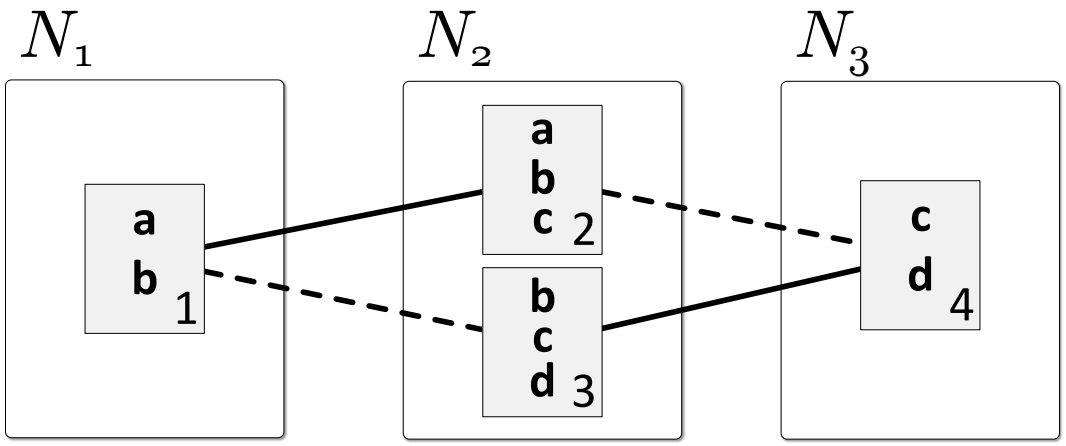
Crowd-sourcing a solution to an NP-hard problem

Amit Kadan

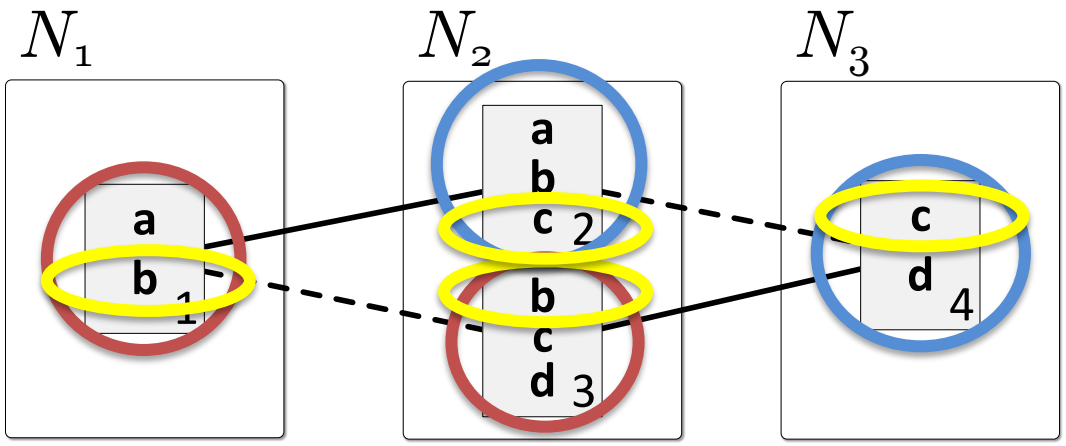
September 29th, 2016

The problem: N-way matching



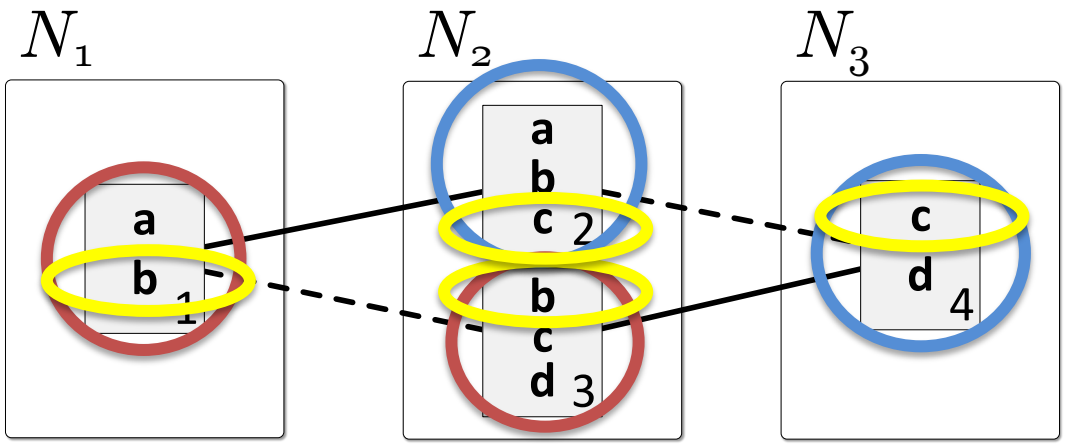
- *N inputs*
  - Inputs have elements (*1, 2, 3, 4*)
  - Elements have properties (*a, b, c, d*)

The problem: N-way matching



- Goal: Match elements from distinct inputs.
  - ➢ More shared properties => better match.

