Online Team Formation under Different Synergies

Matthew Eichhorn, Sid Banerjee, David Kempe

WINE 2022



Paper (Extended): https://arxiv.org/abs/2210.05795

Motivation

Team formation is ubiquitous in many domains

- Online Education/Labor Platforms
- Projects need multi-agent teams
- "Success" of team depends on synergy of members
- Individuals' attributes that affect synergy are unknown

How should we group agents to efficiently find an optimal partition?

Setting

Agents: n individuals with unknown type $\theta_i \in \{0, 1\}$

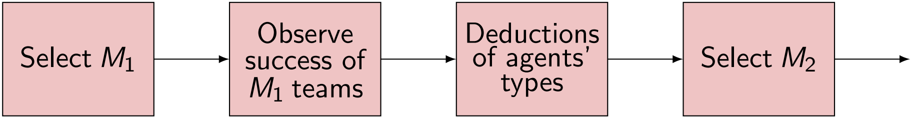
- k = number of type-1 agents (unknown)
- n, k even

Teams: Each team comprises 2 agents

- Success of team $\{i, j\}$ given by known symmetric synergy function $f : \{0, 1\}^2 \to \mathbb{R}$

Rounds: In round t, principal selects matching M_t

- Observes $f(\theta_i, \theta_j)$ for each $(i, j) \in M_t$



Objective

Score: $S(M) = \sum_{(i,j) \in M} f(\theta_i, \theta_i)$

Optimum: $M^* = \underset{M}{\operatorname{argmax}}\{S(M)\}$

Goal: Select matching policy to minimize *Total Regret*

$$\sum_{t=1}^{\infty} \Big(S(M^*) - S(M_t) \Big).$$

- Agents' types selected by an *adaptive* adversary
- Randomization will not help
- Optimal policy depends on team synergy function f

Reducing to Boolean Synergy Functions

- f is completely described by $f(0,0), \quad f(0,1) = f(1,0), \quad f(1,1)$
- Regret is linear in number of each type of team played

Lemma

Total regret under any symmetric f is determined by regret achieved under **boolean** *synergy functions $\{EQ, XOR, OR, AND\}$.*

Main Results

- All bounds are non-trivial and instance (k) dependent
- Algorithms are agnostic of k but achieve near-optimal regret for all k

EQ

Matching Lower/Upper Bound:

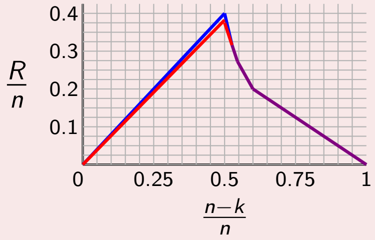
$$R = 2 \cdot \min(k, n-k)$$

XOR

Matching Lower/Upper Bound:

$$R = 2 \cdot \left(\min(k, n-k) - 1\right)^{+}$$

OR



AND

Lower Bound:

$$R \geq n - k$$

Upper Bound:

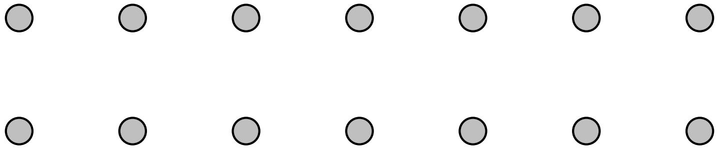
$$R \leq n - k + \left\lfloor \frac{\min(k, n-k)}{4} \right\rfloor$$

The EQ synergy function

- f(0,0) = f(1,1) = 1 f(0,1) = 0
- Promotes cohesive teams
- Optimal matching has as few $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "local search" strategy

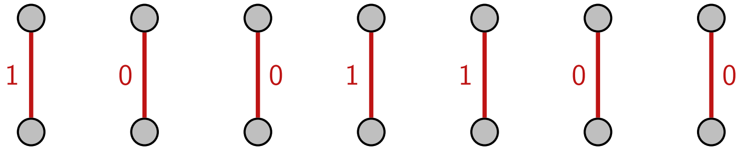
The EQ synergy function

- f(0,0) = f(1,1) = 1   f(0,1) = 0
- Promotes cohesive teams
- Optimal matching has as few $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a “local search” strategy



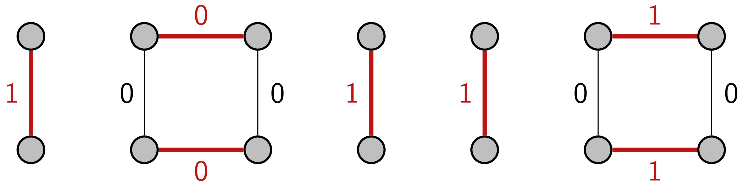
The EQ synergy function

- f(0,0) = f(1,1) = 1   f(0,1) = 0
- Promotes cohesive teams
- Optimal matching has as few $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a “local search” strategy



