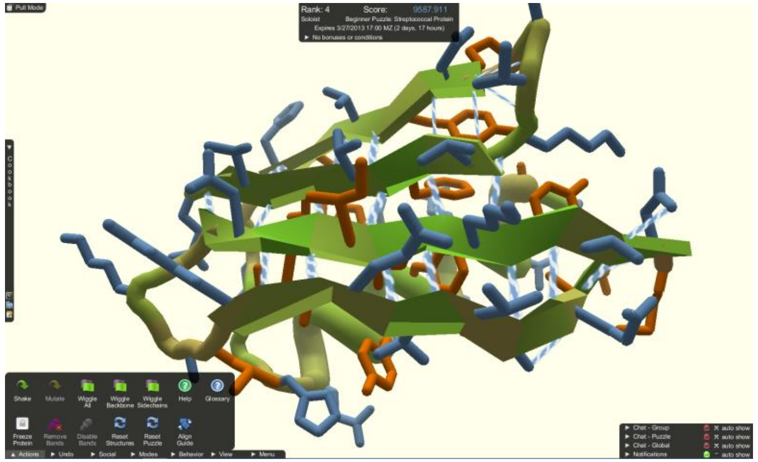
Ordering Levels in Human Computation Games using Playtraces and Level Structure

Anurag Sarkar and **Seth Cooper**

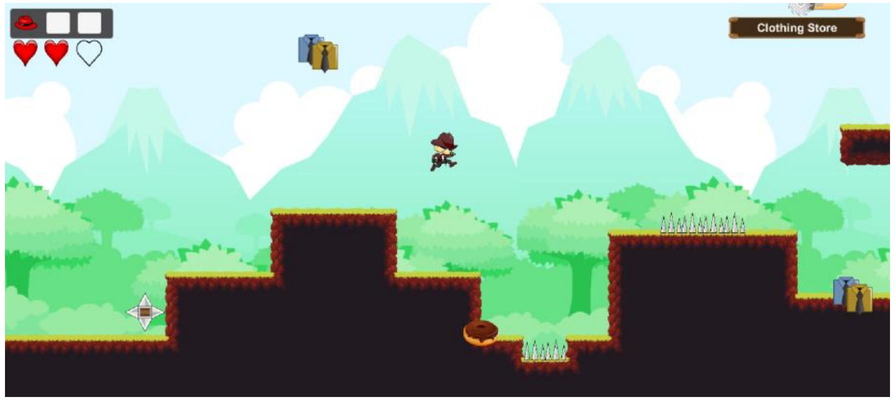
Northeastern University

Background

- Human computation games (HCGs) model real world problems to help solve them through players



