Effects of Player-Level Matchmaking Methods in a Live Citizen Science Game

Alexander D. Stoneman[1], Josh Aaron Miller[2], Seth Cooper[2]

[1]Vanderbilt University,
[2]Northeastern University
alexander.d.stoneman@vanderbilt.edu, {miller.josh, se.cooper}@northeastern.edu

Abstract

Citizen science games must balance task difficulty with player skill to ensure optimal engagement and performance. This issue has been previously addressed via player-level matchmaking, a dynamic difficulty adjustment method in which player and level ratings are used to present levels best suited for players' individual abilities. However, this work has been done in small, isolated test games and left out potential techniques that could further improve player performance. Therefore, we examined the effects of player-level matchmaking in Foldit, a live citizen science game. An experiment with 221 players demonstrated that dynamic matchmaking approaches led to significantly more levels completed, as well as a more challenging highest level completed, compared to random level ordering, but not greater than a static approach. We conclude that player-level matchmaking is worth consideration in the context of live citizen science games, potentially paired with other dynamic difficulty adjustment methods.

Introduction

In 2020, over 214 million people in the United States reported playing video games, including at least one person in 3 of every 4 households (Entertainment Software Association 2021). Citizen science games (CSGs) model computationally-intensive tasks as games to harness the abilities of this large population of players to solve real-world scientific problems (Bonney et al. 2014; Burnett et al. 2016). While the subject matter of the games may be complex, the games themselves are intended to be accessible and enjoyable to the general public. Some of the areas in which CSGs have found success are image labeling (Von Ahn and Dabbish 2004), graph theory (Cusack et al. 2010), genomics (Rallapalli et al. 2015), and protein design (Koepnick et al. 2019).

In order for CSGs to be maximally effective, players must be given tasks that are appropriate for their skill (Jennett et al. 2016; Sweetser and Wyeth 2005). Tasks that are too easy will bore players, while overly-complex tasks will cause frustration (Larche and Dixon 2020). Games commonly attempt to do this by serving levels in order of increasing difficulty, a static approach. However, these methods are not easily translatable to CSGs, because the games' tasks are rooted in real-life phenomena and inherently lack optimal solutions, meaning they have an unknown and unchangable level of difficulty (Cooper, Deterding, and Tsapakos 2016). Past attempts to order CSG levels have been based in heuristics that cannot be easily generalized or validated (Logas et al. 2014). Moreover, adapting the levels themselves can detract from the scientific value of the results.

To solve this problem, some games research has explored dynamic difficulty adjustment (DDA) via player-level matchmaking.

Cooper et al. adapted rating systems such as Elo (Elo 1978) and Glicko-2 (Glickman 2001) to give ratings to both players and levels, and treated a single level assignment as a game between the two entities (Cooper, Deterding, and Tsapakos 2016). Further work in this domain examined the engagement effects of different matchmaking difficulty curves, including logistic, exponential, and constant functions and compositions (Sarkar and Cooper 2019). Games have also used skill chains, or orderings of the skills players acquire through gameplay, to further tailor level difficulty (Cook 2007; Sarkar and Cooper 2020). These techniques were each shown to effectively scale task difficulty with player skill level.

However, all of this prior research has been completed in CSGs that are isolated from a live game environment. In other words, players were usually recruited externally and accessed the game solely to participate in the study, rather than downloading the game and playing on their own. This represents a departure from live CSGs in which most scientific discoveries were made.

Expanding player-level matchmaking to a live CSG could improve DDA in a domain in which it is otherwise challenging. This technique could lead to smoother difficulty curves and onboarding, which would improve player engagement. This could in turn result in more gameplay, and thus scientific contributions, from a greater number of players.

To this end, we determined a definition for a live citizen science game. Then, with past research and overall goals considered, we formulated the following hypotheses:

- H1: Dynamic player-level matchmaking, using skill ratings and chains, improves player performance compared to baseline task assignment in a live, complex citizen sci-

ence game.

- H2: Dynamic player-level matchmaking, using a moderate fixed desired win rate model and skill ratings and chains, improves performance compared to previous win rate models and baseline task assignment.

Given these hypotheses, we decided to perform our experiment in Foldit, a live and more complex CSG with a large player base (Miller et al. 2021; Curtis 2015). Its tasks involve protein folding, which is often non-intuitive for people without prior biochemistry knowledge, and its puzzles are not as linear as previously-researched CSGs (Miller et al. 2021; Khatib et al. 2011).

Foldit recently released Dojo Mode in order to bridge the gap between relatively simple tutorial puzzles and complex real-life science puzzles (Foldit 2021). In the Dojo, players are continuously presented with puzzles with a goal score; once that score is reached, the player advances to the next level. In each level, players also have an amount of "stamina" which decays over time and is spent by player moves. If the player depletes their stamina, they lose the level.

