Agrawal, B. Golshan, E. Terzi: Grouping students in educational settings. ACM SIGKDD 2014
- [ABCGL'10] A. Anagnostopoulos,L. Becchetti,C. Castillo, A. Gionis, S. Leonardi: Power in Unity: Forming teams in large-scale community systems. CIKM 2010
- [ABCGL'12] A. Anagnostopoulos,L. Becchetti,C. Castillo, A. Gionis, S. Leonardi: Online team formation in social networks. WWW 2012
- [CSTC'12] C C. Cao, J. She, Y. Tong, L. Chen: Whom to ask? Jury selection for decision-making tasks on micro-blog services. VLDB 2012.
- [GS'12] A. Gajewar, A. D. Sarma: Multi-skill Collaborative Teams based on Densest Subgraphs. Siam Data Mining 2012
- [GLT'12] A. Gionis, T. Lappas, E. Terzi: Evaluating entity importance via counting set covers. ACM SIGKDD 2012
- [GLT'14] B. Golshan, T. Lappas, E. Terzi: Profit-maximizing cluster hires. ACM SIGKDD 2014
- [KA'11] M. Kargar and A. An: Discovering Top-k Teams of Experts with/without a Leader in Social Networks. CIKM 2011
- [LKT'09] T. Lappas, K. Liu, E. Terzi: Finding experts in social networks. ACM SIGKDD 2009
- [LS' 10] C-T Li, M-K. Shan: Team formation for generalized tasks in expertise social networks: IEEE SocialCom, 2010

Thanks

BOSTON UNIVERSITY

evimaria@cs.bu.edu