Diversity Considerations in Team Formation Design, Algorithm, and Measurement



Bowen Hui, Opey Adeyemi, Kiet Phan,
Justin Schoenit, Seth Akins, Keyvan Khademi



Computer Science, University of British Columbia

Most Memorable Team Experience

- Undergrad software engineering project with 8 students
  - About half A/B-students and half low-performing
  - Taught me the pains of working in large self-managed teams
  - Always wondered what criteria the prof used
- Fast forward 15 years later
  - As a new professor who fumbled into team-based learning
  - How should I form students into teams?
    - Explored with student self-formed teams and strategic criteria
    - Advice from colleagues to diversify team skills and gender

Collaborative Learning Context

- **Team formation task**: assign *all* students into non-overlapping groups
    - *NP-hard* problem

Collaborative Learning Context

- **Team formation task**: assign *all* students into non-overlapping groups
    - *NP-hard* problem
- Common strategies:
    - **Random teams** - can generate unbalanced teams that result in disproportionate individual participation
    - **Self-Assembled teams** - can cause discrimination among students with poor social relationships

Collaborative Learning Context

- **Team formation task**: assign *all* students into non-overlapping groups
  - *NP-hard* problem
- Common strategies:
  - **Random teams** - can generate unbalanced teams that result in disproportionate individual participation
  - **Self-Assembled teams** - can cause discrimination among students with poor social relationships
  - **Strategically formed teams** - which criteria? e.g., demographics, common time, social preferences, projects, diversify vs. concentrate

Collaborative Learning Context

- **Team formation task**: assign *all* students into non-overlapping groups
    - *NP-hard* problem
- Common strategies:
    - **Random teams** - can generate unbalanced teams that result in disproportionate individual participation
    - **Self-Assembled teams** - can cause discrimination among students with poor social relationships
    - **Strategically formed teams** - which criteria? e.g., demographics, common time, social preferences, projects, diversify vs. concentrate
- Goals:
    - Foster balanced interactions so students can maximize learning gains

Literature Overview

	Computer-Supported Collaborative Learning (CSCL)	AI	CS/Engineering Education
Research Questions	• Finds collaborative teams e.g. heterogeneous teams (no projects)	• Solves specific problem instances e.g. finds a best team	• Finds practical solution in classroom

Literature Overview

	Computer-Supported Collaborative Learning (CSCL)	AI	CS/Engineering Education
Research Questions	• Finds collaborative teams e.g. heterogeneous teams (no projects)	• Solves specific problem instances e.g. finds a best team	• Finds practical solution in classroom
Algorithms	• Ad hoc approaches e.g. genetic, search, etc. • Many learner characteristics	• Mostly game-theoretic and agent approaches • Abstract char.	• Varies case by case • Limited characteristics

Literature Overview

	Computer-Supported Collaborative Learning (CSCL)	AI	CS/Engineering Education
Research Questions	• Finds collaborative teams e.g. heterogeneous teams (no projects)	• Solves specific problem instances e.g. finds a best team	• Finds practical solution in classroom
Algorithms	• Ad hoc approaches e.g. genetic, search, etc. • Many learner characteristics	• Mostly game-theoretic and agent approaches • Abstract char.	• Varies case by case • Limited characteristics

Caution: Variable representation and distance calculation
e.g. Gender with 1 = woman, 2 = man, 3 = non-binary, etc.

Literature Overview

	Computer-Supported Collaborative Learning (CSCL)	AI	CS/Engineering Education
Research Questions	• Finds collaborative teams e.g. heterogeneous teams (no projects)	• Solves specific problem instances e.g. finds a best team	• Finds practical solution in classroom
Algorithms	• Ad hoc approaches e.g. genetic, search, etc. • Many learner characteristics	• Mostly game-theoretic and agent approaches • Abstract char.	• Varies case by case • Limited characteristics
Evaluation	• By demonstration • Some algorithm-specific measures • Minimal comparisons with other algorithms • Some learning effectiveness measures	• Mostly simulations (Strong use of metrics and benchmarking) • Some application in trivial instances	• By demonstration

Our Proposal: The Priority Algorithm

	Computer-Supported Collaborative Learning (CSCL)	AI	CS/Engineering Education
Research Questions	• Finds collaborative teams e.g. heterogeneous teams (no projects) **+ general**	• Solves specific problem instances e.g. finds a best team	• Finds practical solution in classroom
Algorithms	• Ad hoc approaches e.g. genetic, search, etc. • Many learner characteristics **+ unbiased**	• Mostly game-theoretic and agent approaches • Abstract char.	• Varies case by case • Limited characteristics
Evaluation	• By demonstration • Some algorithm-specific measures • Minimal comparisons with other algorithms • Some learning effectiveness measures	• Mostly simulations (Strong use of metrics and benchmarking) • Some application in trivial instances **+**	• By demonstration **diversity metrics** (heterogeneity, inter-homogeneity)

Diversity in Teams

- Many educators agree that **team diversity** is important
- Conflicting results that diversity has on team outcomes and how diversity is defined
- Gender-diverse and racial-diverse teams often result in more conflict where minoritized members are:
  - Confronted with microaggressions
  - Perceived as less skillful than peers in homogeneous teams
  - Treated with bias
    - not heard, not given leadership roles, pressured to change behaviors
- Problems are exacerbated when minorities are **tokenized**

Our Proposal: The Priority Algorithm

	Computer-Supported Collaborative Learning (CSCL)	AI	CS/Engineering Education
Research Questions	• Finds collaborative teams e.g. heterogeneous teams (no projects) + **general** + **min. tokenism**	• Solves specific problem instances e.g. finds a best team	• Finds practical solution in classroom
Algorithms	• Ad hoc approaches e.g. genetic, search, etc. • Many learner characteristics + **unbiased**	• Mostly game-theoretic and agent approaches • Abstract char.	• Varies case by case • Limited characteristics
Evaluation	• By demonstration • Some algorithm-specific measures • Minimal comparisons with other algorithms • Some learning effectiveness measures	• Mostly simulations (Strong use of metrics and benchmarking) • Some application in trivial instances	• By demonstration + **diversity metrics** + **solo status**

Schematic View of the Priority Algorithm

Visualizing the **solution space**

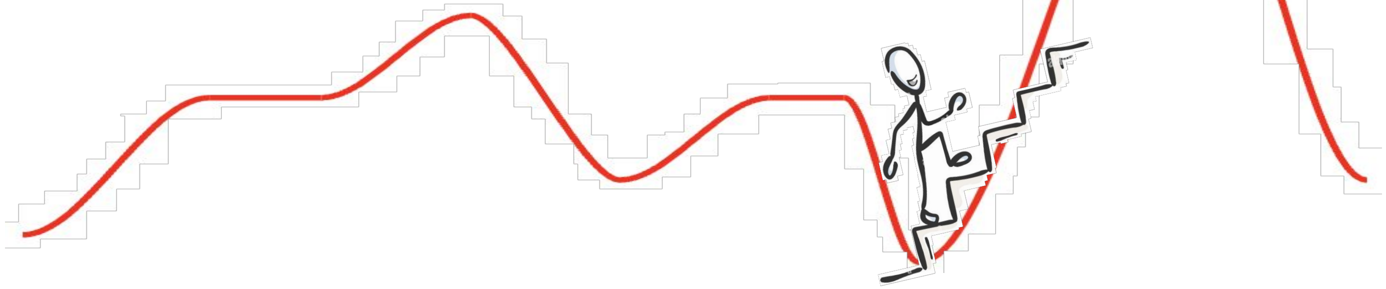
- Objective function maximizes solution criteria