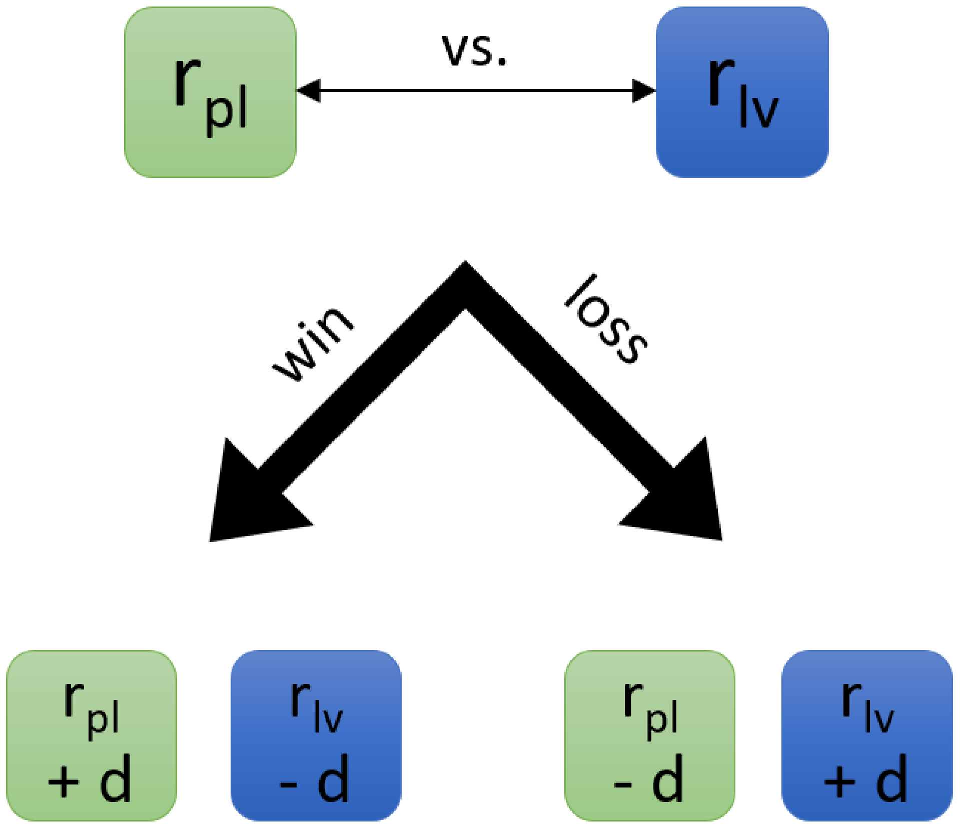
Foldit



Iowa James

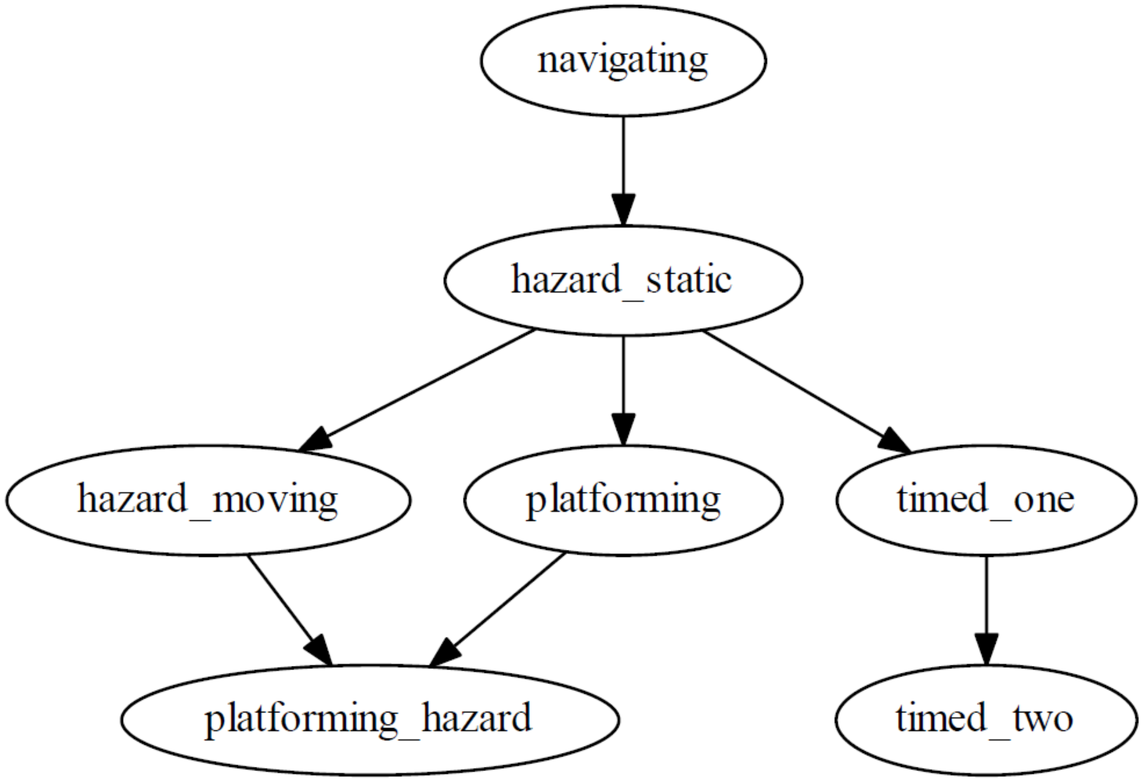
Background

- Human computation games (HCGs) model real world problems to help solve them through players
- Prior works did dynamic difficulty adjustment (DDA) in HCGs using rating systems, framing DDA as PvL matchmaking
  - Player rating → player ability
  - Level rating → level difficulty