We tested four player-level matchmaking conditions in the Dojo. The first was the current Dojo method, including player and level ratings, a skill chain, and a logistic desired win rate (DWR) difficulty curve. The second was similar to the current method but used a constant 70% DWR instead of a logistic DWR curve. The third condition was purely random level ordering, as an experimental baseline. The fourth condition presented levels in strictly increasing order by level rating, independent of player rating, as another baseline.

Ultimately, we found that player-level matchmaking has potential to improve player performance in Foldit. Compared to random level ordering, both the logistic and fixed desired win rate models led to more levels completed and a higher peak level completed. However, neither model significantly outperformed the fixed ordering, meaning these results could have either been a product of DDA or merely a static difficulty increase. Similarly, we found that the moderate fixed desired win rate model was effective in encouraging the quantity and quality of completed levels, but not significantly better than the other matchmaking methods examined. Additionally, we noticed that the static increasing difficulty order, which used level ratings learned partly from gameplay data, outperformed random, indicating the level ratings may be useful to refine a designer's initial estimate of level difficulty. We thus contribute to the literature by expanding DDA research into a live, active CSG, a domain previously untouched.

Background

Level Ordering

In an attempt to keep players adequately challenged, games commonly present levels in order of increasing difficulty. This is not necessarily a linear process, as there can be stipulations to reaching the next level(s): completing a proportion of previous levels, as in Baba Is You (Hempuli 2019), or completing a specific sequence of previous levels, as in Portal (Valve 2007). In all cases, players who complete easier levels face gradually more challenging ones as they advance. Notably, these are static approaches, in which a level's difficulty is designer-built and unmodified during gameplay.

Traditionally, CSGs do not necessarily have such an ordering, instead having many puzzles available for players to solve. Even so, the aforementioned static approaches are not easily transferred to CSGs, because the difficulty of their levels is undefined (Cooper, Deterding, and Tsapakos 2016; Logas et al. 2014). By nature, the problems CSGs are designed to solve do not have optimal solutions, so the skill needed to reach that solution isn't measurable. As a result, researchers generally must resort to coarse heuristic methods to approximate level difficulty. For example, researchers examining the CSGs Xylem (Logas et al. 2014) and Binary Fission (Kate et al. 2016) used task size as an estimate for difficulty. However, with many of these heuristics, their effectiveness has not been validated and they may not be translatable to other types of problems. A more generalizable approach to level adaptation would allow for more widespread usage.

Dynamic Difficulty Adjustment

Dynamic difficulty adjustment (DDA) is the process by which a game's difficulty is continually tailored to the player's abilities over the course of play. Similarly to level ordering, this effect is difficult to accomplish in citizen science games, for the same inherent task-related reasons mentioned previously (Logas et al. 2014).

However, one approach to accomplishing DDA in CSGs is player-level matchmaking. This method is based on well-documented rating systems such as Elo (Elo 1978) and Glicko/Glicko-2 (Glickman 1999, 2001), which have been used to measure the skill of players in games ranging from chess and Go (Au 2020) to Pokémon Showdown (Pokemon Showdown 2021) and Counter Strike: Global Offensive (Dehpanah et al. 2021).

Sarkar et al. adapted these rating systems to CSGs to accomplish DDA via player-level matchmaking (Sarkar et al. 2017). In such a system, ratings are given to both players and levels; player ratings represent player skill, while level ratings represent level difficulty. A difficulty curve gives the "desired win rate" the game attempts to deliver for a player of a given rating. The chosen rating system's matchmaking algorithm calculates the expected win rate of giving a player a certain level, based on both entities' ratings. Of the eligible levels, the player-level pairing with the closest to desired win rate (or a random match within an range of win rate) is the "match" that is ultimately served. The outcome of the level is either a "win" or "loss" and the player's and level's ratings are adjusted accordingly before the process repeats for the next match.

By examining existing data sets from chess and the human computation game Paradox, it was shown that the bipartite nature of the player-task graphs did not decrease the quality of produced ratings. In other words, the fact that players are never compared to other players and levels are never directly compared to other levels should not inhibit player-

level matchmaking from being an effective DDA method (Cooper, Deterding, and Tsapakos 2016). In later work, this player-level matchmaking approach was shown to be effective in implementing DDA and increasing player engagement in Paradox (Sarkar et al. 2017). A rating-based matchmaking approach led to significant increases in level attempts and completions compared to random level ordering; linear level ordering strictly in terms of increasing difficulty led to similar results. However, among levels completed, matchmaking led to the completion of much more difficult levels than the linear level ordering.

Analysis of the matchmaking method's difficulty curve was also conducted in Paradox (Sarkar and Cooper 2019). Functions and compositions were used to make eight different difficulty curves ranging from a various logistic curves to two flat models giving a constant desired win probability for players of all ratings. Altering the difficulty curve of the game caused significant changes in player experiences, including the number and difficulty of levels completed, as well as players' perceptions of the game itself. Notably, neither constant function studied (50% and 90% desired win rate) was found to be Pareto efficient for the number of levels completed and the highest level rating played. This possibly suggested the former made early levels too challenging, while the latter made later tasks overly easy. Relatedly, this study omitted a moderate fixed win rate, which could best balance engagement and performance (Lomas et al. 2013).