Schematic View of the Priority Algorithm

Standard **hill climbing** approach

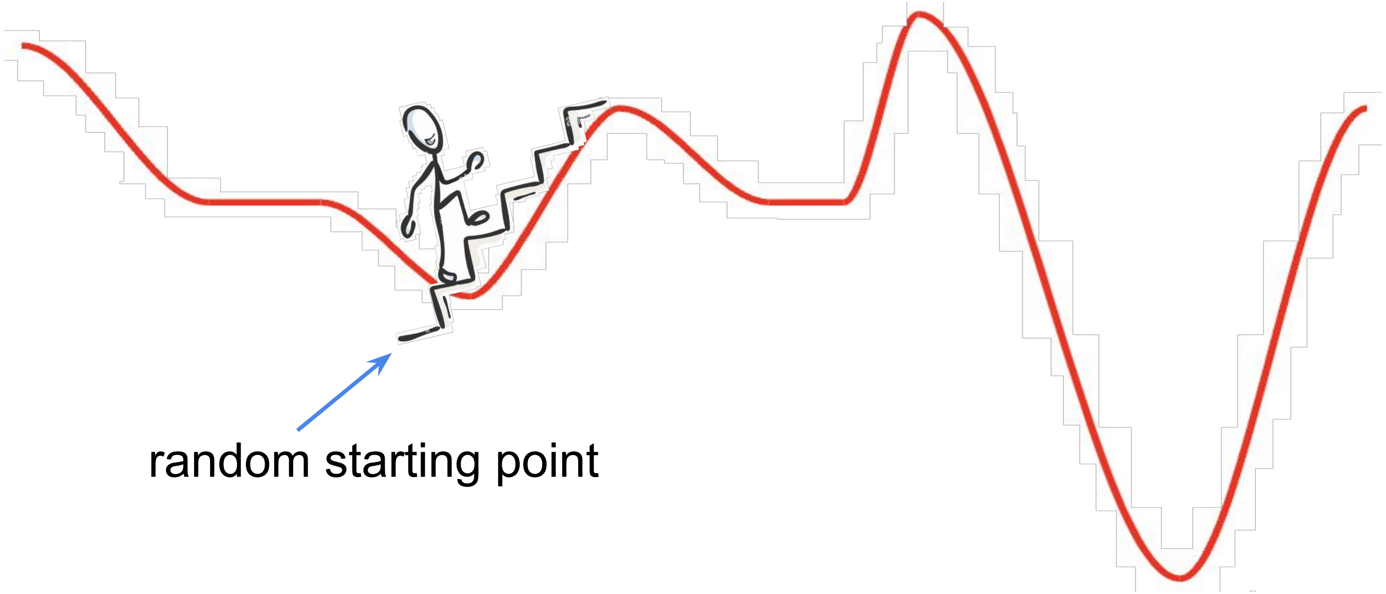
- Objective function handles multiple types of team formation criteria
  e.g. diversify without creating token minorities

eventually reach the optimal solution



random starting point

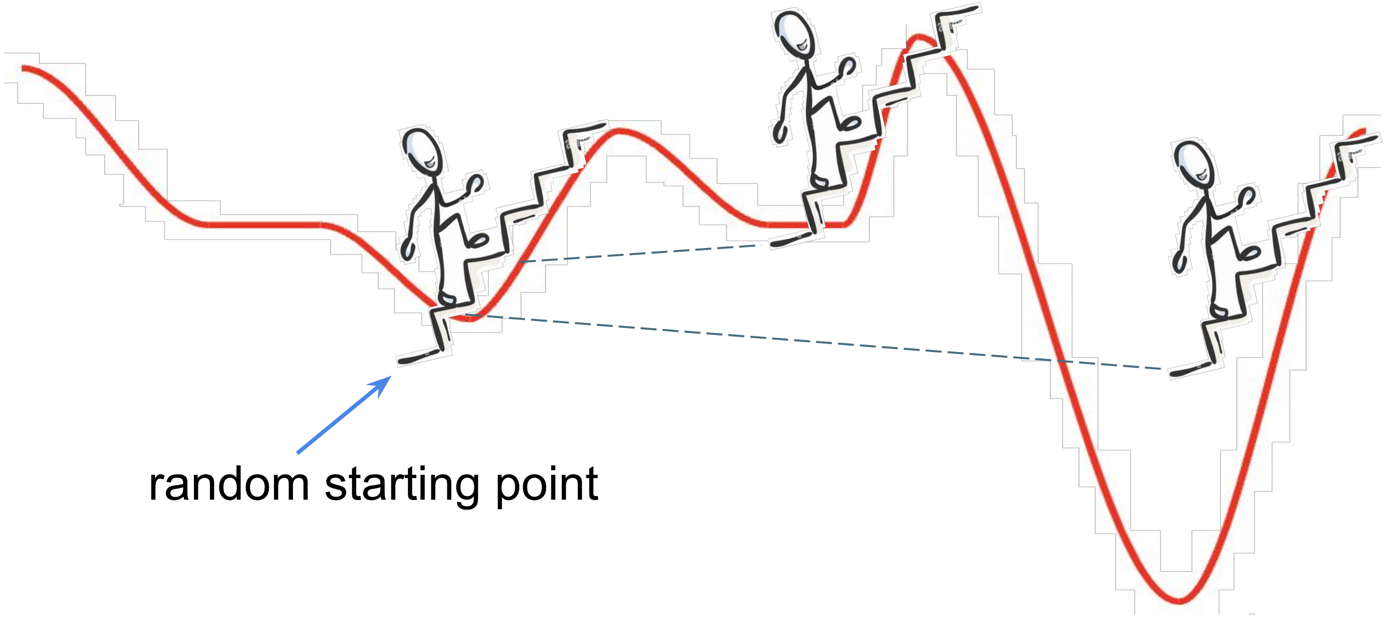
Schematic View of the Priority Algorithm



random starting point

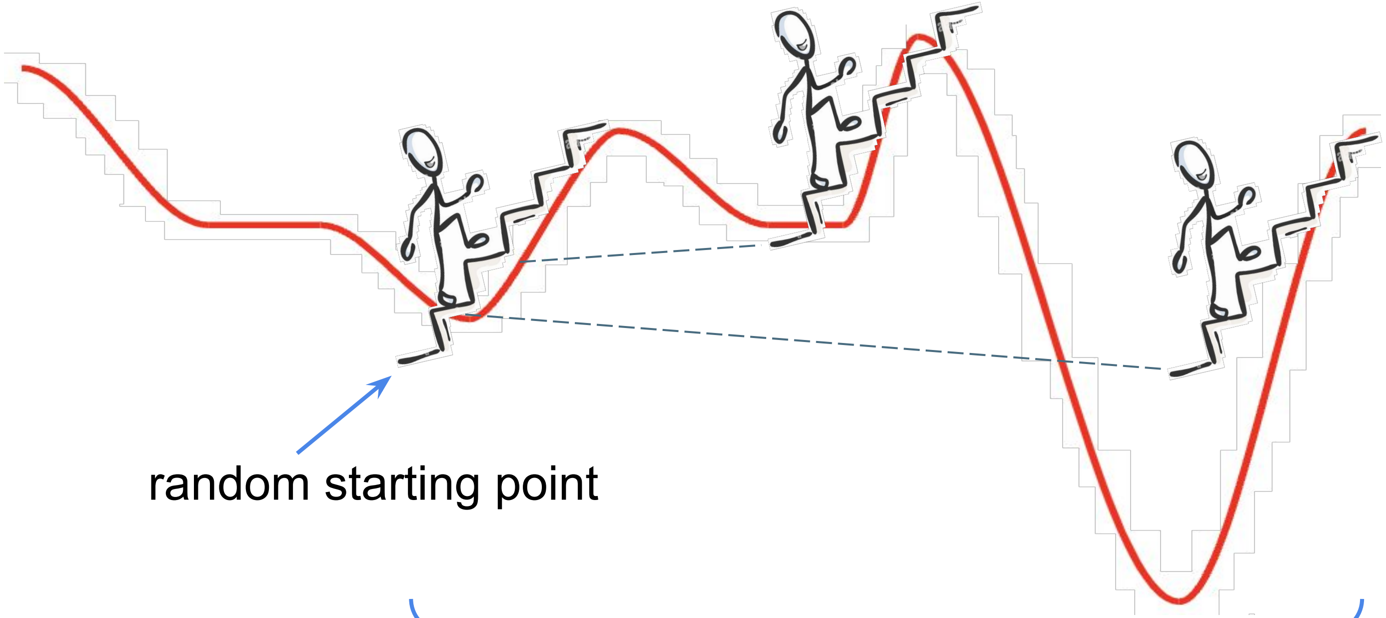
How to avoid local optimum?