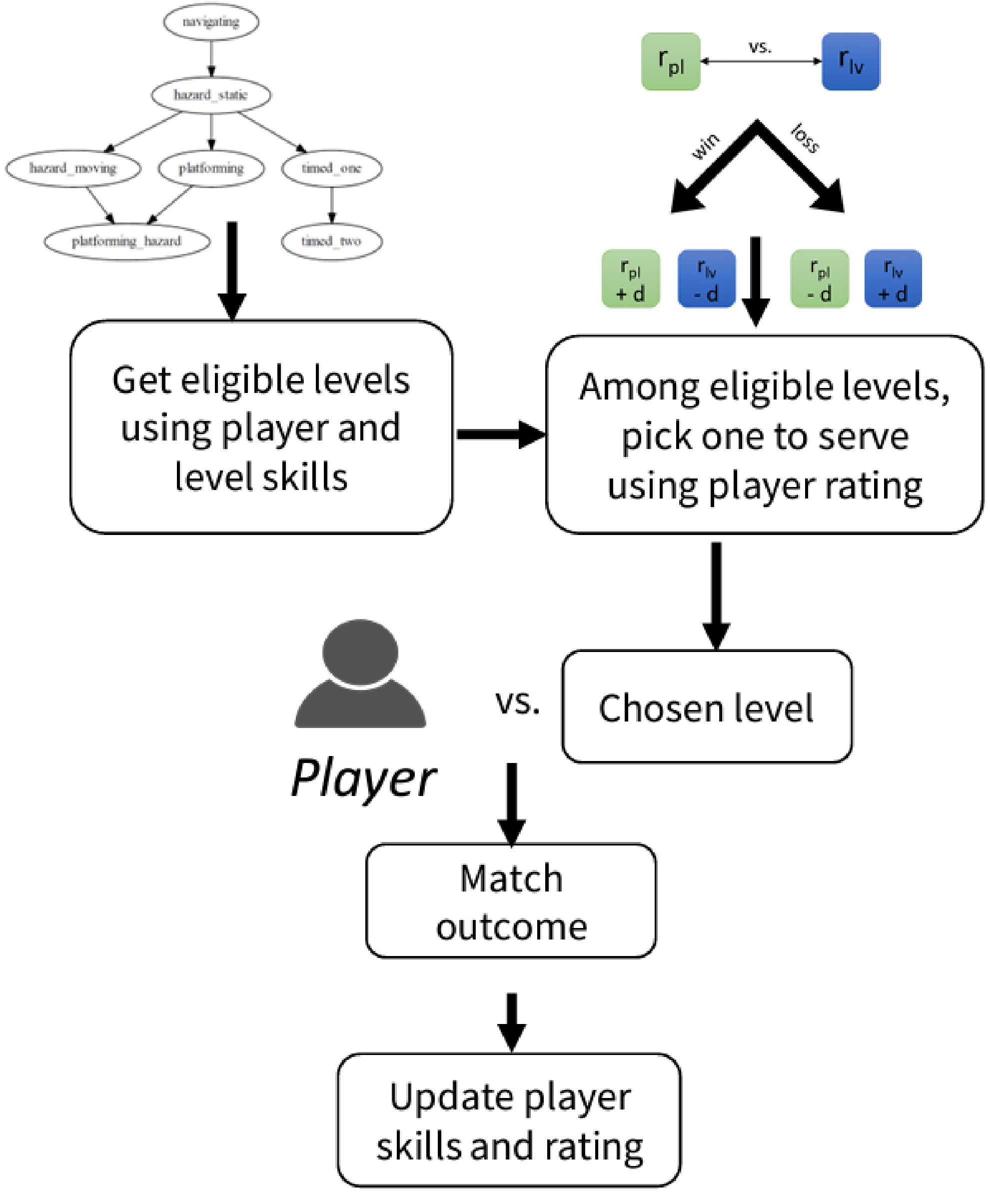
Background

- Human computation games (HCGs) model real world problems to help solve them through players
- Prior works did dynamic difficulty adjustment (DDA) in HCGs using rating systems, framing DDA as PvL matchmaking
  - Player rating → player ability
  - Level rating → level difficulty
- Skill chains
  - Define the order of player skill acquisition during gameplay
  - Can be used to define level progressions of varying difficulty