Skill chains are another separate method of implementing DDA in games. This term refers to a structure consisting of individual skills, or learned in-game mechanics, in the order players acquire them throughout gameplay (Cook 2007). This effectively models skill progression, as skills generally build upon each other (e.g. learning how to run and jump individually are prerequisites to learning a running jump). Each level is assigned the skill(s) necessary to complete it; different levels generally require different sets of skills to complete, and more challenging levels generally require more intricate skills. To best meet players at their skill level, players can only be matched with levels that are appropriate for someone with their set of acquired skills. In practice, players often start with zero skills, and can initially only be matched with levels that require the very first skills in the game's skill tree. As players complete levels, they build their skill tree; as they improve at the game, the pool of available levels increases in size and difficulty. Thus, skill chains can also be used as a DDA technique.

Prior work has combined these approaches—player-level matchmaking and skill chains—to accomplish DDA in human computation games Iowa James and Paradox (Sarkar and Cooper 2020). The quantity and difficulty of completed levels was boosted with the use of skill chains, while the addition of rating systems had different effects based on the game.

However, there has not been work done combining these DDA techniques in a live, online CSG. The games in which previous research has been conducted—Iowa James and Paradox—were not large, active games. Players were recruited mainly through Amazon Mechanical Turk to play the game solely for research purposes. While this does not take away from the quality of the research findings, it does represent a significant difference in the testing environments for DDA methods, as previous work found differences between paid and volunteer recruiting (Sarkar and Cooper 2018). As a result, further experimentation could confirm or refine game recommendations for live CSGs.

Live Citizen Science Games

While there are many CSGs, scientific discoveries primarily originate in a specific subset we define as "live" citizen science games. There are specific criteria that these games meet:

- Presently playable, with an active player base
- Designed to solve novel scientific problems—not merely relating to scientific subjects (i.e. simulations games do not apply)
- Minimal, if any, required preexisting scientific knowledge to play
- Intrinsically motivated players—players are not externally recruited and do not receive compensation or grades
- Available to the general public—not limited access

This definition is intentionally left slightly open due to the variance within the field of CSGs. In particular, there is no quantitative measure provided for the player base, because "active" can vary significantly between games and any concrete value would be arbitrary. Nonetheless, because they embody the overarching scientific purpose of CSGs, these live CSGs warrant special research consideration.

Foldit

Foldit is a citizen science game focused on protein folding and design, a complex process in which human intuition has been shown to be useful (Koepnick et al. 2019; Eiben et al. 2012). It is a free-to-play, downloadable game that has been available to the general public since its 2008 release. Each level in Foldit is a protein structure. Players use a variety of tools to modify the structure. Examples include "wiggle," which automatically optimizes the protein backbone, "shake," which automatically optimizes the side chains, and manual manipulation of the protein. Foldit uses the Rosetta molecular modeling suite to score proteins, with higher scores indicating lower energy states, and thus better-folded structures (Rohl et al. 2004). Some levels have a predetermined target score as a goal, while others are organized as player high-score competitions.

Methods

This experiment was conducted in Foldit's Dojo mode. This is the game's "endless" mode, in which players complete levels continuously as a form of training for science puzzles. A screenshot of a Dojo puzzle is shown in Figure 1.

Dojo Dynamic Difficulty Adjustment

We used player-level matchmaking and skill chains as forms of DDA. Players were initialized with a Glicko-2 rating of 1, which increased with a win and decreased with a loss.

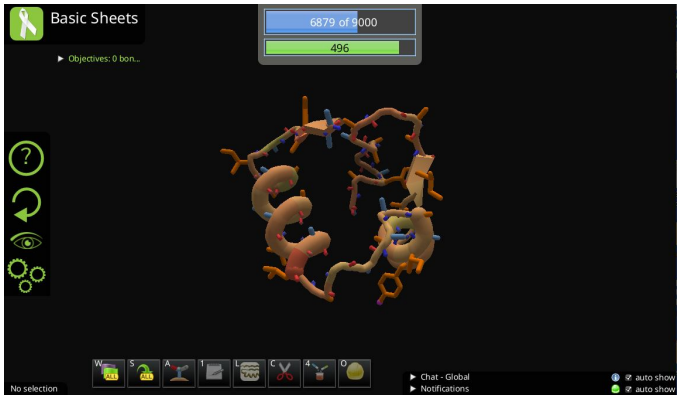

Figure 1: An example of a Foldit Dojo puzzle screen.