Schematic View of the Priority Algorithm



Perform a "random swap" at each step

Schematic View of the Priority Algorithm



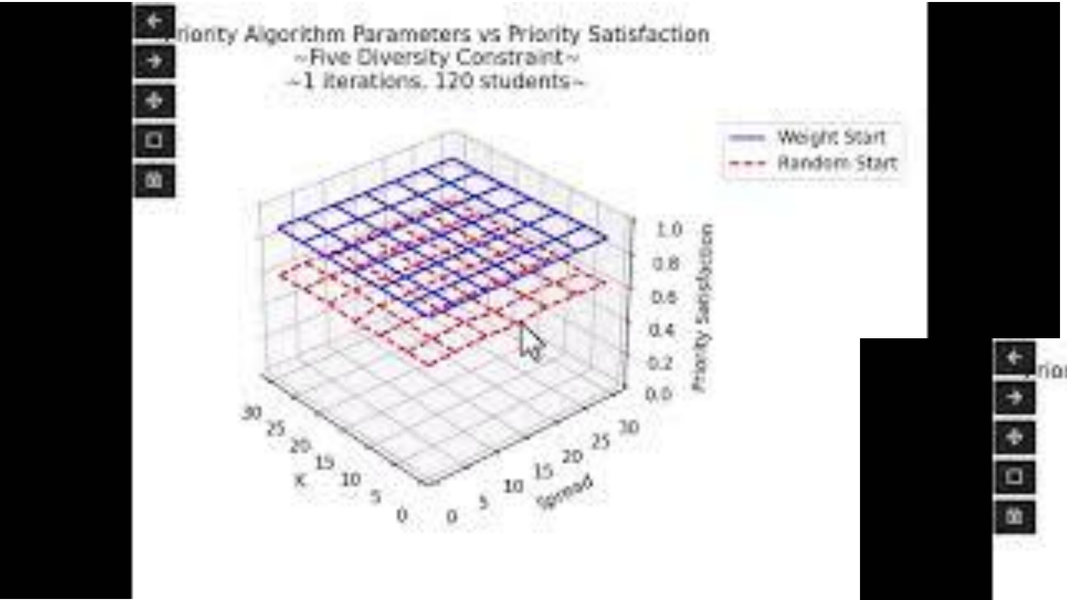
random starting point

MAXITER
= num
steps to
climb

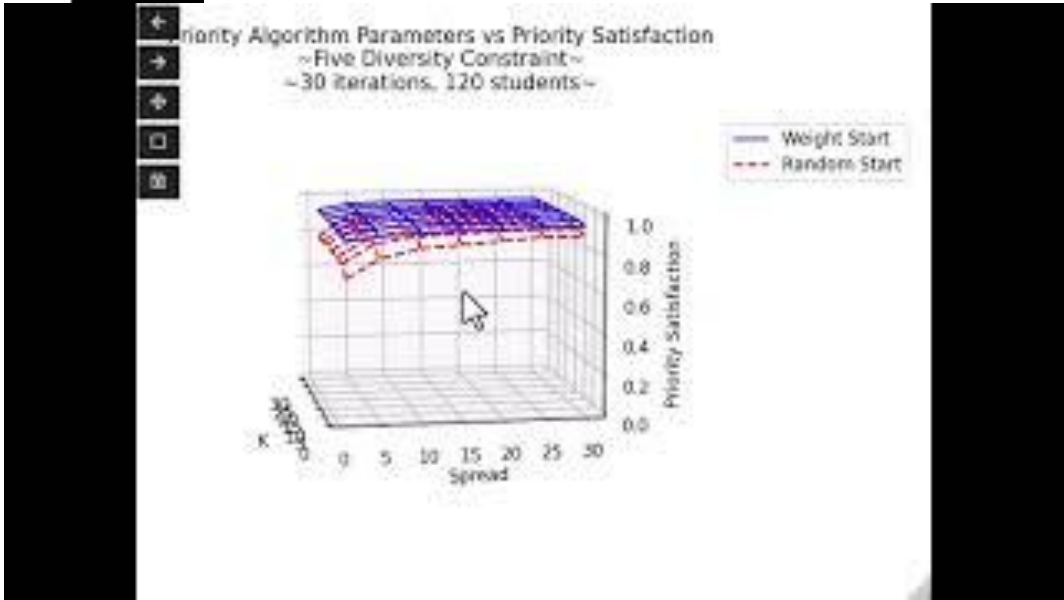
Spread
= num simultaneous paths
(at a given step)

Keep best **K** paths
(for computational efficiency)

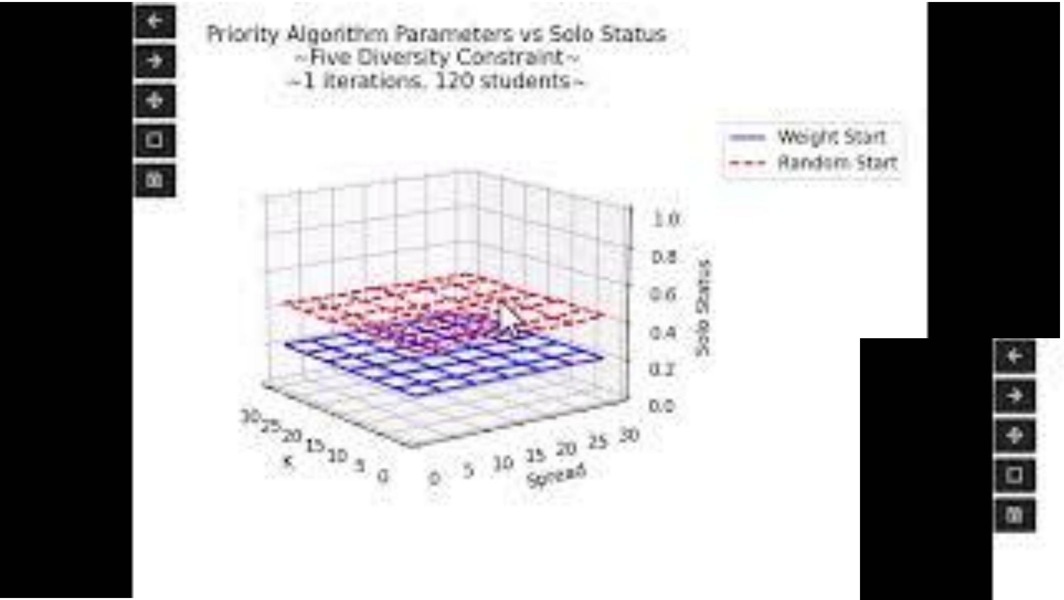
Understanding Algorithmic Behavior



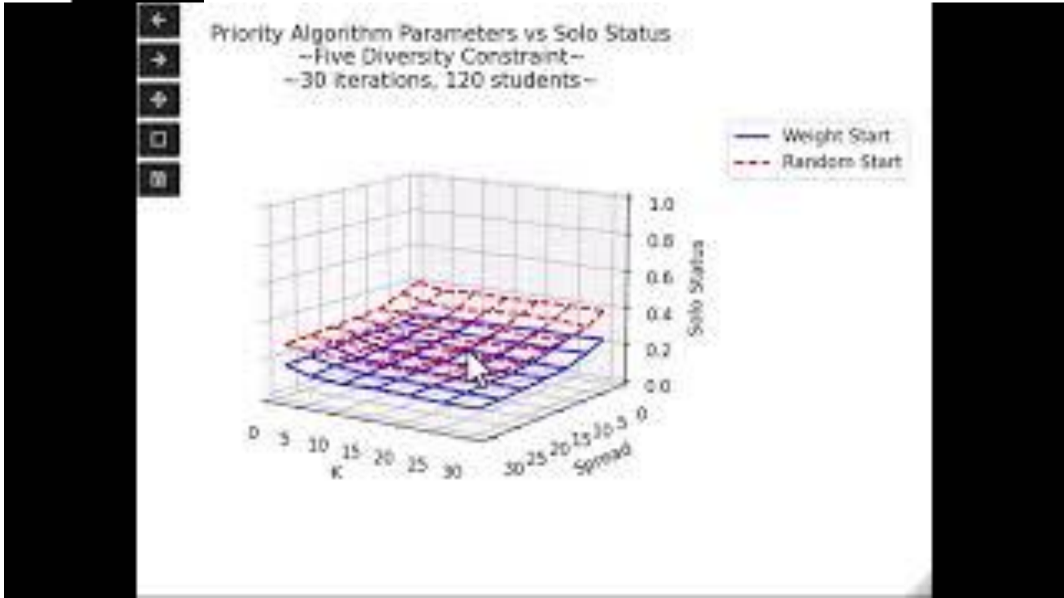
- Varying MAXITER, Spread, K
- Exploring 2 initial algorithms
- Measures priority satisfaction