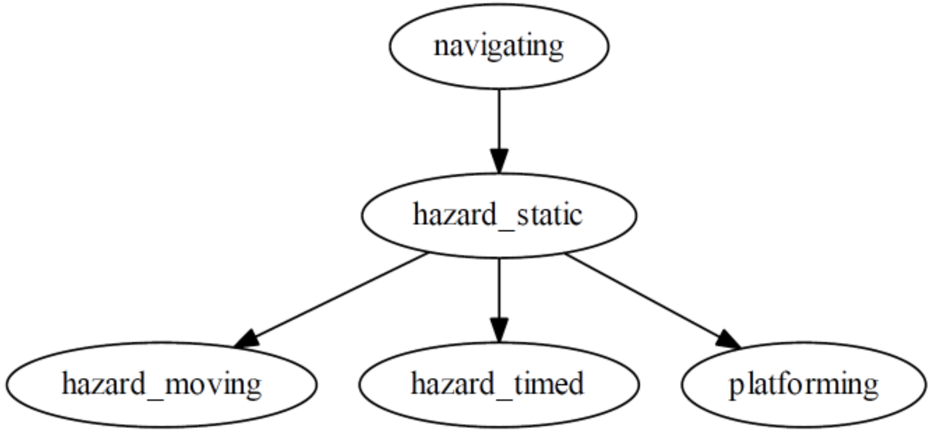
DDA Model

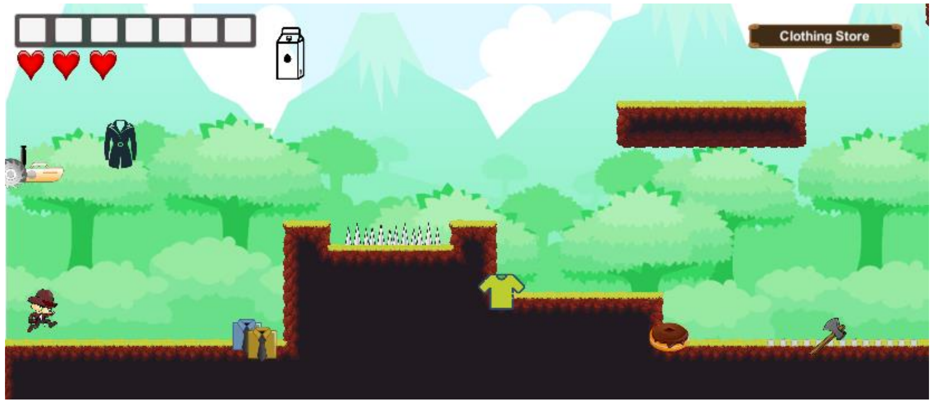
- Two step DDA process:
  - Skill Chain - determine set of eligible levels based on player’s acquired skills and skills required by levels
  - Rating System – from among the eligible levels serve the best match



Problem: Authorial Burden

- Problem:
  The use of skill chains requires significant manual authoring
  --- A skill chain must be defined for a given game
  --- Each level in the game must be annotated with the set of individual skills required to complete that level





Skills: navigating, hazard_static, hazard_timed

Approach

- Two approaches to ordering levels:
    - Compare levels’ relative proportions of similar action-context pairs in playtrace data
    - Compare levels’ similarity of level structures based on K-means clustering
    - Three-part evaluation
        - Determine best playtrace-based ordering
        - Determine best clustering-based ordering
        - Compare two new methods with existing method and random baseline

Approach

- Two approaches to ordering levels:
  - Compare levels' relative proportions of similar action-context pairs in playtrace data
  - Compare levels' similarity of level structures based on K-means clustering
  - Three-part evaluation
    - Determine best playtrace-based ordering
    - Determine best clustering-based ordering
    - Compare two new methods with existing method and random baseline