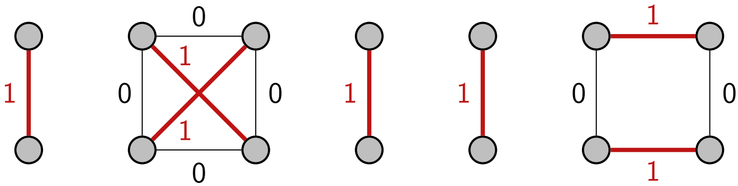
The EQ synergy function

- f(0,0) = f(1,1) = 1   f(0,1) = 0
- Promotes cohesive teams
- Optimal matching has as few $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "local search" strategy



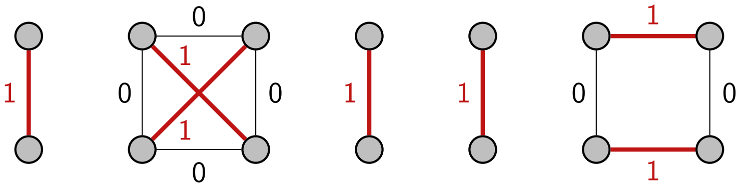
The EQ synergy function

- f(0,0) = f(1,1) = 1   f(0,1) = 0
- Promotes cohesive teams
- Optimal matching has as few $(0,1)$ teams as possible
- Can find M^* in 3 rounds with a "local search" strategy



The EQ synergy function

- f(0,0) = f(1,1) = 1   f(0,1) = 0
- Promotes cohesive teams
- Optimal matching has as few $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a “local search” strategy



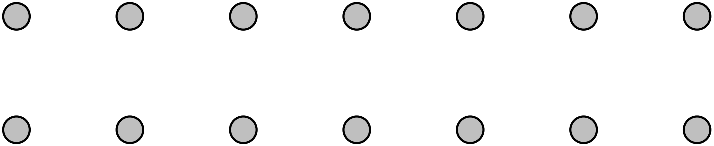
Total Regret $\leq 2 \cdot \min(k, n - k)$

The XOR synergy function

- f(0,0) = f(1,1) = 0   f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy

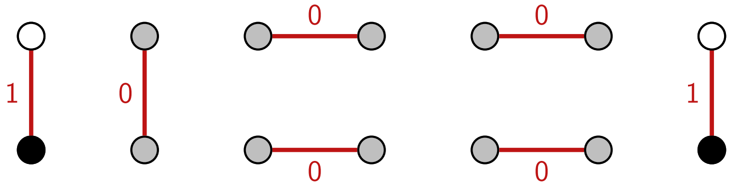
The XOR synergy function

- f(0,0) = f(1,1) = 0   f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



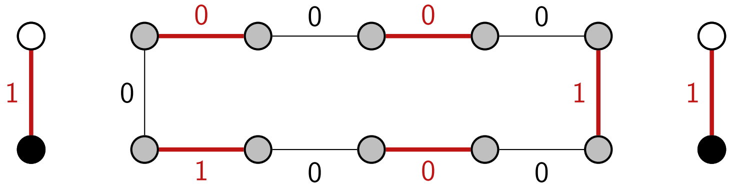
The XOR synergy function

- f(0,0) = f(1,1) = 0   f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



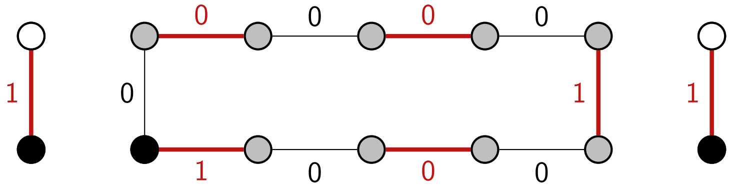
The XOR synergy function

- f(0,0) = f(1,1) = 0   f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a “global deduction” strategy



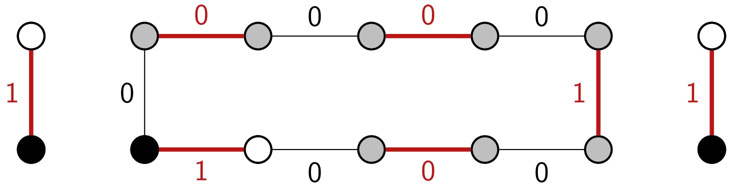
The XOR synergy function

- $f(0,0) = f(1,1) = 0 \quad f(0,1) = 1$
- Promotes diverse teams
- Optimal matching has as many $(0,1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