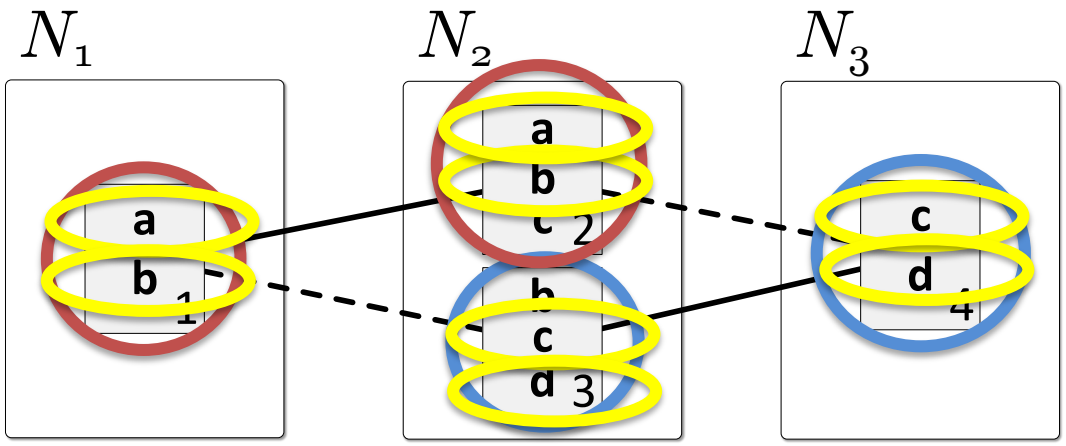
The problem: N-way matching



Suboptimal

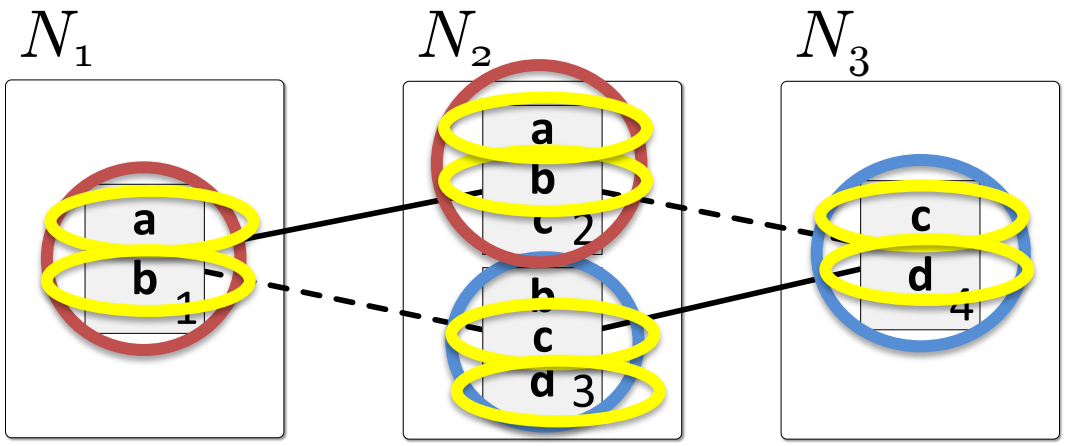
- Goal: Match elements from distinct inputs.
  - More shared properties => better match.

The problem: N-way matching



- Goal: Match elements from distinct inputs.
  - More shared properties => better match.

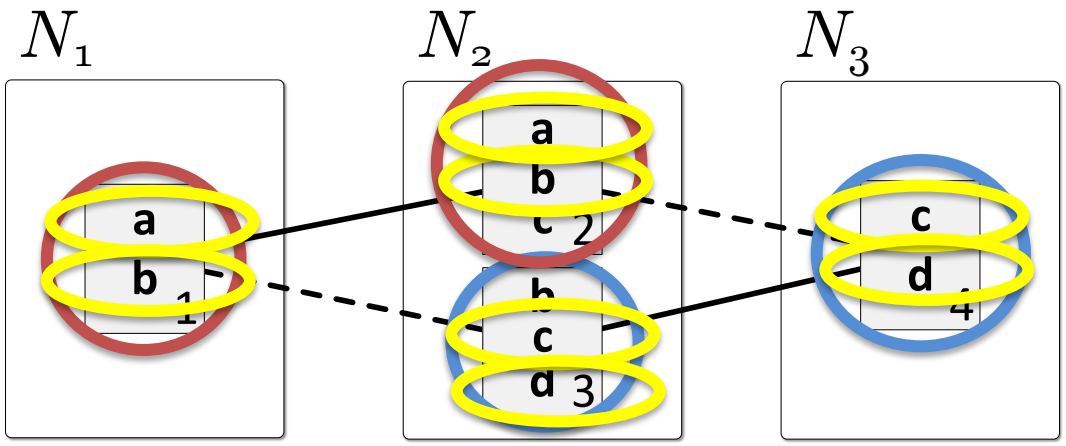
The problem: N-way matching



Optimal!

- Goal: Match elements from distinct inputs.
  - ➢ More shared properties => better match.

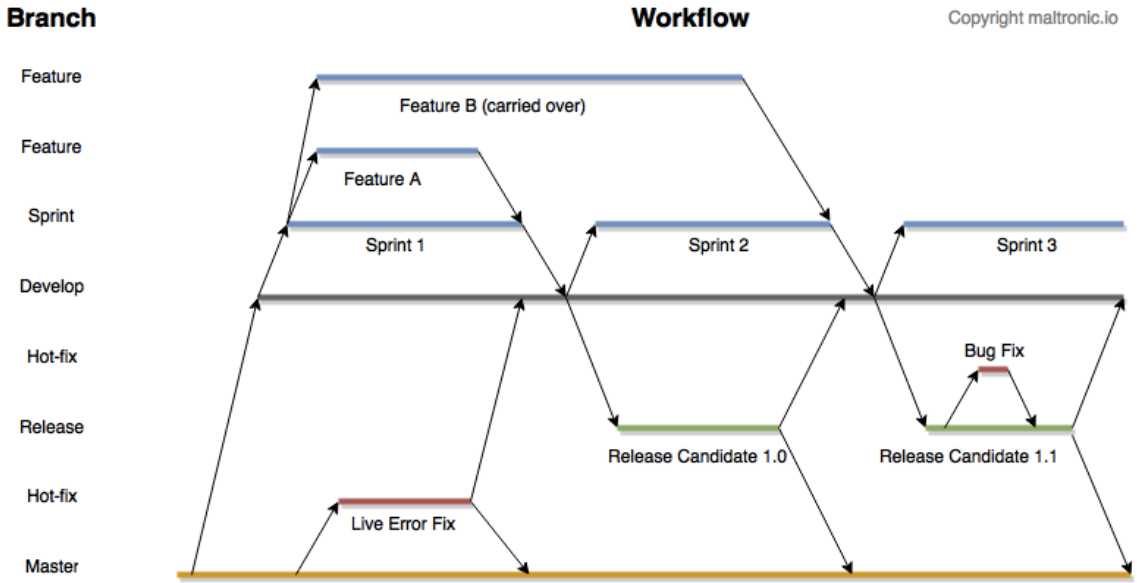
The problem: N-way matching



Optimal!

- Goal: Match elements from distinct inputs.
  - More shared properties => better match.
  - For small cases, optimal easy to find.

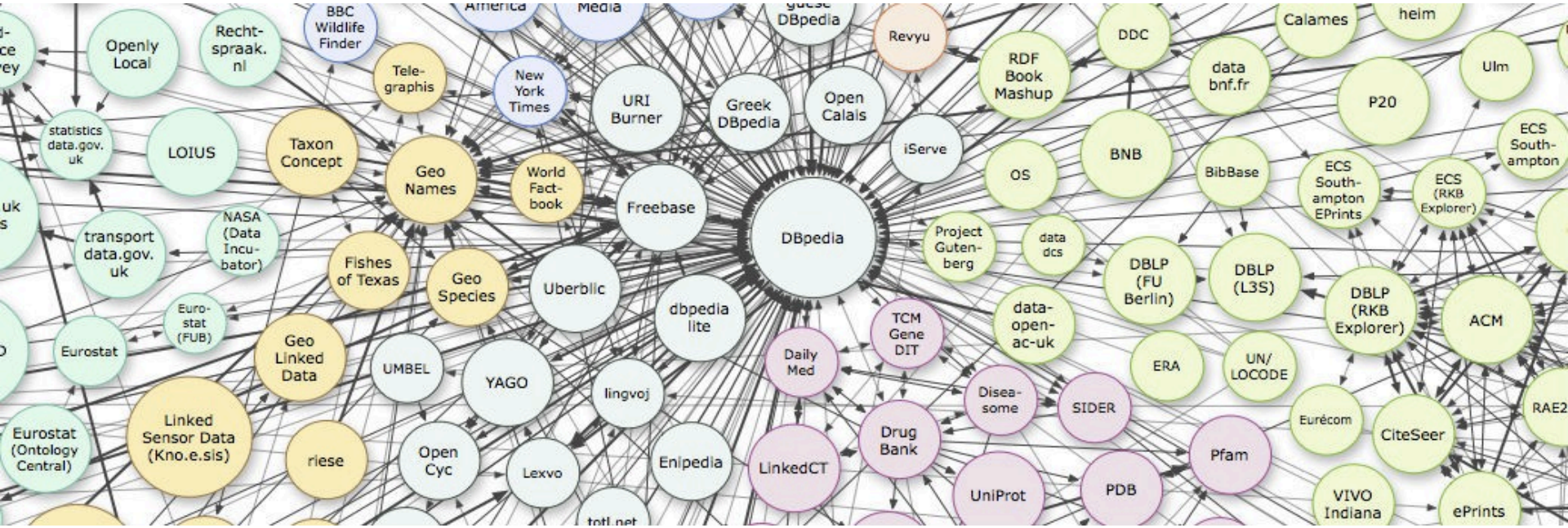
Applications of N-way matching



Application of N-way matching: composing branches in software management system.