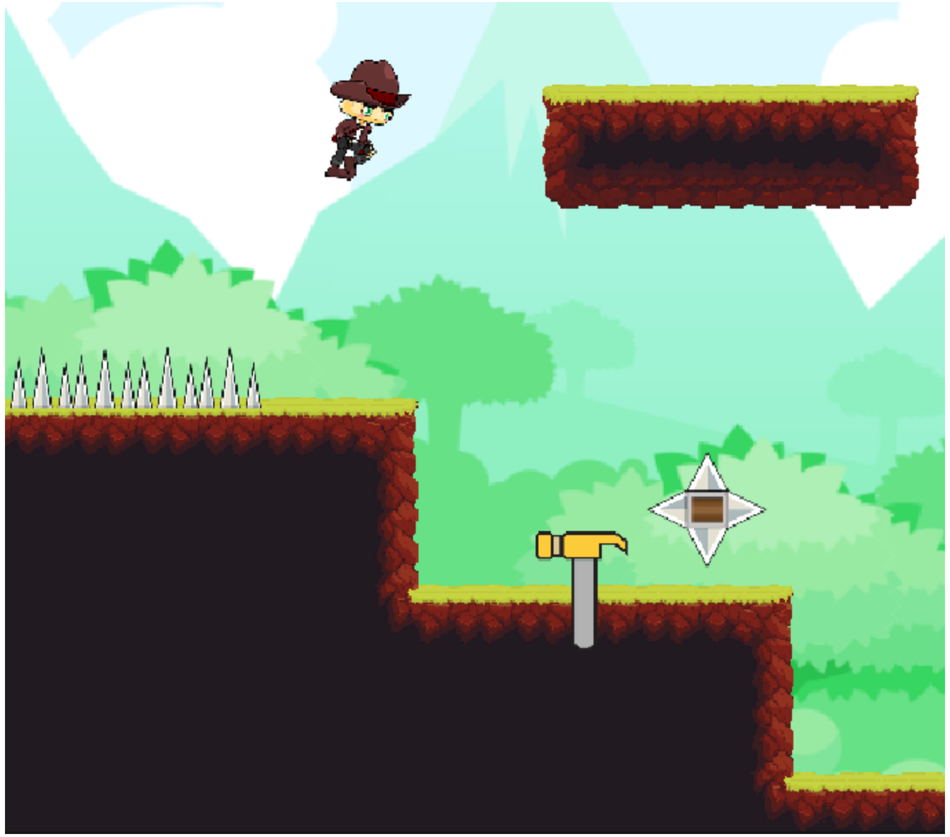
Iowa James

Action-Context Pairs in Playtraces

- Sequences of action-context pairs in playtraces of player wins vs levels

- Pairs were (action, context) 2-tuples
  - Action: Left, Right, Jump, Wrong Item
  - Context: Length-6 bitstring indicating presence/absence of game elements in 10-tile neighborhood of player

- Playtrace data gathered using Mechanical Turk
  - 60 Players
  - Levels served at random
  - Logged trajectory of time-ordered action-context pairs during playthrough
  - Filtered out losing trajectories



Action-Context Pair: (Jump, <101101>)

<Ground, Moving Platform, Item, Spikes, Timed Spikes, Star>

Action-Context Pairs in Playtraces

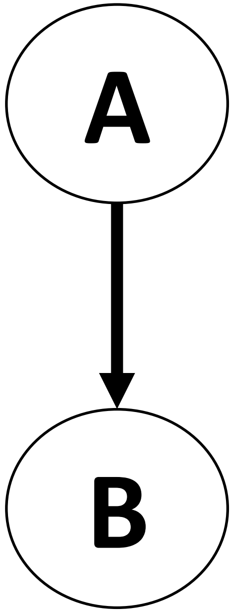
- For each level, determine the set of unique action-context pairs that appear in a threshold percentage of winning trajectories

- For level ordering, for each pair of levels A and B
  - Consider each action-context pair as a skill
  - A comes before B if