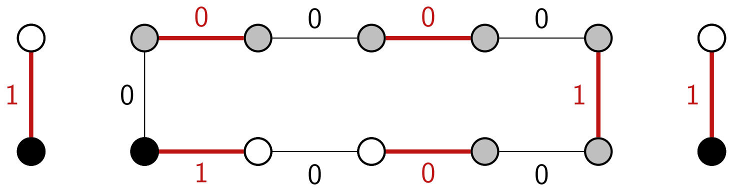
The XOR synergy function

- f(0,0) = f(1,1) = 0 f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



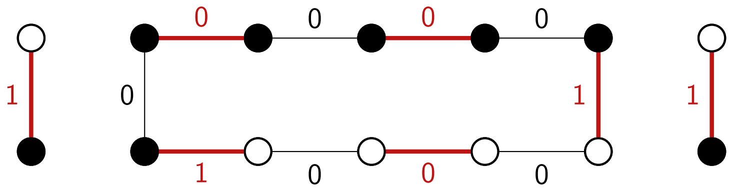
The XOR synergy function

- f(0,0) = f(1,1) = 0 f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



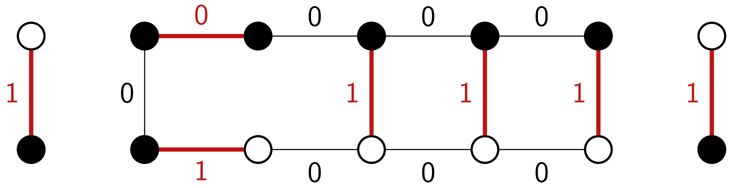
The XOR synergy function

- $f(0,0) = f(1,1) = 0 \quad f(0,1) = 1$
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



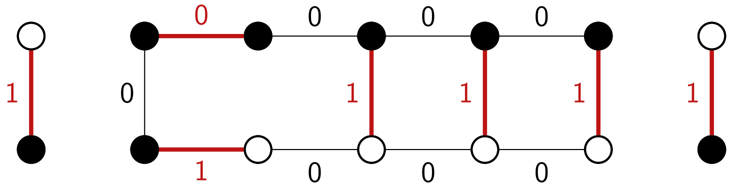
The XOR synergy function

- $f(0,0) = f(1,1) = 0 \quad f(0,1) = 1$
- Promotes diverse teams
- Optimal matching has as many $(0,1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



The XOR synergy function

- f(0,0) = f(1,1) = 0   f(0,1) = 1
- Promotes diverse teams
- Optimal matching has as many $(0, 1)$ teams as possible
- Can find M^* in 3 rounds with a "global deduction" strategy



Total Regret $\leq 2 \cdot \big(\min(k, n-k) - 1\big)^+$

The OR synergy function

- f(0,1) = f(1,1) = 1   f(0,0) = 0
- "Strongest Link Model" if θ_i encodes skill level
- Optimal matching has as many $(0, 1)$ teams as possible

The OR synergy function

- f(0,1) = f(1,1) = 1   f(0,0) = 0
- "Strongest Link Model" if θ_i encodes skill level
- Optimal matching has as many $(0,1)$ teams as possible

- Some $(0,0)$ teams revealed in Round 1
- Locate $(1,1)$ teams to re-partner with the $(0,0)$ teams
- Can't distinguish $(1,1)$ teams from $(0,1)$ teams

The OR synergy function

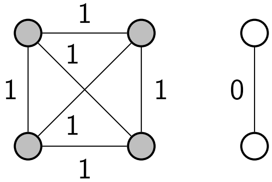
- f(0,1) = f(1,1) = 1   f(0,0) = 0
- "Strongest Link Model" if θ_i encodes skill level
- Optimal matching has as many $(0,1)$ teams as possible

- Some $(0,0)$ teams revealed in Round 1
- Locate $(1,1)$ teams to re-partner with the $(0,0)$ teams
- Can't distinguish $(1,1)$ teams from $(0,1)$ teams

- Use known 0-agents to "explore" types of unknown agents
- Can be slow, incur too much regret

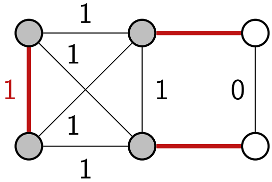
Key Algorithmic Insight

4-cliques of successful teams can quickly "neutralize" unsuccessful teams.