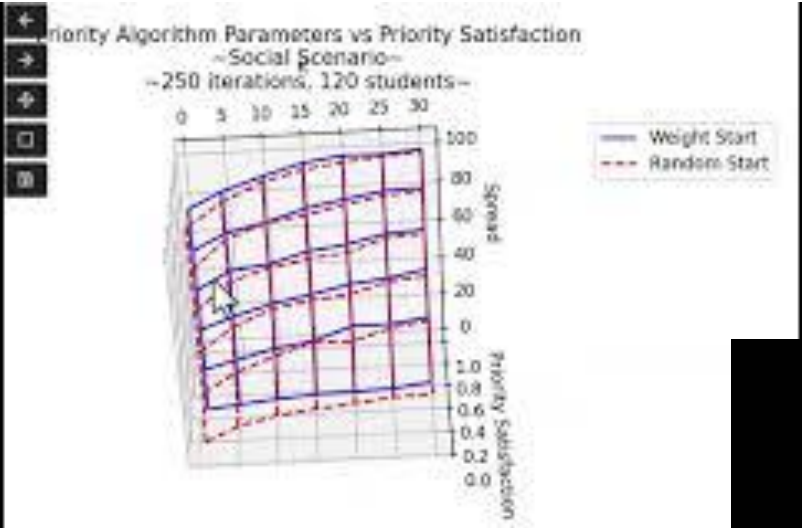
Understanding Algorithmic Behavior



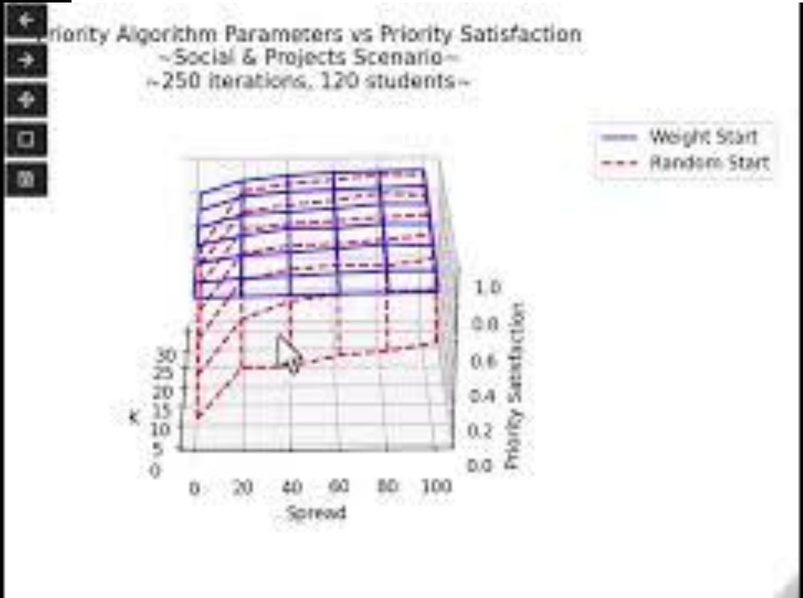
- Measures solo status



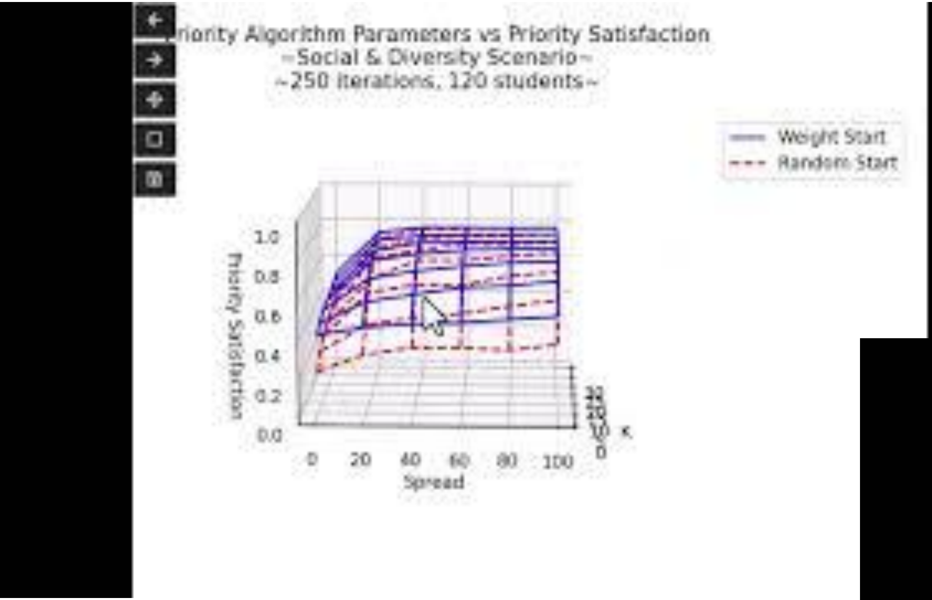
Understanding Algorithmic Behavior



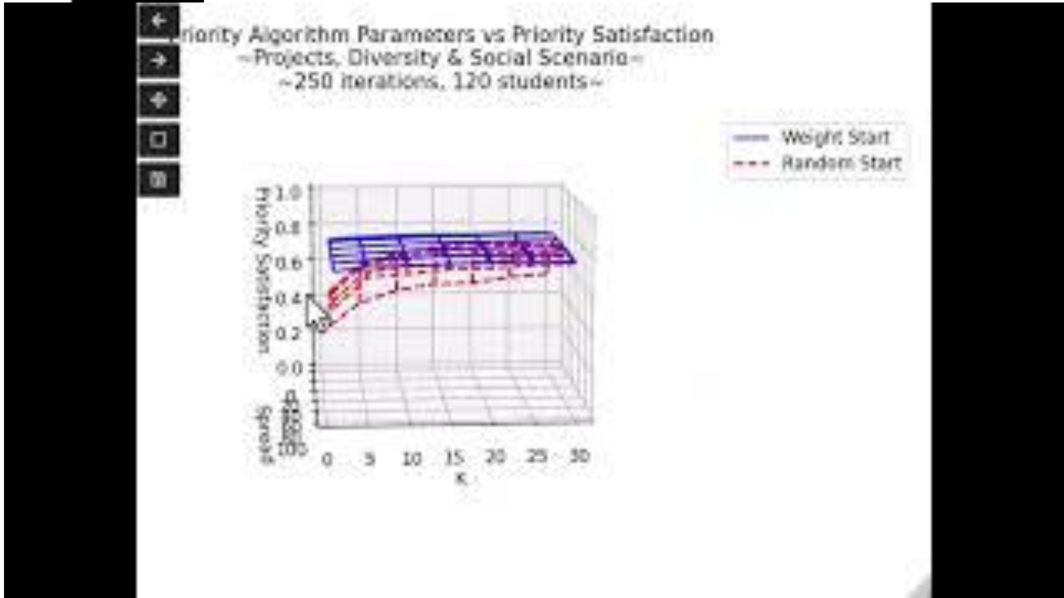
- Scenario with social preferences
- Only 1 mutual friend