    --- % of A's skills in B > % of B's skills in A

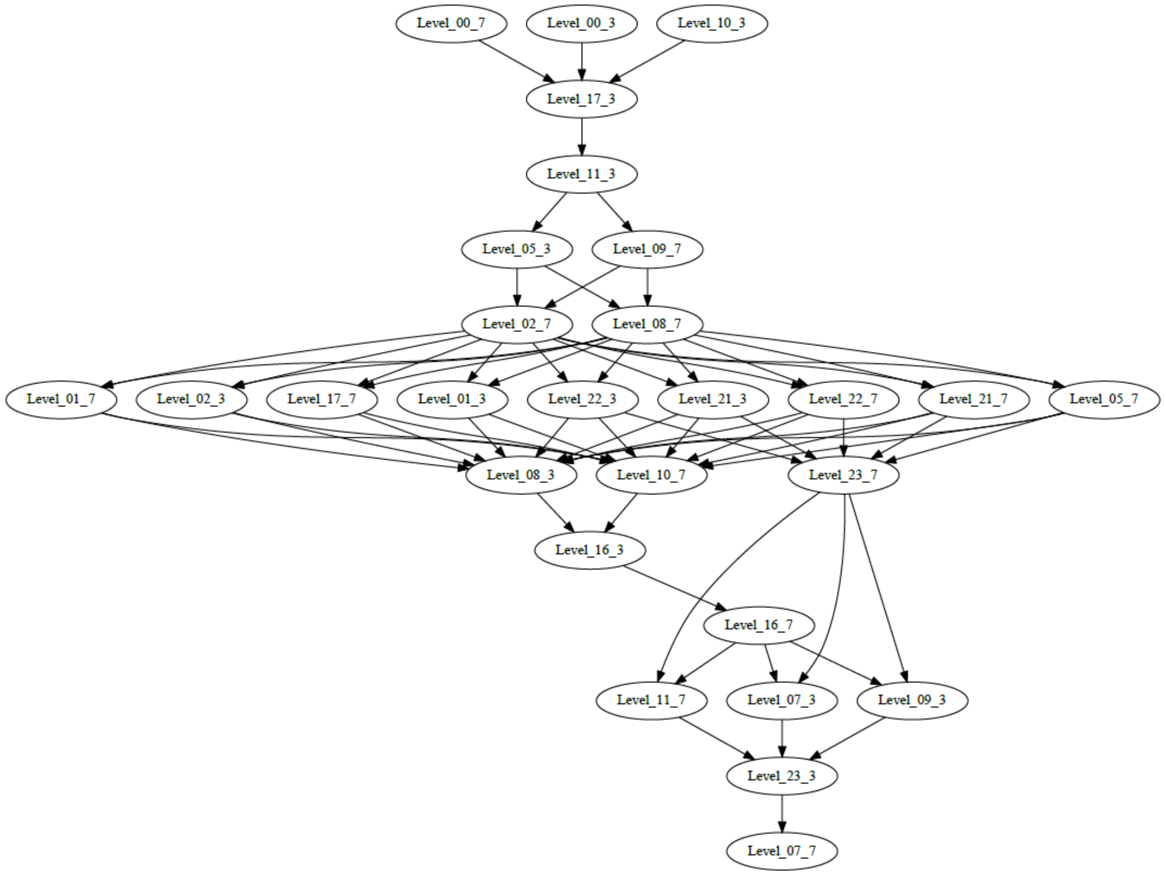
Level A: {navigating}
Level B: {navigating, platforming}

100% of A's skills required by B
50% of B's skills required by A



Action-Context Pairs in Playtraces

- Obtain a level ordering graph after processing all pairs
- To determine percentage threshold, generated orderings for thresholds=10, 20, … 100%
  - Used our knowledge of the game to judge goodness of generated orderings
  - Lower thresholds → graphs closer to expectation
  - Used 10% (PT-10) and 20% (PT-20) for experiment



Action-Context Pairs Experiment

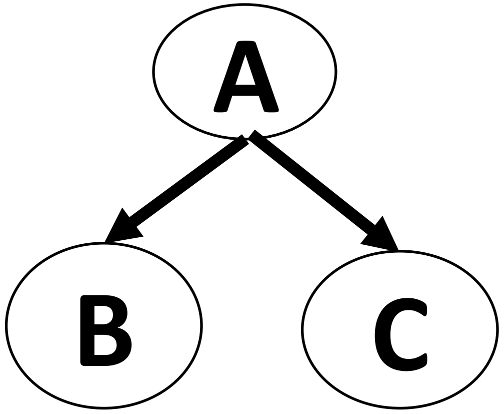
- 111 players recruited through Mechanical Turk
- Players randomly assigned to one of the 2 orderings:
  - PT-10 (10% thresholding)
  - PT-20 (20% thresholding)
- Variables
  - *Levels Completed*
  - *Total Matches*

Variable	PT−10 (n=59)	PT−20 (n=52)
Levels Completed ($p = .039$)	2	1
Total Matches ($p = .24$)	6	5.5

Clustering

- Applied K-means clustering on 16x16 segments extracted from all 50 levels
- Clusters represent groups of segments that have similar level structures
- For each level, assign length-k bitstring indicating clusters that contain at least 1 segment from that level
- For ordering, for each pair of levels A and B
  - A comes before B if A's cluster memberships form a subset of B's cluster memberships
- E.g. k=3, A = {100}, B = {101} and C = {110}