- Compose branches of software configuration management system.
- Combine many product lines into single copy.
- Reconcile representations of many stakeholders.

Applications of N-way matching



- The applications of n-way match rely on big cases.
- i.e. 10 inputs, 200 elements, 250 distinct properties.
- For $N > 2$ (# inputs), finding optimal is NP-hard…
  - ➢ Need heuristic solutions.

Heuristic solutions to N-way matching

Algorithm	Approximation Factor?	Feasible?
Greedy		
Greedy + Local Search		
Pairwise		
Greedy (by chunk)		
NwM		

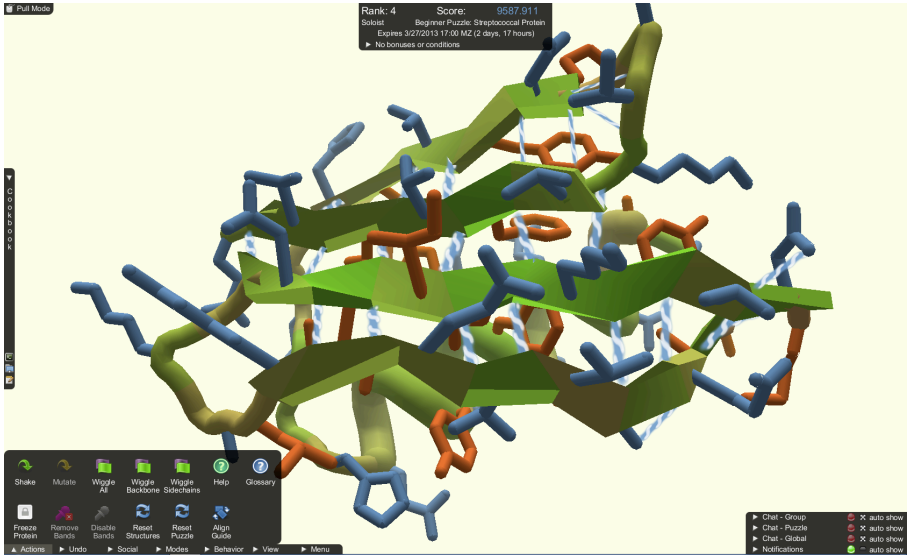
Best heuristic solution so far: NwM

- Pros
  - Feasible (low time complexity).
  - Best performance out of feasible solutions.
- Cons
  - No approximation factor.
  - Lots of room for improvement.
  - Sometimes can be slow (5 min for big cases).

Approach: Improve solution of NwM using a serious game

Thanks to Christina Chung, Asako Matsuoka, Elsie Yang, Fei Huang, Si Hua Cao Lui, Lionheart Xiong, Angel You, and Nicole Sultanam for designing and implementing the game