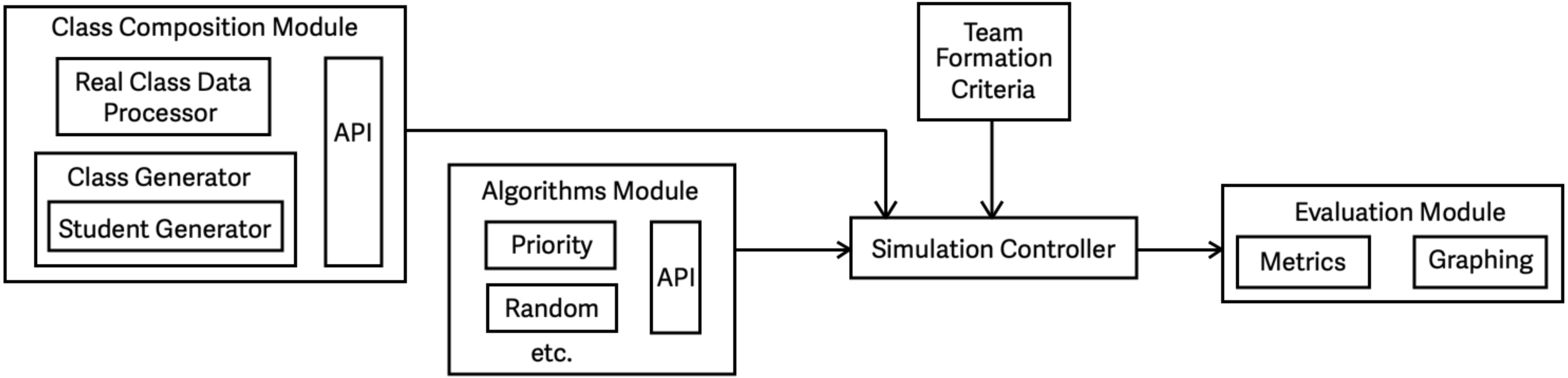
Understanding Algorithmic Behavior



- More complex social scenarios



Simulation Framework



- Modules:
  - Simulation Controller
  - **Inputs:** Team Formation Criteria, Class Composition, Algorithms
  - **Output:** Evaluation

Simulation Comparisons with Other Algorithms

- **Scenario 1:**
  - 1. Match project requirements
  - 2. Diversify females without tokenizing them
  - 3. Diversify African-descent without tokenizing them
  - Class composition:
    - 20% females, 80% males
    - 15% African background, 85% European backgrounds
    - 10-80% students can meet each requirement
    - 5 unique projects, 3-5 requirements each, duplicates to form teams of four for class sizes 20, 100, 240, 500, 1,000
- Scenario 2
- Simulation settings

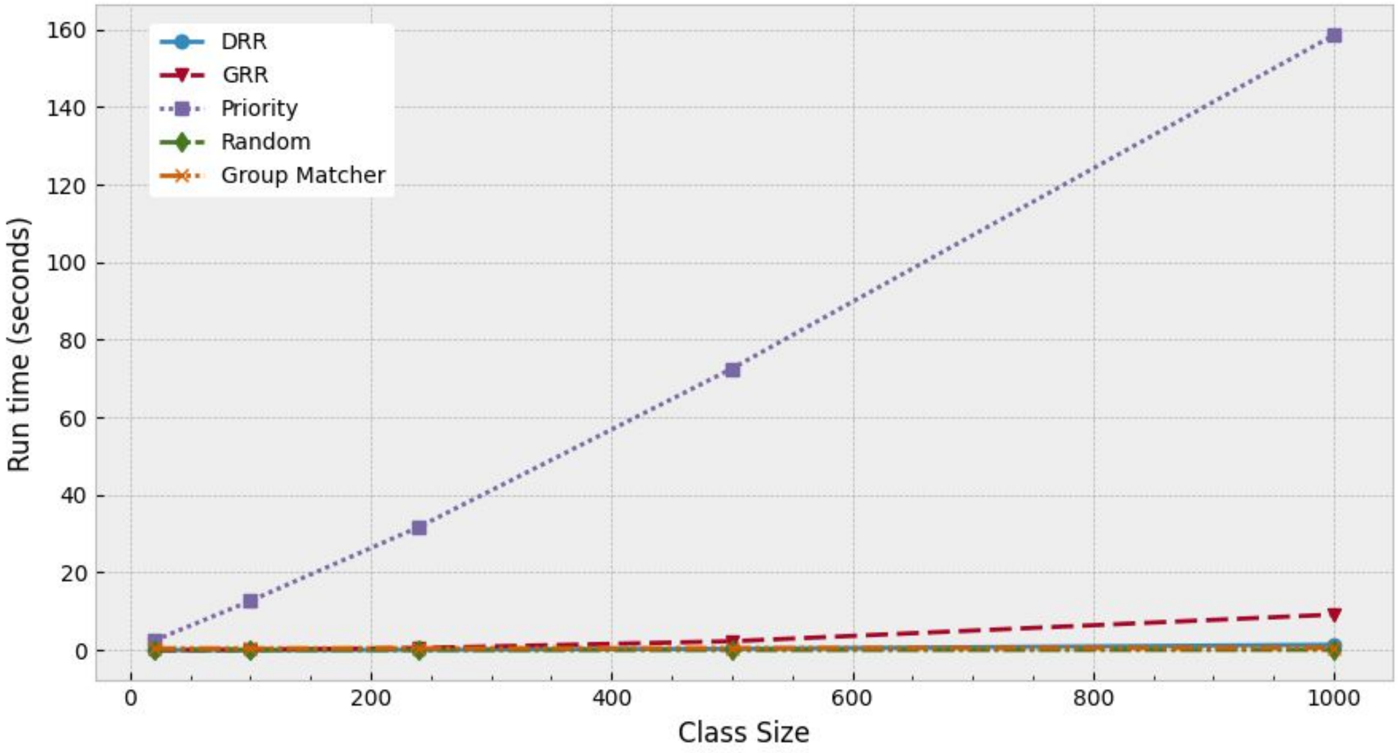
Simulation Comparisons with Other Algorithms

- Scenario 1
- **Scenario 2:**
  - 1. Concentrate on common time availabilities
  - 2. Diversify females without tokenizing them
  - 3. Diversify African-descent without tokenizing them
  - Class composition:
    - 10 timeslots and each student had 3-5 available times
    - Gender, cultural backgrounds, class sizes: same as before
- Simulation settings

Simulation Comparisons with Other Algorithms

- Scenario 1
- Scenario 2
- Simulation settings:
  - MAXITER=250, Spread=100, K=30, initial algorithm=weight
  - Results averaged over 100 trials
  - Comparison algorithms:
    - **Random**
    - **Double round robin (DRR)** - project matching
    - **Greedy round robin (GRR)** - more general purpose
    - **Group matcher** - mentoring based on time and tokenism
  - Metrics specific for each criterion