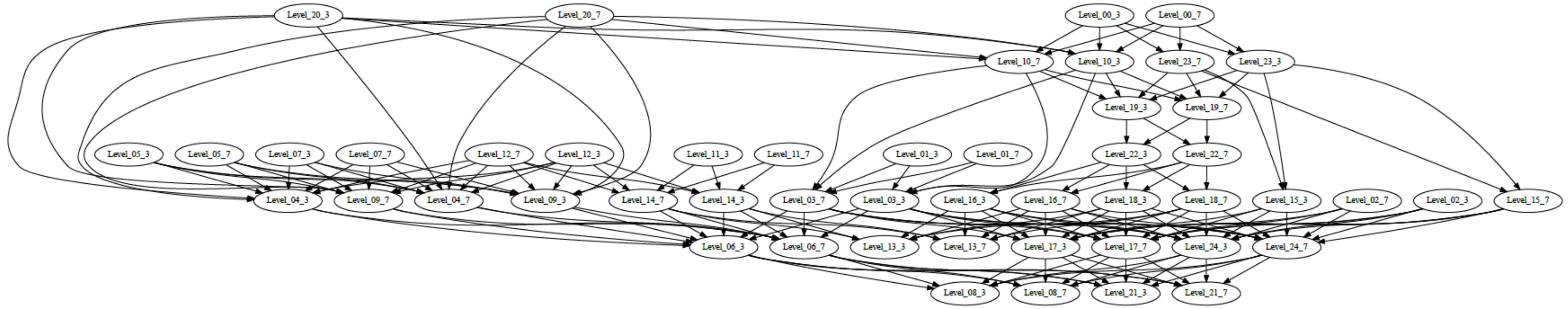
Level A: {100}
Level B: {101}
Level C: {110}



Clustering

- After processing all level pairs, obtain a level ordering graph

- To determine value of k to use, generated orderings for k=1 to 20
  - Use knowledge of the game to judge goodness of orderings
  - Prefer deeper over shallower graphs
  - Lower values of k → flatter, broader graphs due to fewer clusters leading to more levels having similar cluster memberships
  - Tested k=6 (KM-6) and k=20 (KM-20) in the following experiment



Clustering Experiment

- 113 players recruited through Mechanical Turk

- Players randomly assigned to one of the 2 orderings:
  - KM-6 (6 clusters)
  - KM-20 (20 clusters)

- Variables
  - *Levels Completed*
  - *Total Matches*

Variable	KM−6 (n=55)	KM−20 (n=58)
Levels Completed ($p = .52$)	1	2
Total Matches ($p = .9$)	6	6

Evaluation

- Recruited 335 players using Mechanical Turk

- Players randomly assigned to one of the 4 orderings:
  - RAND – randomly serve a level yet to be completed
  - SKILL – use prior DDA system
  - KM-20 - 20 cluster-based ordering
  - PT-10 – 10% thresholding playtrace-based ordering

- Variables
  - *Levels Completed*
  - *Total Matches*
  - *Correct Items*
  - *Incorrect Items*
  - *Highest Level Rating*

Evaluation

Variable	RAND (n=78)	SKILL (n=96)	KM-20 (n=85)	PT-10 (n=76)
Levels Completed ($p < .01$)	1^a	2^b	2^b	2^b
Total Matches ($p = .77$)	8	6	6	6
Correct Items ($p = .052$)	7.5^a	9.5^{ab}	8^{ab}	14^b
Incorrect Items ($p = .33$)	7	6	6	7.5
Highest Level Rating ($p < .01$)	1496^a	1669^b	1669^b	1854^c

- Takeaways

  --- Significant differences for Levels Completed, Correct Items, Highest Level Rating

  --- New KM-20 and PT-10 orderings allowed players to complete a similar amount of levels as prior SKILL method while reducing authorial load

  --- PT-10 (playtrace) allowed players to complete significantly harder levels

  --- KM-20 (clustering) does not outperform SKILL or PT-10 but requires least manual input while not doing any worse

Future Work

- Apply on other types of HCGs
- Learn progressions for educational games
- Context relationships subsets

Future Work

- Apply on other types of HCGs
- Learn progressions for educational games
- Context relationships subsets

Contact
Anurag Sarkar
Northeastern University
sarkar.an@northeastern.edu