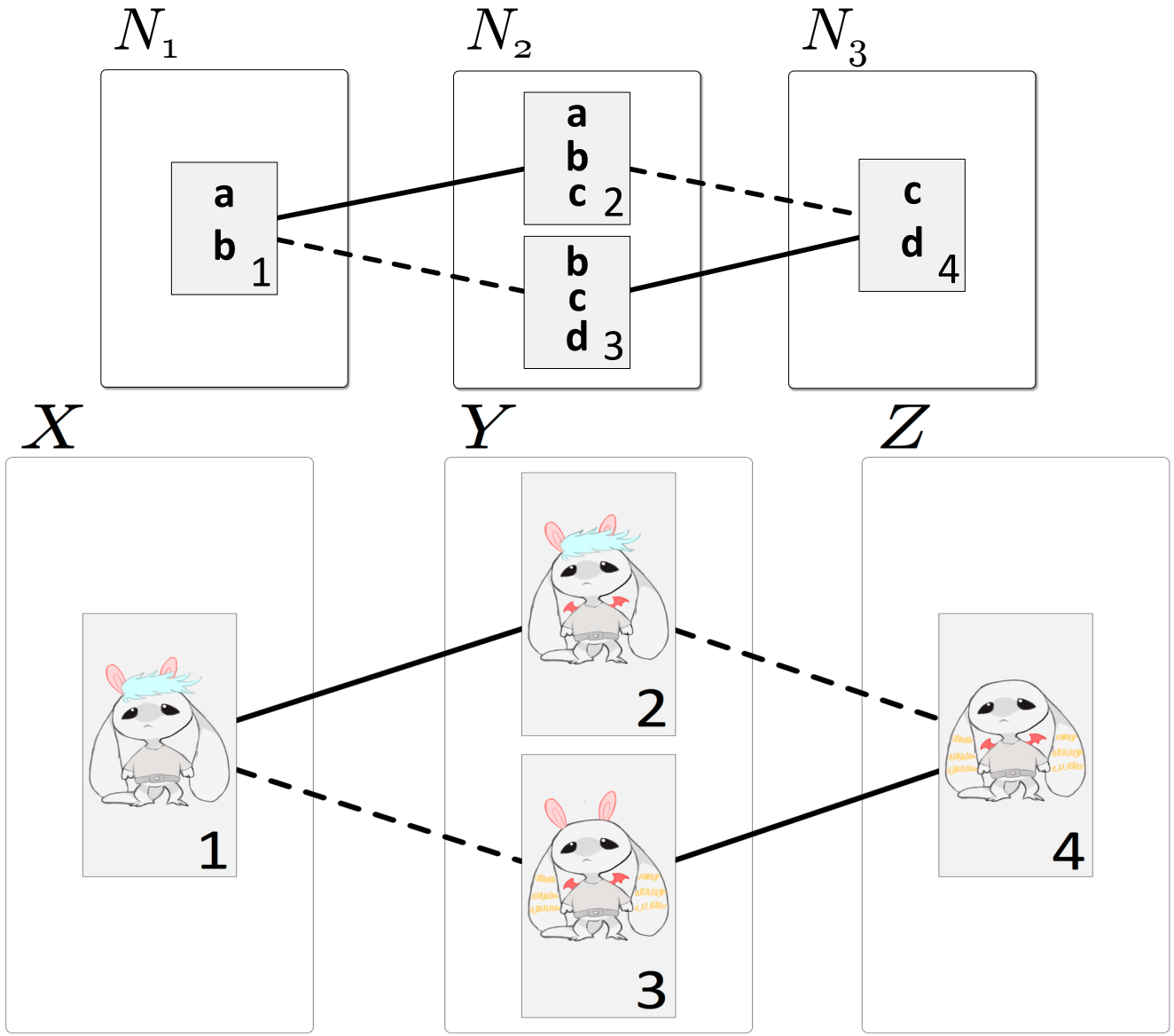
Serious games



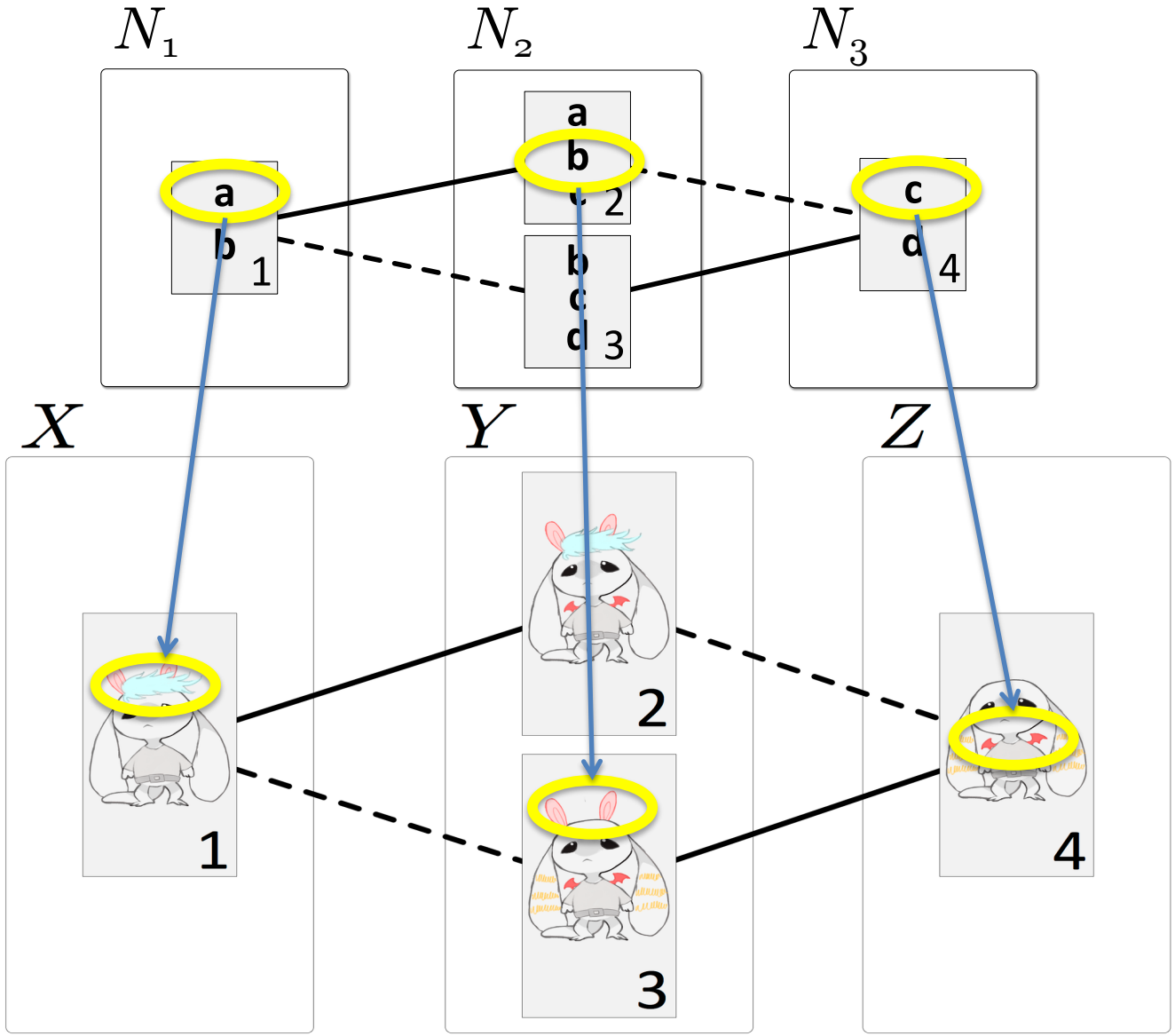
Example of serious game: Foldit.

- Serious games
  - Not for entertainment.
  - Take advantage of human cognition.
  - Typically used for cognitive tasks, recently also to solve computational problems (i.e. Foldit).

Encoding of game



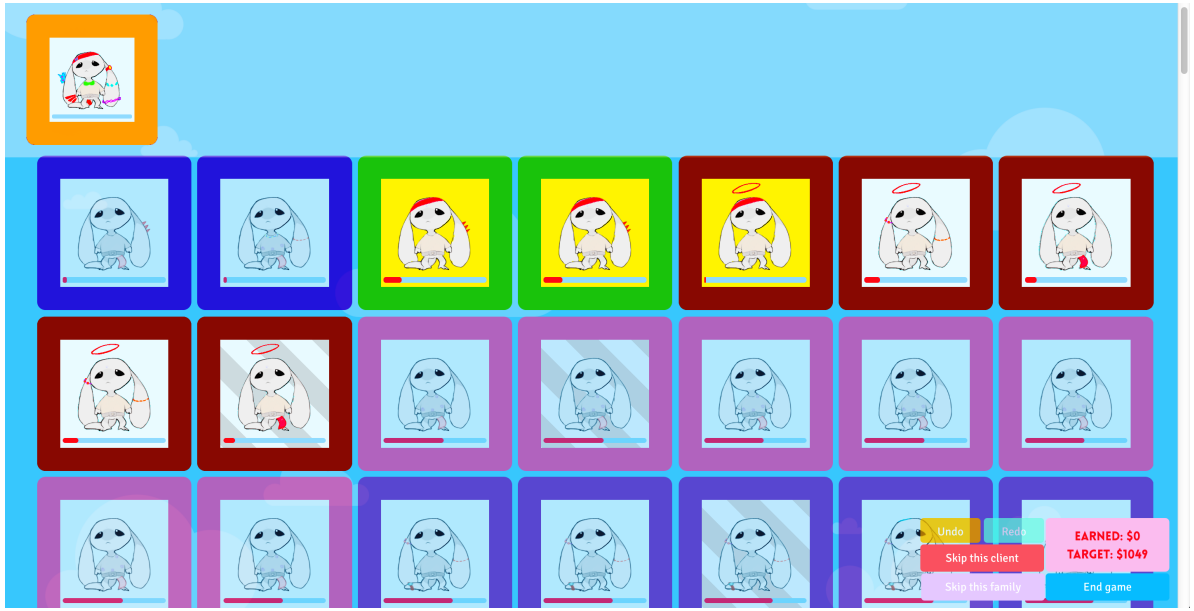
Encoding of game



How the game works

- Players consider all elements at once (as in NwM).
- Players improve solution of NwM.
- Improvements are incremental.
  - Each player starts from best solution found by any player thus far.