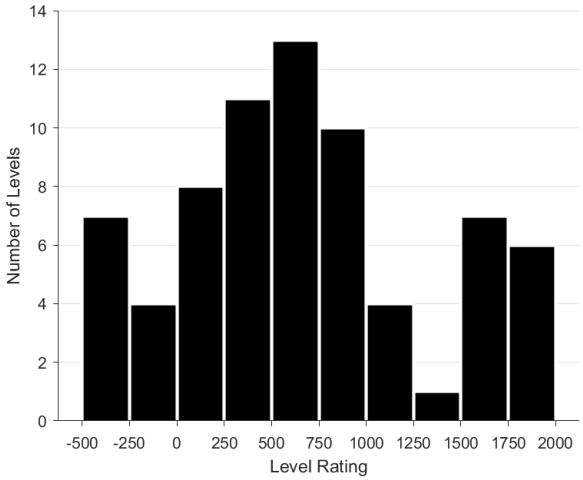

Figure 2: A histogram of the ratings of the 71 Foldit levels used in Dojo mode. The level ratings ranged from -449.2 to 1900.

Level ratings were initially set heuristically by a designer, and then modified by prior gameplay data using the matchmaking system (increased following a player loss, decreased following a player win). The distribution of level ratings used in this experiment is shown in Figure 2.

Dojo matchmaking incorporated skill chains in a manner similar to a 2020 experiment by Sarkar and Cooper (Sarkar and Cooper 2020). Levels were manually assigned skills on the Foldit skill chain deemed necessary to complete them. Skills in the Foldit skill chain included "cut" and "idealize" (referring to Foldit tools), as well as "hydro" and "sheets" (referring to relevant scientific concepts). The pool of available levels is determined by first excluding levels that the player has already completed, and then collecting all levels whose skills are one skill chain link away from skills the player possesses. If no such levels existed, the search was expanded to two skill chain links, and so on, until matches could be made.

Given that pool of eligible levels, the system then calculated the match's desired win rate given the player's rating. Then, among all possible player-level pairings, the "match" whose Glicko-2-calculated expected win rate is closest to the desired value is served to the player.

A total of 71 Foldit puzzles were used for the Dojo level pool. Dojo puzzles are not science puzzles themselves, but special puzzles with arbitrary "win" conditions (scores), because they are training levels, rather than science puzzles. Additionally, each level gave players a fixed amount of "stamina," which decayed over time and is spent by player moves. If the player depleted their stamina, they lost the level. Otherwise, if they folded the protein well enough to surpass the target score, they won the level.

Experimental Setup

Methods had approval by the appropriate Institutional Review Board. 221 players participated in the experiment. All participants were Foldit players who created accounts after the experiment began. Such players who entered Dojo mode and completed one level were included in the analysis, except one player who had logging errors preventing inclusion. We used no recruitment methods outside of Foldit itself. Once players completed the basic part of the tutorial, we displayed a popup containing the text "Test your skills in the Dojo!" and an optional button to enter Dojo mode. Players could also enter Dojo mode directly from the main menu.

The first level was the same for all players. After winning, losing, or giving up on a level, players were shown the number of moves and amount of time spent on that level, above a button leading to the next puzzle. Players were given Dojo levels until they reached three losses or decided to exit. Players could re-enter Dojo mode from the main menu at any time.

Dojo players were placed randomly into one of four experimental conditions, in which matchmaking method was varied:

1. LOGISTIC - Existing Dojo matchmaking, with a logistic desired win rate curve based on player rating r described below, so that the desired win rate goes down as the player's rating goes up.

$$W(r) = 1 - \frac{1}{1 + e^{-0.0110413(r-200)}}$$

2. FIXED - Modified version of existing Dojo matchmaking, using a constant 70% desired win rate.

$$W(r) = 0.7$$

3. RANDOM - Matches are made randomly among the set of levels not already completed.
4. LINEAR - Levels are presented in increasing order of Glicko-2 rating.

The LOGISTIC and FIXED conditions represented the dynamic difficulty adjustment being examined in this paper. The RANDOM and LINEAR conditions serve as baselines for player engagement.

Variable	Result
Time Spent	$p < 0.05$, H(3) = 10.12
Levels Completed	$p < 0.001$, H(3) = 44.77
Highest Rating	$p = 0.1264$, H(3) = 5.71
Highest Level Completed	$p < 0.001$, H(3) = 28.60

Table 1: Summary of omnibus statistical analysis results.

The parameters for the LOGISTIC and FIXED conditions were selected based on motivating research and gaps in past studies (Sarkar and Cooper 2019). The LOGISTIC parameters were chosen such that the player's first match will have a 90% desired win rate, and a match has a desired win rate of 50% when r=200. Previous studies examined fixed 50% and 90% desired win rate models, so we opted for a more moderate 70% desired win rate as our FIXED condition.

During the experiment, all puzzle ratings were frozen at their current state, rather than being adjusted by each match. This was done so that level ratings would be the same for all players during the experiment, and since variable level ratings may increase the noise of the data collected.