Simulation Comparisons with Other Algorithms



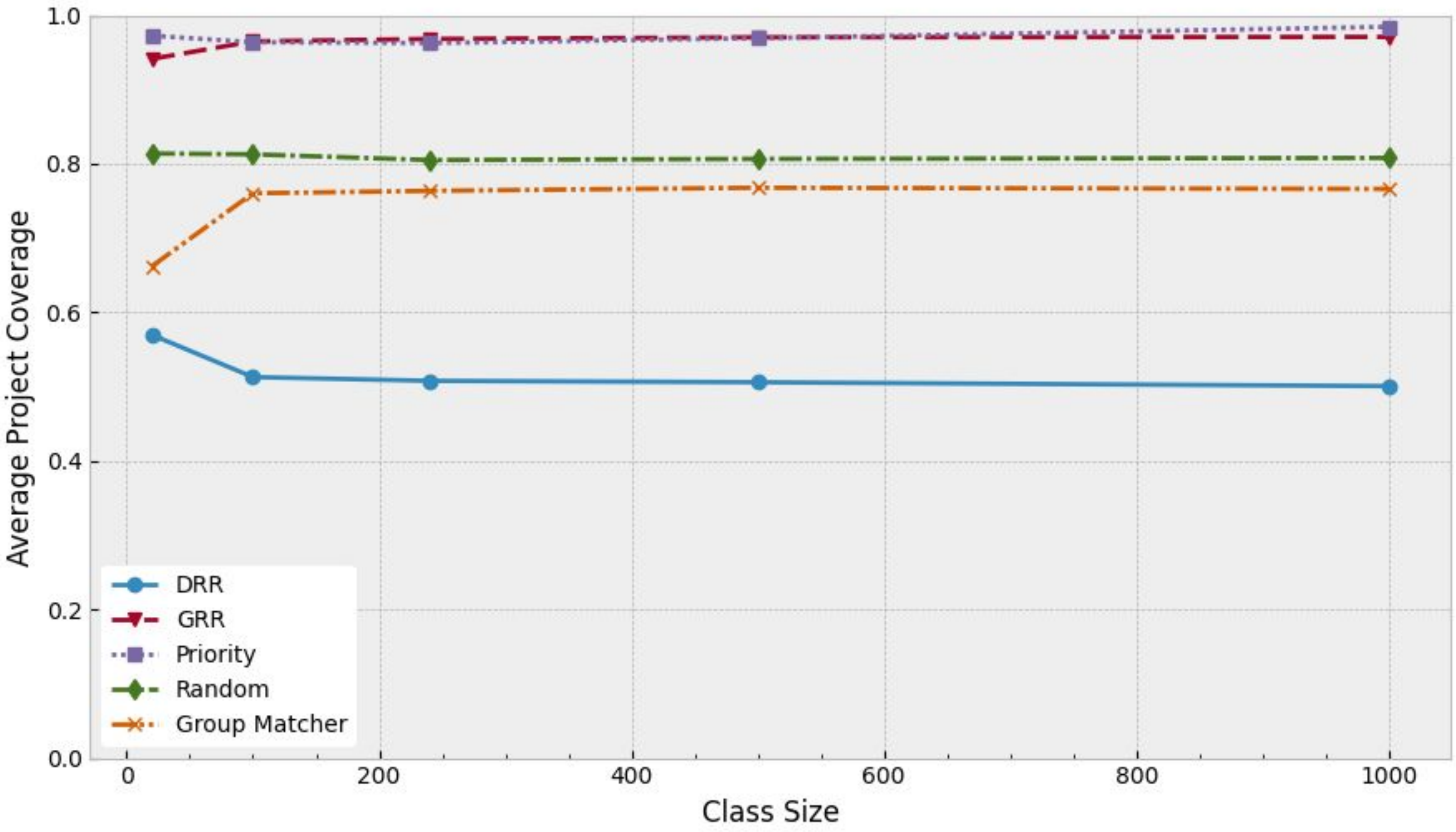
Manual teams reported 60+ hours for 240 students;

CSCL algorithms take 2 hours on small class sizes

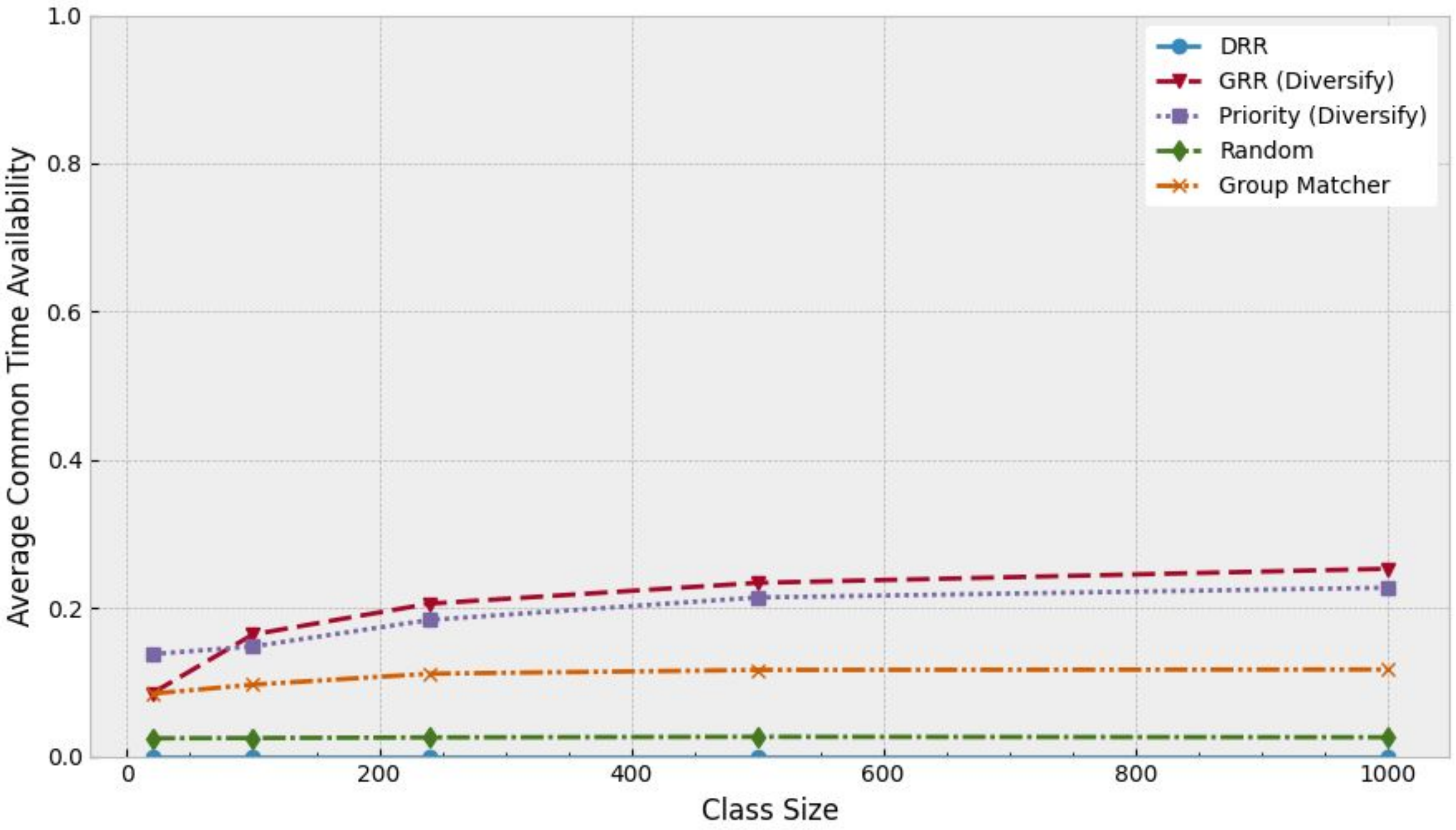
AI algorithms mostly applied to classes < 40