Gameplay

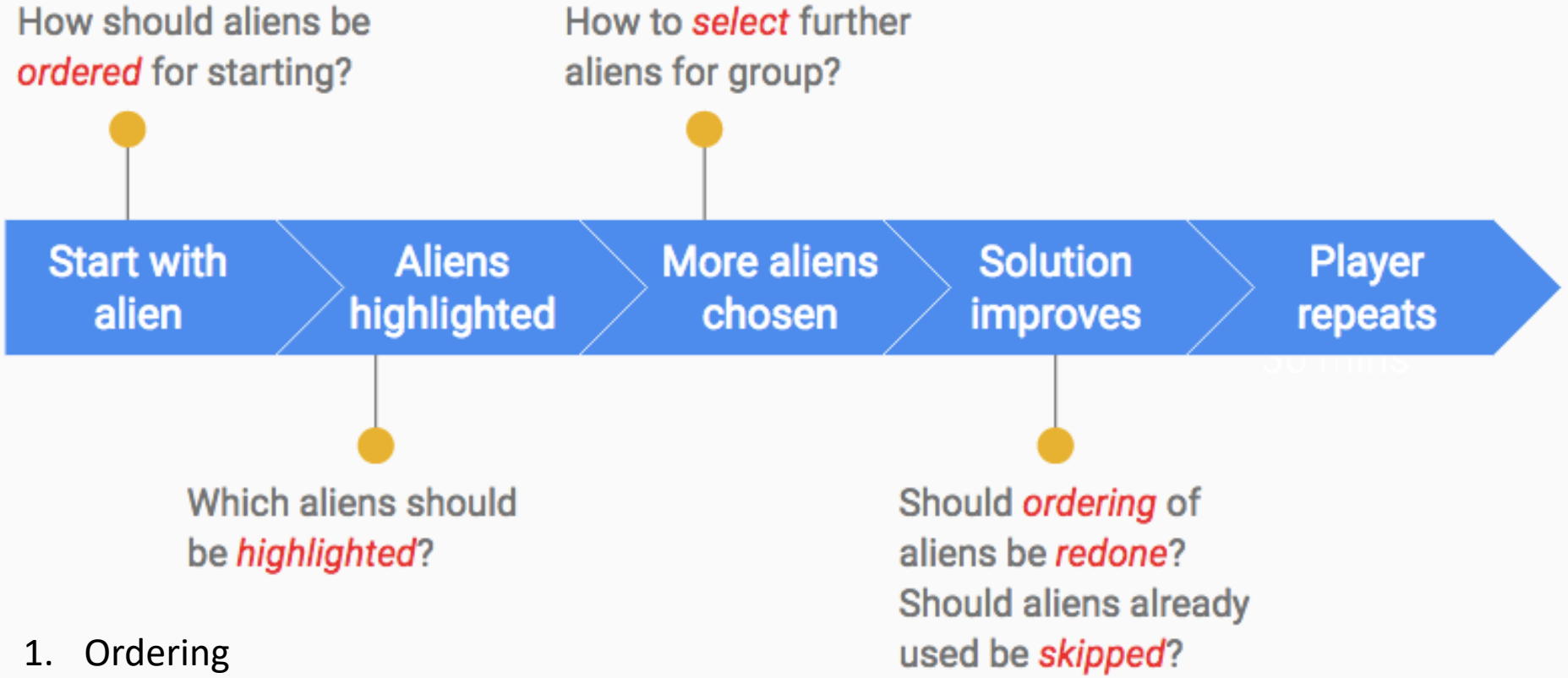


1. Player starts with some alien.
2. Similar aliens highlighted.
3. Choose alien to add to group.
4. Repeat 2-3 until score goes up or player gives up.
5. Go to step 1.

Goal of summer research: **Identify game configuration that will lead to player success**

Approach: **Build simulator that mimics human player.**

Design of simulator: Identifying configuration parameters



1. Ordering
2. Highlighting
3. Selecting *
4. Reordering

* Not a game parameter

Why a simulator?

Can try many different configurations of the game.

- Cheaply (i.e. don't need a bunch of people)
- Quickly – machine can perform matching much quicker than people.

Design of simulator: The algorithm

1. Order aliens with ordering strategy.
2. Start a new group with first alien in ordering.
3. Highlight related aliens with highlighting strategy
4. Add alien to group with selection strategy.
5. While selection criteria still satisfied, keep adding aliens to group.
6. If score of solution goes up during group formation, save solution.
7. If solution improves, reorder aliens.
8. Repeat steps 1-7 until tried very single alien with no improvement

Research Questions

RQ1: Which configuration of simulator will result in best solutions being formed?

RQ2: Can simulator significantly improve solutions found by NwM?

Experiment 1: Identify best configuration parameters

- Answer RQ1: Which configuration of simulator will result in best solutions being formed?
  - ➢ Test many different settings of parameters over many different cases.

Experiment 1: Picking cases

- Cases taken from previous papers studying n-way matching [1], [2], [3].
- 5 cases taken from [1]
- 7 cases from [2] generated using FeatureHouse tool.
- 3 cases taken from [3].

[1]. J. Rubin and M. Chechik, “N-Way Model Merging,” in Proc. ESEC/FSE, 2013, pp. 301–311.

[2]. S. Apel, C. Kastner, and C. Lengauer, “Language-Independent and Automated Software Composition: The FeatureHouse Experience,” IEEE Transactions of Software Engineering, vol. 39, no. 1, pp. 63–79, 2013.

[3]. L. Linsbauer, R. E. Lopez-Herrejon, and A. Egyed, “Recov- ering Traceability between Features and Code in Product Variants,” in Proc. of the International Software Product Line Conference (SPLC’13), 2013, pp. 131–140.

Experiment 1: The different configurations

Configuration Parameter	Different Settings
Ordering	• Random • Size +/- • Score +/- • Bar +/-
Highlighting	• None • 1 p in common with one • 1 p in common with all • Group size in common
Selecting	• Best local score increase • Best global score increase
Reordering	• Reorder + skip • Reorder • none

Experiment 1: Methodology

- Test all variants of simulator over all cases and see which variant has best average performance.
  - 7 x 4 x 2 x 3 = 168 variants of simulator.
  - 168 x 15 = 1920 runs of simulator.

Experiment 1: Results

Configuration Parameter	Different Settings
Ordering	• Random • Size +/- • Score +/- • Bar +/-
Highlighting	• None • 1 p in common with one • 1 p in common with all • Group size in common
Selecting	• Best local score increase • Best global score increase
Reordering	• Reorder + skip • Reorder • none