For each Dojo puzzle, we recorded the amount of time spent, the number of moves the player used, the final score the player reached on the puzzle, and the ultimate result of the level, a win or loss. From this information, we analyzed the following variables:

- Time Spent: Sum of all of the individual Dojo puzzle times for a given player, in seconds.
- Levels Completed: The number of levels for which a given player reached the target score.
- Highest Rating: The highest rating a given player reached at any point in their Dojo runs.
- Highest Level Completed: The highest rating among all levels completed by a given player.

Because the data were not normally distributed, we used non-parametric tests for analysis. For all variables, we determined whether there were significant differences among all four conditions using an omnibus Kruskal-Wallis test. If a significant difference was found, we performed post-hoc pairwise Wilcoxon rank-sum tests with the Holm correction to find pairwise significant differences. We chose a significance threshold of $\alpha = 0.05$.

Results

Box plots of all results are shown in Figure 3. We found significant differences among the four conditions in three of the four variables examined. Omnibus test results are summarized in Table 1. For Time Spent, Levels Completed, and Highest Level Completed, omnibus Kruskal-Wallis tests gave p-values lower than 0.05, with the latter two variables having $p < 0.001$. Post-hoc analysis was completed for these three variables.

Pairwise analysis results are summarized in Table 2. For Time Spent, the only significant pairwise comparison was between LOGISTIC and RANDOM conditions, in which LOGISTIC outperformed RANDOM.

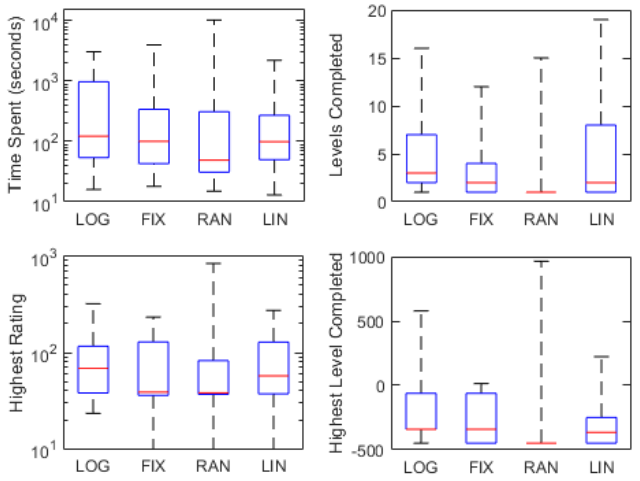

Figure 3: Box plots of the results for each variable and condition. Note the logarithmic axes for Time Spent and Highest Rating. The minimum Highest Rating value for FIXED, RANDOM, and LINEAR were all 1.

For Levels Completed, we found the RANDOM condition to be significantly less than all three other conditions. We also saw that LOGISTIC led to significantly more levels completed than the FIXED condition.

For Highest Level Completed, we also found the RANDOM condition to be significantly lower than all three other conditions.

Discussion

Our first hypothesis was that dynamic player-level matchmaking, using skill ratings and chains, improves player performance compared to baseline task assignment in a live, complex citizen science game. To address this, we looked at comparisons between the matchmaking-based methods and two baseline methods: LOGISTIC and FIXED vs. LINEAR and RANDOM, respectively. Overall, our results indicate that LOGISTIC significantly outperformed RANDOM in Time Spent, and both LOGISTIC and FIXED significantly outperformed RANDOM in Levels Completed and Highest Level Completed. However, neither matchmaking-based method outperformed the LINEAR condition in any metric. Therefore, LOGISTIC and FIXED were better than random assignment, but not better than a static level ordering, so H1 is partially supported.

This result indicates that the success of player-level matchmaking has the potential to translate to live citizen science games. Dynamic level presentation can encourage players to complete more levels, as well as complete levels of higher difficulty, despite not playing the game for tremendously longer. This did not lead to greater objective progress through the Dojo, though, as their peak ratings were not significantly higher. This could be a result of matchmaking correctly meeting players at their current skill level, and thus best using and improving their skill set, as intended. However, because similar results were seen for the static

Variable	Time Spent (Seconds)	Levels Completed	Highest Level Completed
LOGISTIC / FIXED	122.5 / 102 $p = 0.73$	3 / 2 $p < 0.05$	-341.7 / -341.7 $p = 0.26$
LOGISTIC / RANDOM	122.5 / 49 $p < 0.05$	3 / 1 $p < 0.001$	-341.7 / -449.2 $p < 0.001$
LOGISTIC / LINEAR	122.5 / 99.5 $p = 0.73$	3 / 2 $p = 0.48$	-341.7 / -365.3 $p = 0.17$
FIXED / RANDOM	102 / 49 $p = 0.20$	2 / 1 $p < 0.001$	-341.7 / -449.2 $p < 0.01$
FIXED / LINEAR	102 / 99.5 $p = 0.99$	2 / 2 $p = 0.07$	-341.7 / -365.3 $p = 0.87$
RANDOM / LINEAR	49 / 99.5 $p = 0.12$	1 / 2 $p < 0.001$	-449.2 / -365.3 p < 0.001