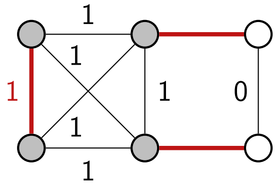
Key Algorithmic Insight

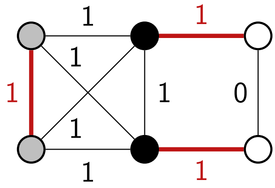
4-cliques of successful teams can quickly “neutralize” unsuccessful teams.



Key Algorithmic Insight

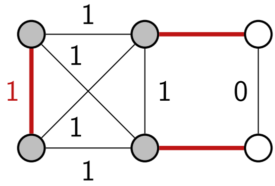
4-cliques of successful teams can quickly "neutralize" unsuccessful teams.

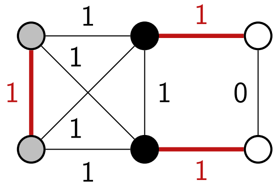




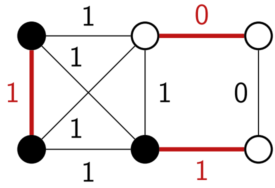
Key Algorithmic Insight

4-cliques of successful teams can quickly "neutralize" unsuccessful teams.





or

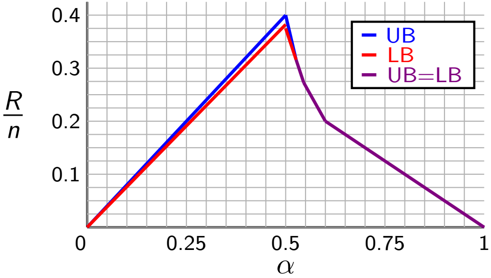


Max Exploit w/ 4-Cliques

Algorithm Idea: Explore with known 0-agents while re-pairing successful teams to form 4-cliques.

Regret Guarantee

- Parameterized by $\alpha = \frac{n-k}{k}$, fraction of low-skill agents
- Analysis is involved; different techniques for different ranges of α
- Nearly-matching upper/lower bounds that agree for $\alpha \geq \frac{10}{19}$



The AND synergy function

- f(0,0) = f(0,1) = 0   f(1,1) = 1
- "Weakest Link Model" if θ_i encodes skill level
- Optimal matching has as few $(0, 1)$ teams as possible

The AND synergy function

- f(0,0) = f(0,1) = 0   f(1,1) = 1
- "Weakest Link Model" if θ_i encodes skill level
- Optimal matching has as few $(0,1)$ teams as possible

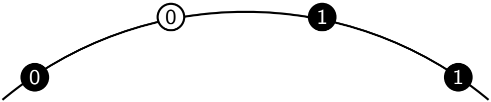
- Incur one unit of regret when we select two $(0,1)$ teams
- Must efficiently locate $(0,1)$ teams hidden among $(0,0)$ teams

The AND synergy function

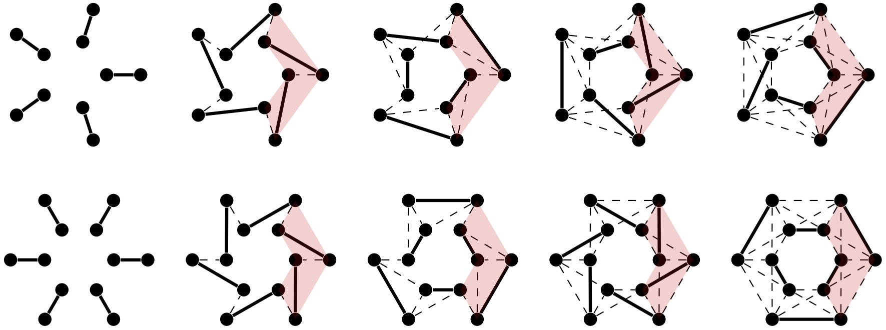
- f(0,0) = f(0,1) = 0   f(1,1) = 1
- "Weakest Link Model" if θ_i encodes skill level
- Optimal matching has as few $(0,1)$ teams as possible

- Incur one unit of regret when we select two $(0,1)$ teams
- Must efficiently locate $(0,1)$ teams hidden among $(0,0)$ teams

- **Idea:** "Charge" regret to 0-agents when they pair with 1-agent
- Place agents in big ring, pair with neighbors at increasing distances
- Before 0-agents pairs with 3 1-agents, two will have been paired up



- Must use "double ring" structure to deal with parity issues
- Case-specific repair step when $(1, 1)$ teams discovered



- Total Regret $\leq n - k + \left\lfloor \frac{\min(n-k,k)}{4} \right\rfloor$

Open Directions

- Can the gaps in our regret bounds be closed/tightened?
- We assume perfect feedback. What if signal of team success is noisy?
- How can we handle larger teams?

Thanks for Listening

For more details, please see our paper!