Simulation Comparisons with Other Algorithms

Excellent project coverage



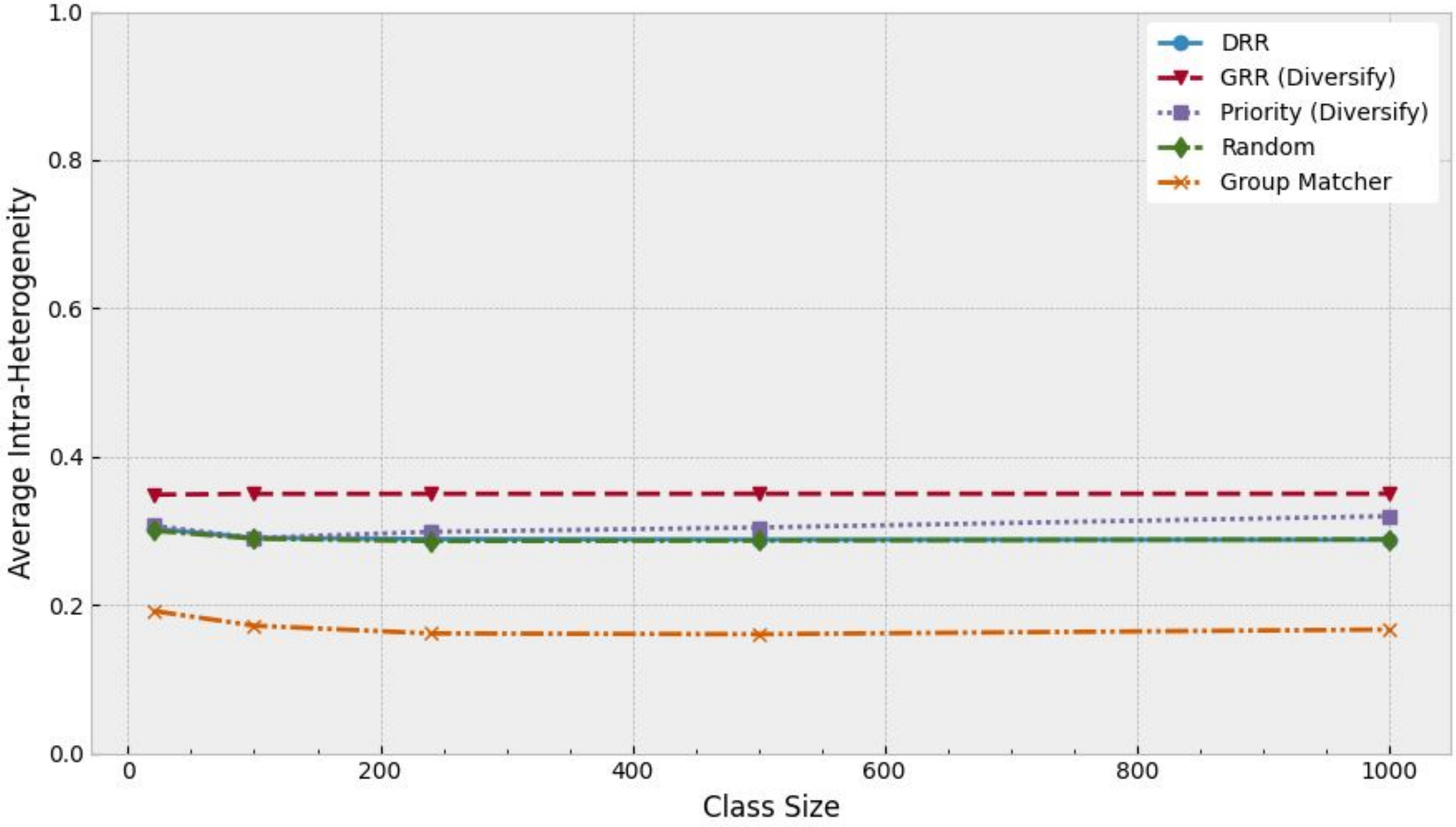
Simulation Comparisons with Other Algorithms



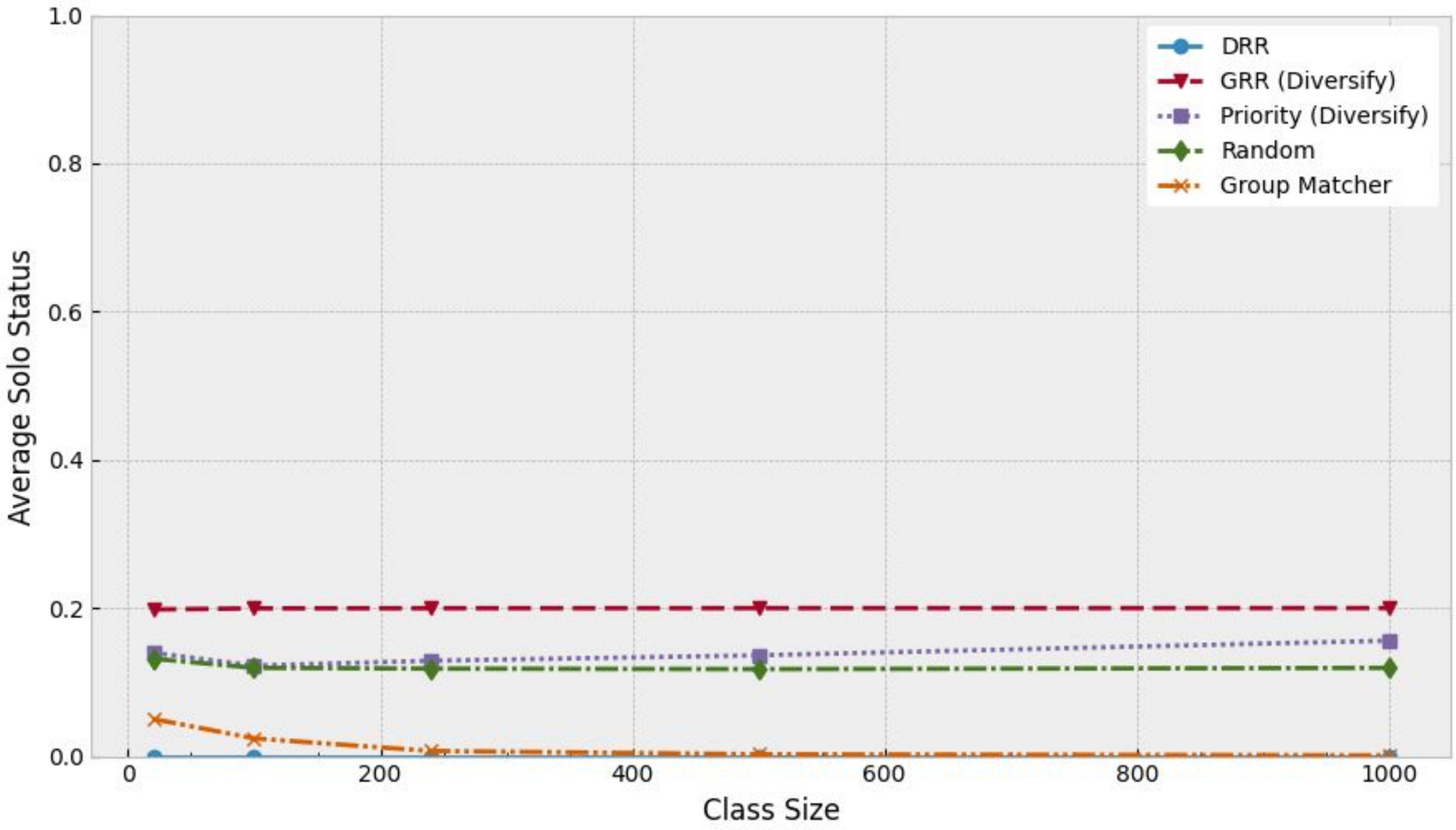
Better time matching than Group Matcher

Simulation Comparisons with Other Algorithms

More diverse than random



Simulation Comparisons with Other Algorithms



Performs worse on minimizing tokenism

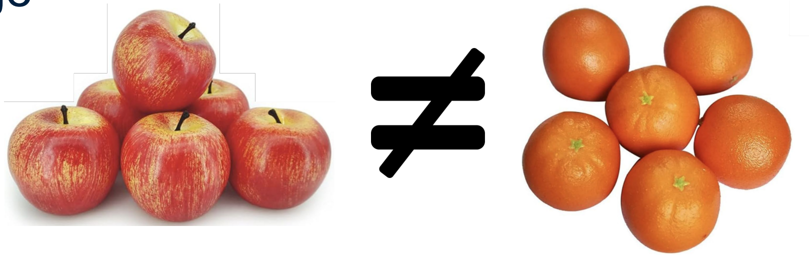
Insights from Simulation Analysis

- **Why DRR does so well on solo status in Scenario 2?**
  - Putting everyone into one team (nobody tokenized)

Insights from Simulation Analysis

- **Why DRR does so well on solo status in Scenario 2?**
    - Putting everyone into one team (nobody tokenized)