Table 2: Summary of pairwise statistical analysis results. First line contains median values. Second line contains degree of significance. Shaded cells represent pairwise comparisons deemed significant.

level ordering, this performance improvement could also be a product of the player-level matchmaking merely increasing the difficulty of levels as players advanced in the Dojo. Nonetheless, because CSGs often obtain their scientific advancements from the results of challenging tasks, player-level matchmaking represents an opportunity to optimally harness players' abilities to contribute to the games' overall goals. Because the combined usage of matchmaking and skill chains saw improvements in performance, potential inclusion of these methods alongside other DDA techniques could provide an even better-tailored player experience.

Our second hypothesis was that dynamic player-level matchmaking, using a moderate fixed desired win rate model and skill ratings and chains, improves performance compared to previous win rate models and baseline task assignment. To address this, we compared the FIXED condition and all three other conditions. Overall, our results showed that FIXED significantly outperformed RANDOM in Levels Completed and Highest Level Completed, but did not outperform the LOGISTIC and LINEAR conditions. Therefore, FIXED was better than one of the two baselines, but not the other experimental group, so H2 is partially supported.

These results indicate that the fixed desired win rate model is not a clear-cut improvement on alternative difficulty curves. Relative to random level ordering, the fixed model accomplished its goal of improving player performance, but failed to distinguish itself as a particularly beneficial method of DDA. In fact, compared to the logistic model, its experimental counterpart, the fixed desired win rate model performed slightly worse. Because the logistic difficulty curve translated to significantly more levels completed, it could be a marginally superior method for improving sheer task volume via player-level matchmaking. As with player-level-matchmaking in general, as described previously, this approach could be moderately effective because it simply increases level difficulty in any capacity as players advance. Alternatively, it could provide too rigid of a model for players on extreme ends of the skill spectrum. A 70% win rate could be too challenging for a player's first level but far too easy for an expert; this would be an issue inherent to all fixed win rate models, no matter what constant value is selected. As a result, fixed desired win rate matchmaking models likely should not be the sole option considered for player-level matchmaking, but still warrant consideration given their relative success.

Although not part of one of our hypotheses, comparing the two baselines RANDOM and LINEAR shows that the static increasing difficulty order outperforms random ordering. In particular, LINEAR led to a significantly greater number of levels completed and highest level completed, as well as over twice as long spent playing the game. LINEAR used an ordering based on level ratings that were initially set by a designer and refined by gameplay data. This indicates level ratings may be useful to refine a designer's initial estimate of level difficulty.

Limitations and Future Work

Although this study sought to examine how matchmaking involving a skill chain affects DDA, the skill chain used in this study was constructed manually. Rather than using previous player data determine the manner in which skills are organized in the skill chain, the structure was manually created by the developers. This lack of a player-informed skill chain could impact player-level matchmaking's overall effect on assigning engaging tasks. For example, a game designer could inaccurately identify a vital early-game skill as one "learned" later in gameplay, making levels too challenging.

Additionally, the Foldit Dojo had minimal level variability for players in their initial stages of gameplay. There were only 11 levels with ratings below 0, which would comprise the first levels presented to players. If the difficulty or requisite skills for these levels was inaccurate, players could have been poorly paired with these levels by the matchmaking methods.

Finally, this experiment involved "freezing" post-match level rating updates. This equalized the set of levels for all players, making analysis more robust, but in the live Dojo, a particular level's rating may vary for each player. As a result, the results of players in this experiment did not inform

the future ratings of levels, limiting the overall player-level matchmaking process.

Future work in this domain can alter the aforementioned details. Allowing levels' ratings to be updated after "wins" and "losses" would permit dynamic difficulty adjustment more similar to that of a live game, similar to a recent study (Sarkar and Cooper 2021). While analysis could potentially be more challenging, experiments modeling live games more accurately are vital for ultimate application of player-level matchmaking to CSGs. Furthermore, incorporating player-driven skill chains would further tailor the gameplay experience to a particular player. Dynamically-updated skill chains based on real-time results would be one possible method of combining these ideas going forward.

Continued research could also delve further into these matchmaking approaches' effect on player strategy. While little of that analysis was performed in this experiment, outside of measuring time spent on levels, logging of player actions by type and level would be insightful into the manner in which the different matchmaking methods change players' approaches to the levels they are presented. Further observations could be made via a direct feedback form to which players identify the tools they deem most useful and any particular approaches they find useful.

While we did not gather data to compare using LINEAR to using the original designer-set ratings in increasing order, comparison of different static orderings could be an interesting area for future work.