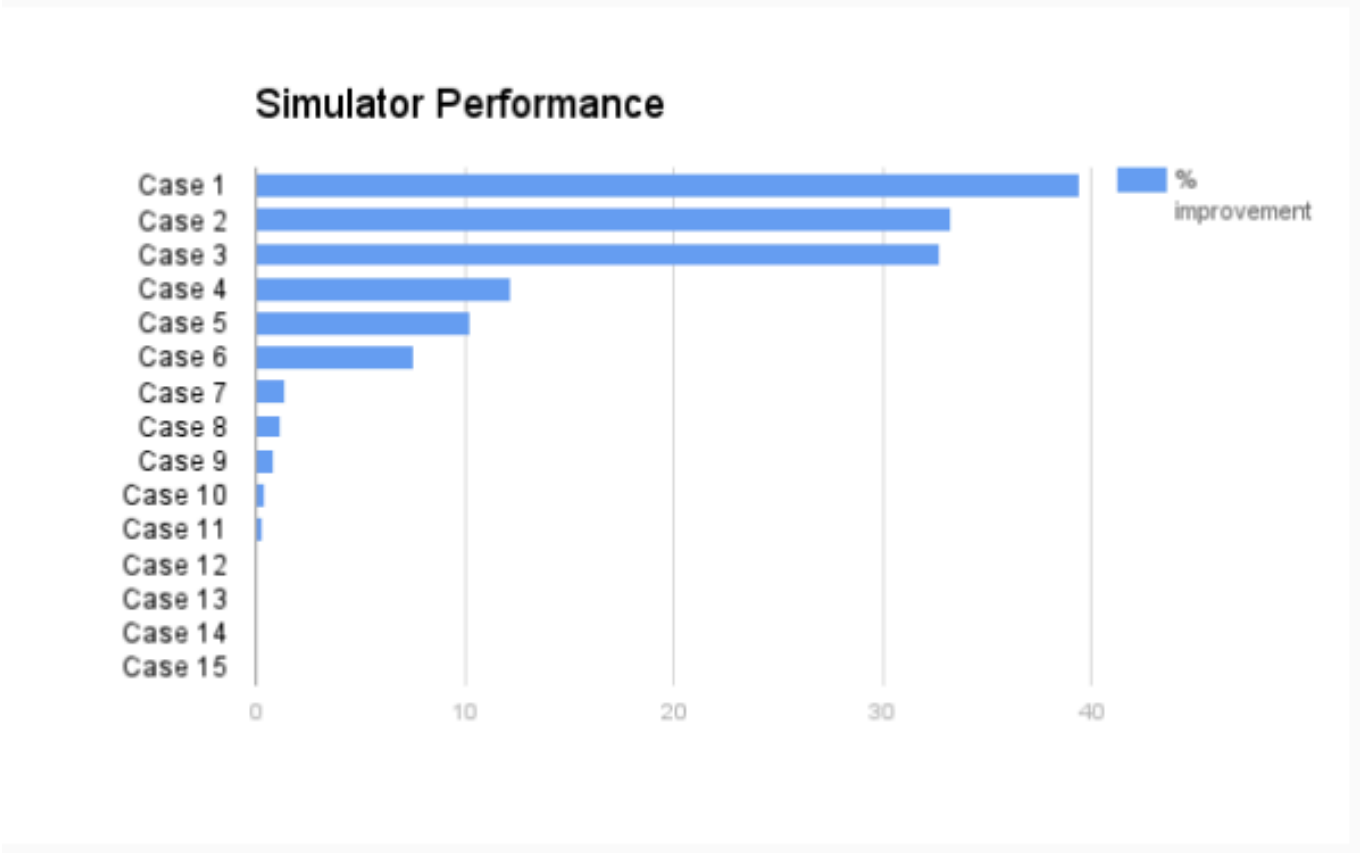
Experiment 2: Significant Improvement over NwM

- Answer RQ2: Can simulator make a significant improvement over solution found by NwM?
  - ➢ Measure improvement of best configuration of simulator over NwM.

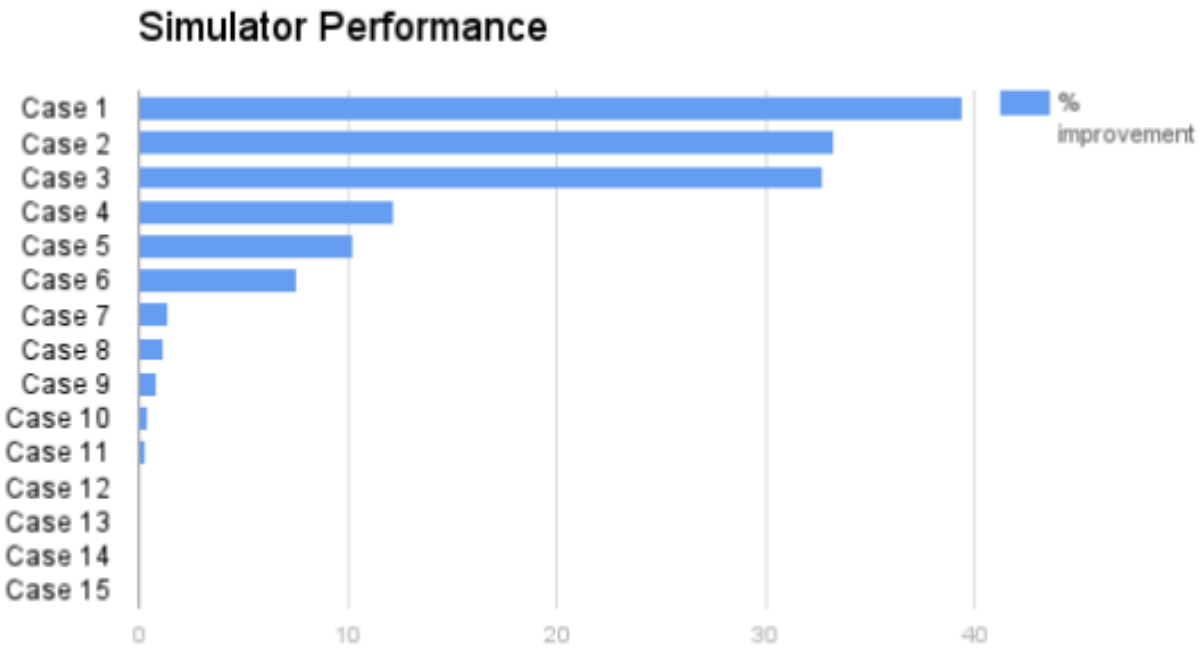
Experiment 2: Methodology

- Same cases used as in Experiment 1.
- Measure improvement of simulator for each case.
- Perform paired t-test to see whether overall improvement is significant.