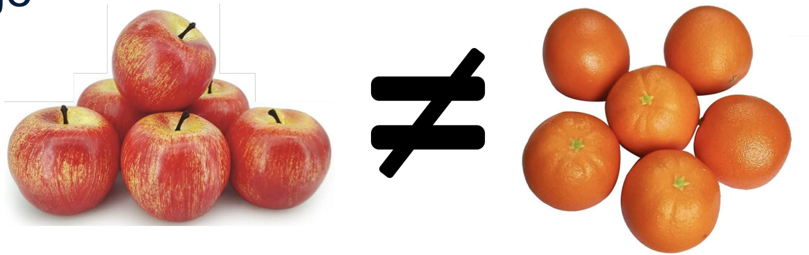
- **What is Group Matcher doing?**
    - Generating teams of 3-6 members (not teams of 4)
    - Generating fewer teams on average
    - Not trying to diversify at all



Insights from Simulation Analysis

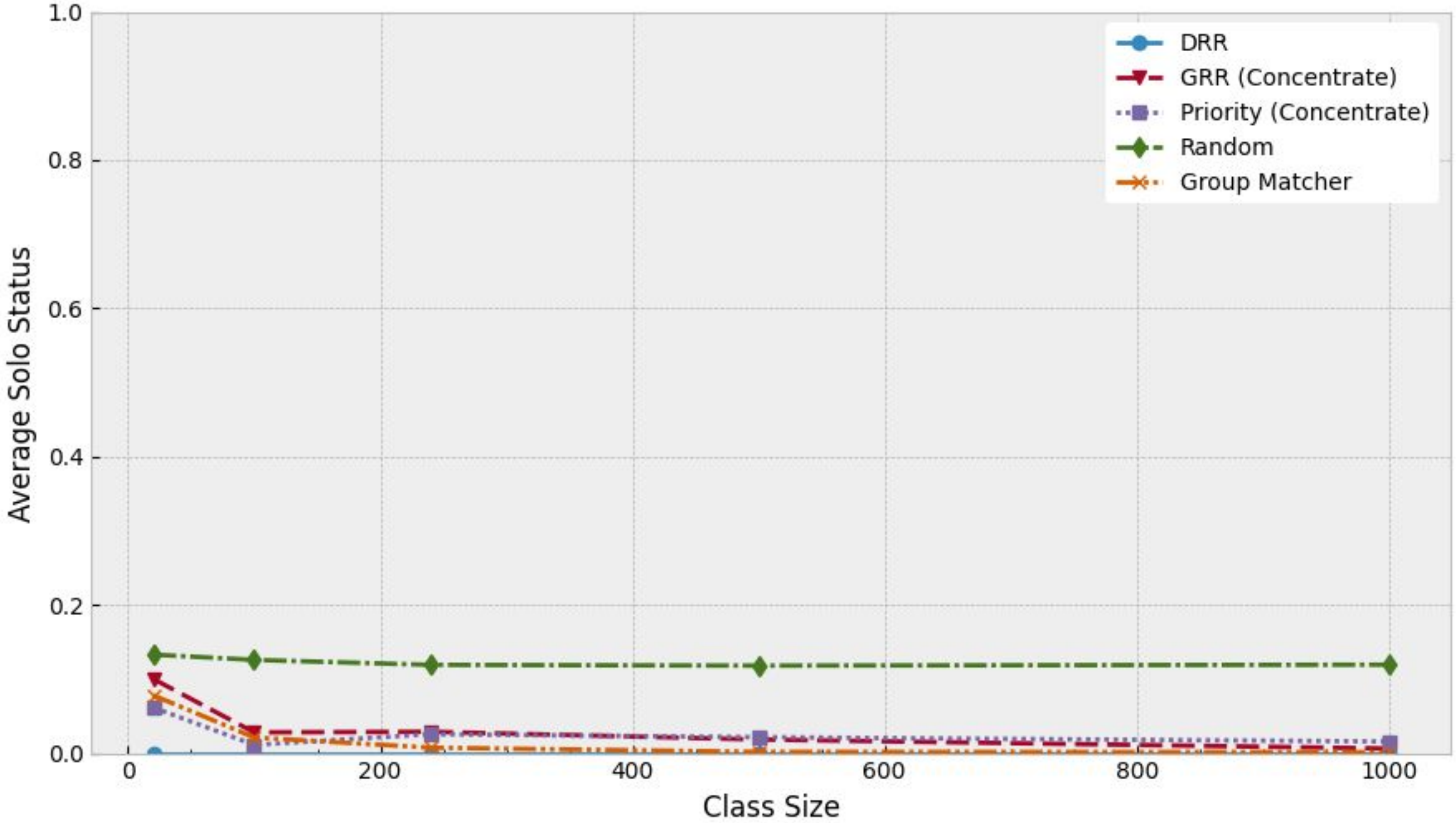
- **Why DRR does so well on solo status in Scenario 2?**
  - Putting everyone into one team (nobody tokenized)

- **What is Group Matcher doing?**
  - Generating teams of 3-6 members (not teams of 4)
  - Generating fewer teams on average
  - Not trying to diversify at all



- **What if Priority …**
  - Generating teams of 6?
  - *Concentrated* instead of diversify?

Simulation Comparisons with Other Algorithms



Becomes competitive with Group Matcher

Real Data Comparisons with Other Algorithms

- **HCI class:**
  - 215 students but only 175 responded
  - 168 undergrads, 7 grads
  - 37 females, 135 males, 2 non-binary, 1 prefer not to answer
  - 6 timeslots, with 31 to 138 students available in each slot

Table 1. Results for a scenario that considers common times, diversifies women but avoids tokenizing them, and diversifies graduate students. The best percent is bolded for each metric. GM is short of group matcher.

Class 1 Results	Random	GM	GRR	P4div	P6div	P4conc	P6conc
Common Times	11.4	**25.9**	23.9	23.1	17.8	23.5	18.9
Intra-Heterog. (Gender)	33.3	27.4	**45.8**	34.9	35.1	11.7	11.3
Inter-Homog. (Gender)	29.2	33.6	**14.9**	29.2	22.1	24.0	22.7
Solo Status (Women)	12.57	1.7	22.9	12.57	5.7	1.1	**0.6**
Intra-Heterog. (Year)	**7.9**	0.0	**7.9**	**7.9**	7.8	6.1	7.3
Inter-Homog. (Year)	18.5	**0.0**	18.5	18.5	14.3	17.3	15.3

Real Data Comparisons with Other Algorithms

- **Summer Capstone class:**
  - 9 client-sponsored projects, 2 duplicates, each with 2-5 requirements
  - 41 students, all responded
    - 4 A+'s, 16 A's, 18 B's, 3 D's, 0 F's (C's were forgotten)
    - 6 timeslots, with 13 to 33 students available in each slot
    - 15+ students could meet all but 1 req., nobody could meet last req.
    - each student has up to 3 friends and 3 enemies

Table 2. Results for a scenario that considers project requirements, diversifies GPA, concentrates time availabilities, and satisfies social preferences. The best percent score for each metric is bolded.

Class 2 Results	Random	DRR	Group Matcher	GRR	Priority (P4Div)
Project Coverage	**97.8**	87.6	**97.8**	**97.8**	**97.8**
Common Times	13.0	4.76	**33.3**	20.4	**33.3**
Intra-Heterog. (GPA)	70.0	61.2	66.3	71.9	**77.4**
Inter-Homog. (GPA)	15.4	14.2	12.3	12.6	**11.7**
Social Satisfaction	0.0	0.0	0.0	11.1	**44.4**

Conclusions

- Diversity considerations:
  - Representing demographic attributes in an unbiased way
  - New metrics for group diversity and token minorities
  - Approaches for formulating tokenism criteria
  -

Conclusions

- Diversity considerations:
  - Representing demographic attributes in an unbiased way
  - New metrics for group diversity and token minorities
  - Approaches for formulating tokenism criteria
- Specific contributions:
  - **Priority algorithm** is a new general-purpose team formation algorithm
    - Handles project requirements matching, social preferences, diversity constraints, tokenism
  - Simulation approach for algorithmic advancement

Contact: bowen.hui@ubc.ca