Wider-scale analysis of difficulty curves would aid in optimizing the player-level matchmaking experience. Performing a similar experiment with additional varied difficulty curves—spanning exponential, linear, and other flat or logistic models—would provide a basis upon which games can decide how to best mold their player-level matchmaking methods to their desired player experience—e.g., as in (Sarkar and Cooper 2019).

Finally, within the field of machine learning, the strategy of curriculum learning—in which a model is trained using gradually more challenging data—is worth considering in the context of DDA in live CSGs (Wang, Chen, and Zhu 2020). Previous work in Foldit used Stratabots specifically deigned for educational games to model players of varying skill levels (Horn et al. 2018). The possible interplay of these fields, in which models are more effectively trained with the help of dynamic difficulty adjustment, represents an exciting opportunity for even more scientific advancement.

Conclusion

In this paper, we defined a live citizen science game and examined the effects of dynamic difficulty adjustment via player-level matchmaking and skill chains in one such game, Foldit. Our results showed that DDA methods were effective in improving the number of levels completed, as well as the difficulty of the most challenging level completed, compared to a baseline random ordering, but not better than a static level sequence. Likewise, the moderate fixed desired win rate model improved player performance by the same two metrics, but not significantly greater than the logistic matchmaking or static order approaches.

All in all, this paper contributes an analysis of DDA methods to a newer, more practical domain—live citizen science games. This specific category of games warrant special research consideration, and our results indicate that there is potential for use of player-level rating and matchmaking within them. The approach used in this experiment is generalizable to all games with "wins" and "losses," as well as skill chains. While there is no clear best option among these approaches, the dynamic matchmaking methods saw significant improvements in the metrics that drive the scientific progress made by CSGs.

Future work in this area can better model a true live game environment by maintaining dynamic level ratings, or incorporate additional DDA techniques to seek even greater performance improvements. Inquiries into the general effects of additional difficulty curves and level orderings, as well as the impact of player-level matchmaking on player strategy, are similarly worth consideration.

Acknowledgments

This material is based upon work supported by the National Science Foundation under grant nos. 1652537 and 1950697. We thank all of the players of Foldit.

References

Au, R. 2020. Making Fair Games of Go. *Counting Stuff* (Substack). https://counting.substack.com/p/making-fair-games-of-go.

Bonney, R.; Shirk, J. L.; Phillips, T. B.; Wiggins, A.; Ballard, H. L.; Miller-Rushing, A. J.; and Parrish, J. K. 2014. Next Steps for Citizen Science. *Science*, 343(6178): 1436–1437.

Burnett, S.; Furlong, M.; Melvin, P. G.; and Singiser, R. 2016. Games that Enlist Collective Intelligence to Solve Complex Scientific Problems. *Journal of Microbiology & Biology Education*, 17(1): 133–136.

Cook, D. 2007. The Chemistry of Game Design. *Game Developer*. https://www.gamedeveloper.com/design/the-chemistry-of-game-design.

Cooper, S.; Deterding, S.; and Tsapakos, T. 2016. Player Rating Systems for Balancing Human Computation Games: Testing the Effect of Bipartiteness. In *Proceedings of the First International Joint Conference of DiGRA and FDG*.

Curtis, V. 2015. Motivation to participate in an online citizen science game: A study of Foldit. *Science Communication*, 37(6): 723–746.

Cusack, C.; Largent, J.; Alfuth, R.; and Klask, K. 2010. Online games as social-computational systems for solving NP-complete problems. *Meaningful Play*.

Dehpanah, A.; Ghori, M. F.; Gemmell, J.; and Mobasher, B. 2021. Evaluating Team Skill Aggregation in Online Competitive Games. *2021 IEEE Conference on Games (CoG)*, 01–08.

Eiben, C.; Siegel, J.; Bale, J.; Cooper, S.; Khatib, F.; Shen, B.; Eccles, D.; Stoddard, B.; Popovic, Z.; and Baker, D. 2012. Increased Diels-Alderase Activity through Foldit Player Guided Backbone Remodeling. *Nature biotechnology*, 30: 190–2.

Elo, A. E. 1978. *The Rating of Chessplayers, Past and Present*. New York: Arco Publishing.

Entertainment Software Association. 2021. 2020 Essential Facts About the Video Game Industry. https://www.theesa.com/resource/2020-essential-facts/. Accessed: 2021-12-20.

Foldit. 2021. Introducing Dojo Mode! https://fold.it/portal/node/2011602. Accessed: 2021-12-21.

Glickman, M. E. 1999. Parameter Estimation in Large Dynamic Paired Comparison Experiments. *Journal of the Royal Statistical Society Series C*, 48(3): 377–394.

Glickman, M. E. 2001. Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics*, 28(6): 673–689.

Hempuli. 2019. Baba Is You. https://hempuli.com/baba/. Accessed: 2022-04-13.

Horn, B.; Miller, J. A.; Smith, G.; and Cooper, S. 2018. A Monte Carlo Approach to Skill-Based Automated Playtesting. *Proc AAAI Artif Intell Interact Digit Enterain Conf*, 2018: 166–172.