Experiment 2: Results



Experiment 2: Results

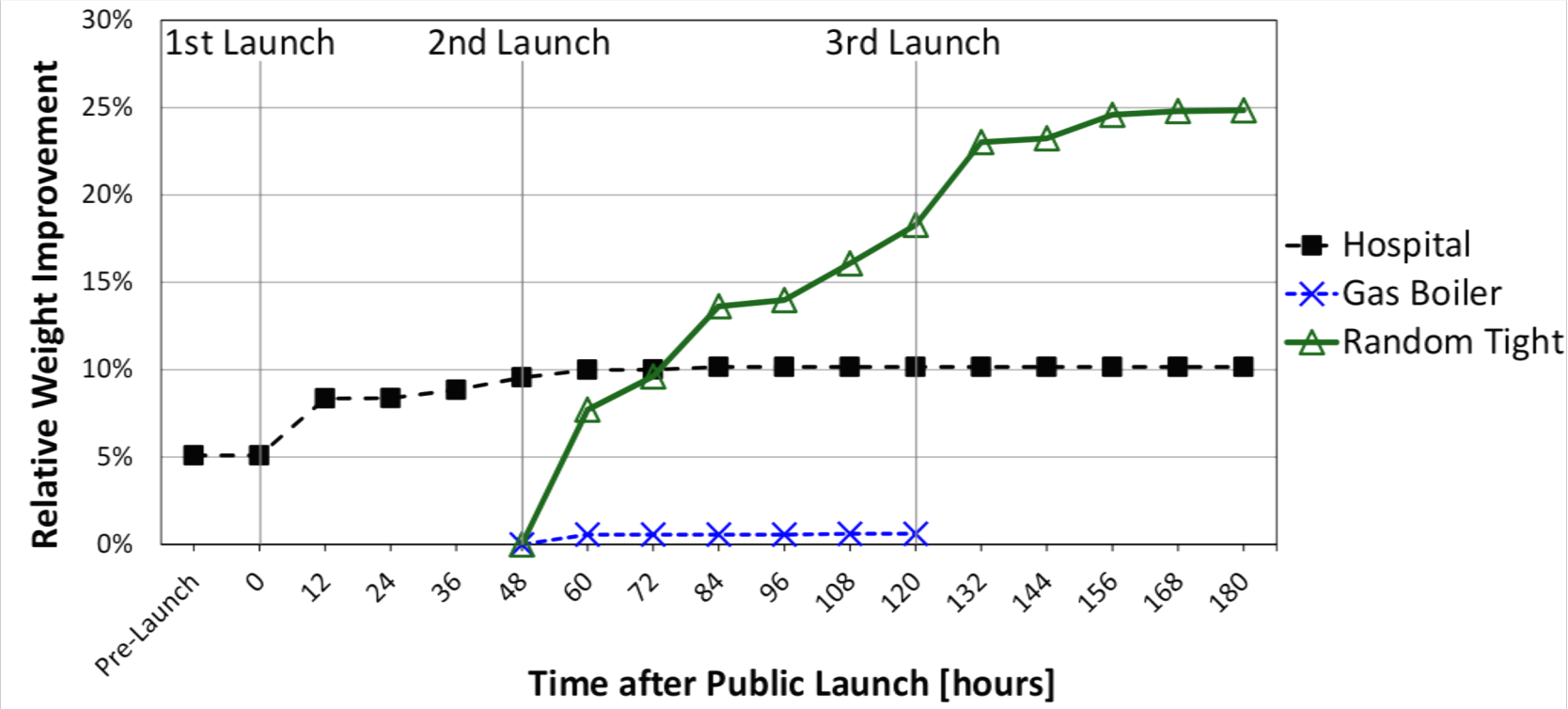


- Paired t-test returned p-value of 0.002.
  - ➢ Significant improvement achieved.

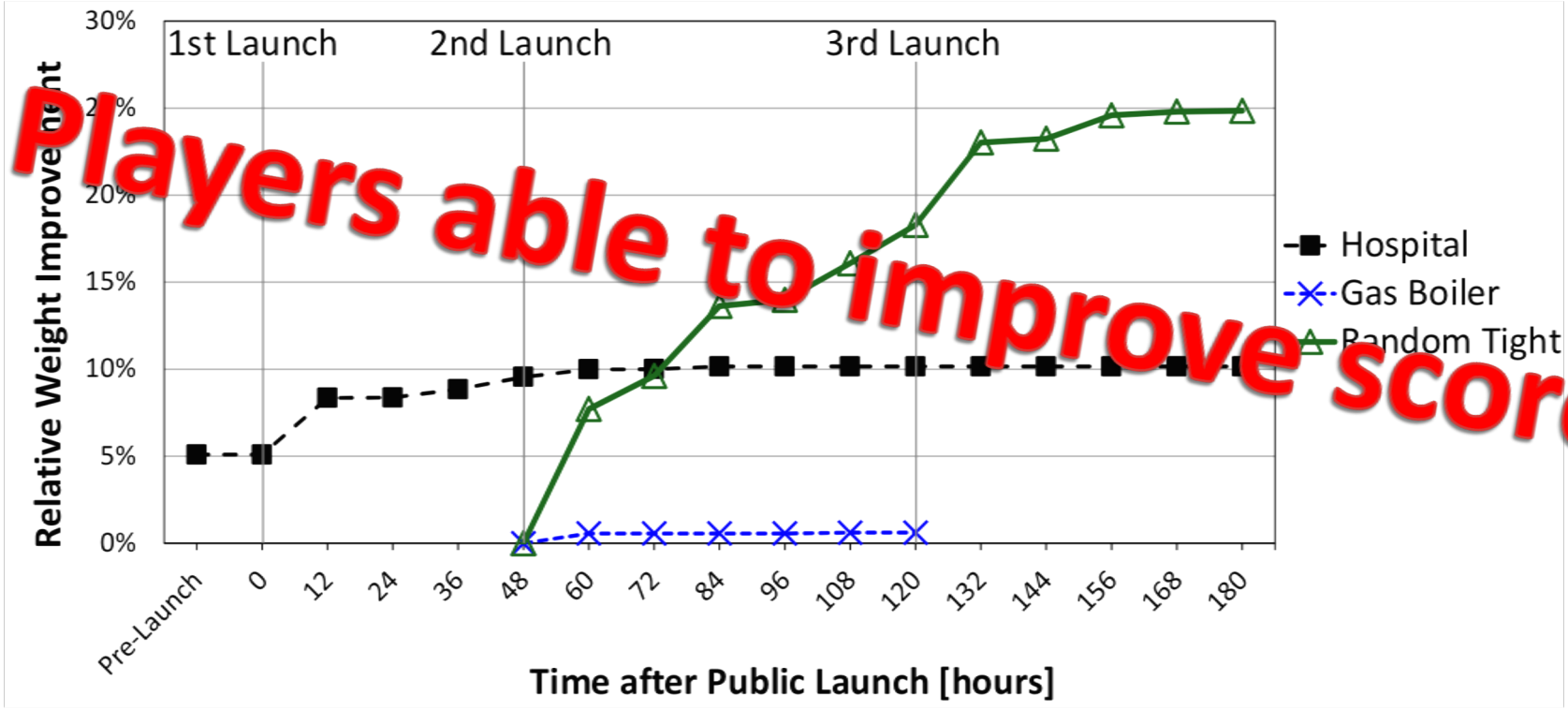
How does this relate to the game?

- Have identified good game configuration.
- Significant improvement suggests people can significantly improve the score of NwM.