Jennett, C.; Kloetzer, L.; Schneider, D.; Iacovides, I.; Cox, A.; Gold, M.; Fuchs, B.; Eveleigh, A.; Mathieu, K.; Ajani, Z.; and Talsi, Y. 2016. Motivations, learning and creativity in online citizen science. *Journal of Science Communication*, 15.

Kate, C.; Heather, L.; C., O. J.; Chandranil, C.; Kelsey, C.; Daniel, F.; Dylan, L.-E.; Zhongpeng, L.; Jo, M.; Afshin, M.; Johnathan, P.; Husacar, S.; Jim, W.; and Brenda, L. 2016. Design Lessons From Binary Fission: A Crowd Sourced Game for Precondition Discovery. In *Proceedings of the First International Joint Conference of DiGRA and FDG*.

Khatib, F.; Cooper, S.; Tyka, M. D.; Xu, K.; Makedon, I.; Popović, Z.; and Baker, D. 2011. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47): 18949–18953.

Koepnick, B.; Flatten, J.; Husain, T.; Ford, A.; Silva, D.-A.; Bick, M. J.; Bauer, A.; Liu, G.; Ishida, Y.; Boykov, A.; Estep, R. D.; Kleinfelter, S.; Nørgård-Solano, T.; Wei, L.; Players, F.; Montelione, G. T.; DiMaio, F.; Popović, Z.; Khatib, F.; Cooper, S.; and Baker, D. 2019. De novo protein design by citizen scientists. *Nature*, 570(7761): 390–394.

Larche, C. J.; and Dixon, M. J. 2020. The relationship between the skill-challenge balance, game expertise, flow and the urge to keep playing complex mobile games. *Journal of Behavioral Addictions*, 9(3): 606–616.

Logas, H.; Whitehead, E. J.; Mateas, M.; Vallejos, R.; Scott, L.; Shapiro, D. G.; Murray, J. T.; Compton, K.; Osborn, J. C.; Salvatore, O.; Lin, Z.; Sánchez, H. A.; Shavlovsky, M.; Cetina, D.; Clementi, S.; and Lewis, C. 2014. Software verification games: Designing Xylem, The Code of Plants. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*.

Lomas, D.; Patel, K.; Forlizzi, J. L.; and Koedinger, K. R. 2013. Optimizing Challenge in an Educational Game Using Large-Scale Design Experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 89–98.

Miller, J. A.; Horn, B.; Guthrie, M.; Romano, J.; Geva, G.; David, C.; Sterling, A. R.; and Cooper, S. 2021. How do Players and Developers of Citizen Science Games Conceptualize Skill Chains? *Proceedings of the ACM on Human-Computer Interaction*, 5: 1–29.

Pokemon Showdown. 2021. How the ladder works. https://pokemonshowdown.com/pages/ladderhelp. Accessed: 2022-01-07.

Rallapalli, G.; Players, F.; Saunders, D. G.; Yoshida, K.; Edwards, A.; Lugo, C. A.; Collin, S.; Clavijo, B.; Corpas, M.; Swarbreck, D.; Clark, M.; Downie, J. A.; Kamoun, S.; Cooper, T.; and MacLean, D. 2015. Lessons from Fraxinus, a crowd-sourced citizen science game in genomics. *eLife*, 4: e07460–e07460. 26219214[pmid].

Rohl, C.; Strauss, C.; Misura, K.; and Baker, D. 2004. Protein structure prediction using ROSETTA. *Methods in enzymology*, 383: 66–93.

Sarkar, A.; and Cooper, S. 2018. Comparing Paid and Volunteer Recruitment in Human Computation Games. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*.

Sarkar, A.; and Cooper, S. 2019. Transforming Game Difficulty Curves Using Function Composition. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–7.

Sarkar, A.; and Cooper, S. 2020. Evaluating and Comparing Skill Chains and Rating Systems for Dynamic Difficulty Adjustment. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 16(1): 273–279.

Sarkar, A.; and Cooper, S. 2021. An Online System for Player-vs-Level Matchmaking in Human Computation Games. In *2021 IEEE Conference on Games (CoG)*, 1–4.

Sarkar, A.; Williams, M.; Deterding, S.; and Cooper, S. 2017. Engagement Effects of Player Rating System-Based Matchmaking for Level Ordering in Human Computation Games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

Sweetser, P.; and Wyeth, P. 2005. GameFlow: a model for evaluating player enjoyment in games. *Comput. Entertain.*, 3: 3.

Valve. 2007. Portal. https://www.thinkwithportals.com/index.php. Accessed: 2022-04-13.

Von Ahn, L.; and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 319–326.

Wang, X.; Chen, Y.; and Zhu, W. 2020. A Comprehensive Survey on Curriculum Learning. *CoRR*, abs/2010.13166.