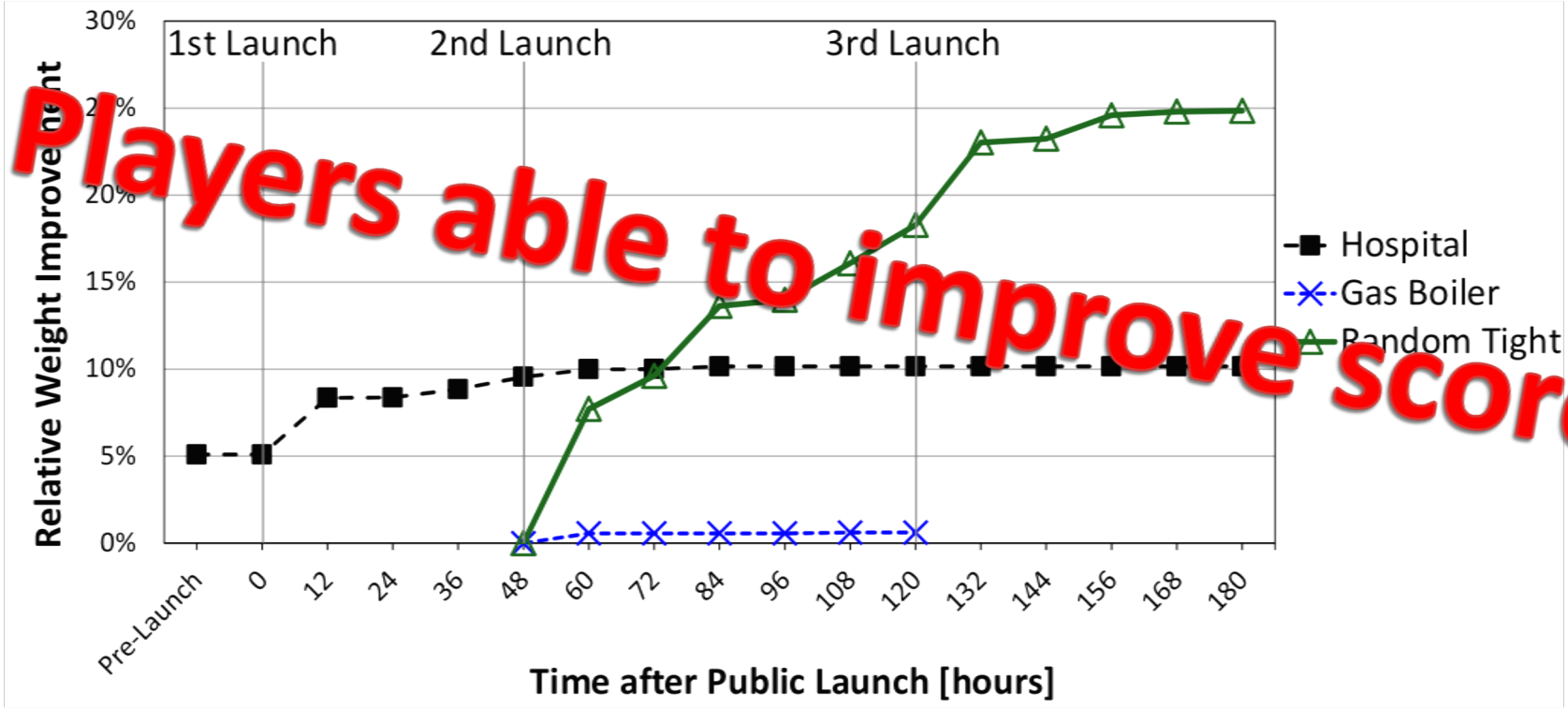
Player performance



Player performance

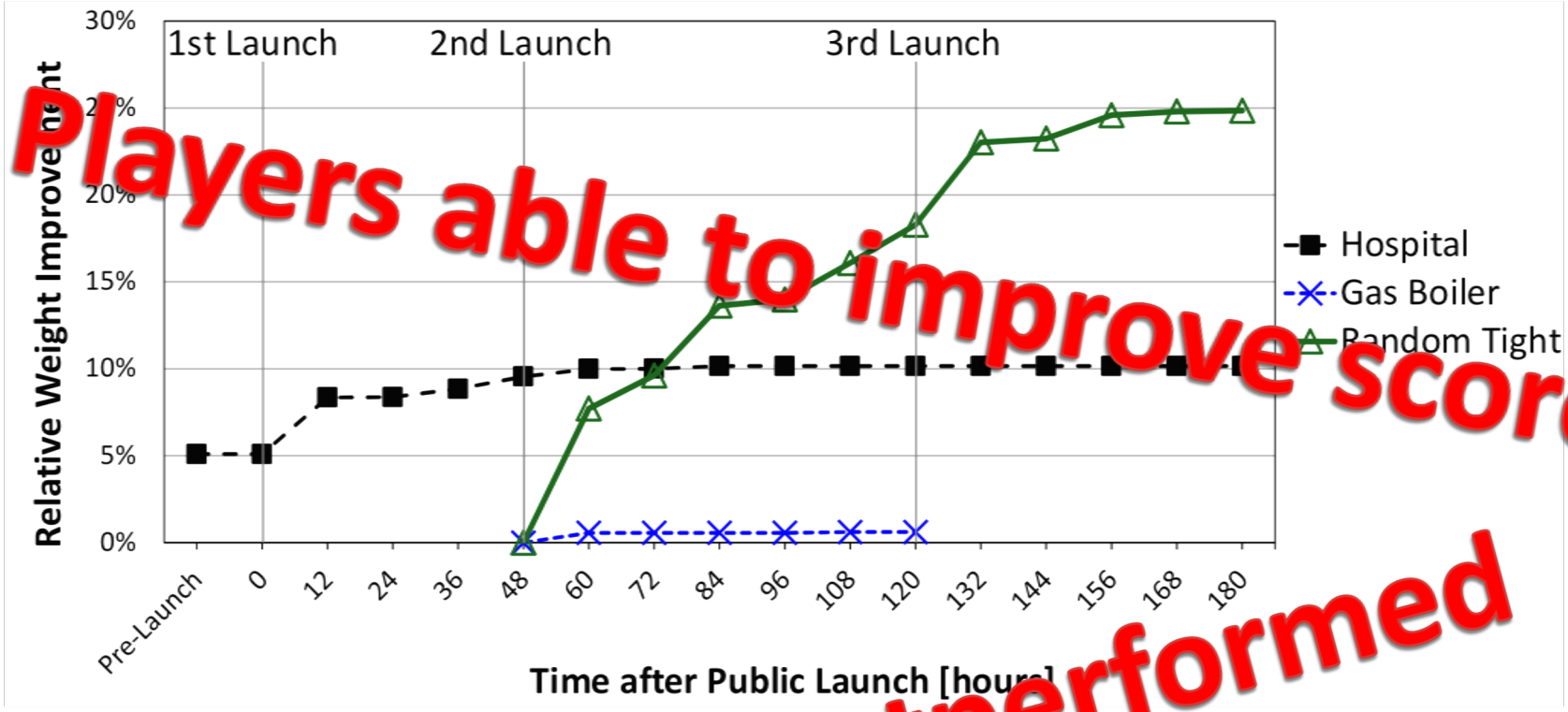


Player performance



Case Study	NwM	HSim	MatchMakers
hospital	45952	50684	50619
gasBoilerSystem	136286	136772	137117
randomTight	9582	12724	11964

Player performance



Players able to improve score!

Players outperformed simulator in one case

Case Study	NwM	[illegible]IM	MATCHMAKERS
hospital	45[illegible]2	50684	50619
gasBoilerSystem	136286	136772	137117
RandomTight	9582	12724	11964

Conclusion: Review

- **Main Goal:** Improve scores achieved by best heuristic algorithm – NwM – using serious game.
  - Players managed to play game and significantly improve score of NwM.
- **Sub-goals:** Identify good game configuration, test whether significant improvement is feasible using simulator.
  - Configuration identified by simulator allowed people to improve NwM scores.

Present & Future Work

- Simulator usually outperforms players (i.e. players give up before beating simulator score)
  - Make game more captivating.
  - Identify better game configuration to make game easier.
  - Allow more flexibility in game (i.e. writing scripts).
- Simulator based on naïve intuitions of how game is played.
  - With insights from simulator and NwM, should be able to construct better algorithm for doing n-way match.