Evaluating and Comparing Skill Chains and Rating Systems for Dynamic Difficulty Adjustment

Anurag Sarkar and Seth Cooper
Northeastern University
sarkar.an@northeastern.edu, se.cooper@northeastern.edu

Abstract

Skill chains define how in-game skills build on each other and the order in which players ideally acquire them during gameplay. This can enable dynamic difficulty adjustment (DDA) by serving levels based on the skills that players currently have and those required to solve a given level. Similarly, DDA can also be achieved by using rating systems to match players with suitable levels by assigning ratings to players and levels based on ability and difficulty respectively. However, the relative effects of these two methods remain unclear, particularly in the context of human computation games (HCGs). In this paper, we present a general model for using skill chains and rating systems in a combined DDA system along with an evaluation comparing the two for difficulty balancing within HCGs, focusing on the relative merits of both methods when used separately as well as together. We evaluate our methods using the HCGs *Iowa James* and *Paradox*. Our findings suggest that incorporating skill chains can improve upon previously shown benefits of using only rating systems for DDA in HCGs.

Introduction

Dynamic difficulty adjustment (DDA) refers to techniques for dynamically modifying in-game difficulty in response to player ability. Much prior work has focused on various DDA techniques including player modeling (Zook and Riedl 2012), adjusting level design (Valve Corporation 2008), parameter tuning (Hunicke 2005) and machine learning (Jennings-Teats, Smith, and Wardrip-Fruin 2010). However, implementing DDA in human computation games (HCGs) poses a unique challenge. HCGs seek to harness collective player ability to help solve computationally complex tasks by representing them as game levels and have been used in various domains such as image labeling (von Ahn and Dabbish 2004) and protein design (Koepnick et al. 2019). However, since levels represent real problems, they can't be readily modified for DDA without potentially compromising how well the problems are modeled. Previous works have tried overcoming this by framing difficulty-based level ordering as a player-versus-level matchmaking problem i.e. using rating systems like Elo (Elo 1978) and Glicko-2 (Glickman 2001) to match players of a certain ability with levels of comparable difficulty. Since it balances difficulty by adjusting level ordering rather than modifying levels themselves, this ratings-based approach is useful for DDA in HCGs (Cooper, Deterding, and Tsapakos 2016).

Iowa James Paradox

Figure 1: Screenshots from the two games used in this work.

However, though prior work has shown the effectiveness of rating systems for DDA and player engagement in HCGs (Sarkar et al. 2017), such systems are not informed about the skills that players acquire during gameplay, nor about the skills needed to complete any given level. Here, 'skills' refers to the discrete mechanics of a game. Thus, the order of skill acquisition is often used by designers to help define difficulty progressions. In doing so, designers leverage the game's skill chain which defines how complex in-game skills build on simpler ones during gameplay. Hence, the skillset required to solve easier levels is a subset of that required to solve harder ones. In this way, based on the skills needed to complete levels, designers can tailor in-game difficulty in accordance with the skills acquired by the player.

Thus, recent work (Sarkar and Cooper 2019b) applied a hybrid model that incorporated skill chains into the ratings-based DDA approach for HCGs. Here, the game's skill chain was used to define different hierarchies of levels based on the skills needed to solve them and ratings were then used to decide which level in a selected hierarchy to serve to the player. The combined model was effective in demonstrating that DDA systems for HCGs can leverage both skill chains and rating systems. However, the model focused on HCGs where game mechanics and tasks required separate sets of abilities and tracked player ability for these independently. Thus, the

use of skill chains for DDA in HCGs in general, where mechanics and tasks are more interdependent, remains unexplored, as does the particular effects of the interaction between skill chains and rating systems in a combined model. Therefore, in this work, we use two different HCGs—the puzzle game *Paradox* and the platformer *Iowa James*, shown in Figure 1—to study variations of this DDA model that use skill chains and rating systems separately as well as taken together in order to evaluate the relative effects and interactions of the two components. This work contributes a general approach for combining skill chains and rating systems for DDA in HCGs and a comparative study evaluating the relative merits of each of these two components.

Background

Skill Chains and Rating Systems Skill chains (Cook 2007) define the order in which players acquire skills during gameplay. They consist of *skill atoms* which represent individual skills with simpler atoms feeding into more complex ones in the chain. Such skill chains can be seen as directed graphs where nodes and edges represent skills and skill dependencies respectively. An ordering of such a graph thus defines a sequence by which players can acquire skills and can thus help in the design of level progressions. Skill chains can also be incorporated into existing games to improve their progression as shown by Echeverria et al. (2012) who redesigned a physics educational game using this approach. In HCGs, Horn et al. (2018) used skill chains to design AI agents of varying abilities to analyze difficulty progressions in the puzzle HCG *Foldit*.

Like skill chains, rating systems can also help define level progressions by being reformulated as Player-vs-Level (PvL) rather than Player-vs-Player (PvP) as in chess and MOBAs. This allows ratings to represent ability for player and difficulty for levels and are computed using PvL outcomes during gameplay. In turn, players are served levels compatible with their current abilities, thus enabling DDA. Prior work (Sarkar et al. 2017) has shown the effectiveness of such systems, like Glicko-2 (Glickman 2001), in defining level progressions for DDA in HCGs. However, while effective in improving engagement, such systems do not let designers control level ordering without changing the formulation for determining a good match between player and level (Sarkar and Cooper 2019a). Designers may wish that players learn skills and/or concepts in a certain order while still being able to reap the DDA benefits afforded by ratings. To this end, more recent work (Sarkar and Cooper 2019b) introduced a unified skill model that combined skill chains with rating systems; using the former to define hierarchies of levels requiring the same skillset and using the latter to determine which level in a given hierarchy is the most suitable match given the player's current ability. This model enables players to acquire skills in a desired order while still serving them levels dynamically as determined by the rating system.

Joint and Disjoint Design in HCGs The combined DDA model from Sarkar and Cooper (2019b) that we adapt is originally applicable to HCGs with a *disjoint* design i.e. game mechanics are unrelated to the underlying task. Such games include *OnToGalaxy* (Krause et al. 2010), a space shooter for populating ontologies; the *Landspotting* games (Sturn et al. 2013) about labeling land cover data, spanning strategy, tagging and tower defense mechanics; and *Gwario* (Siu, Guzdial, and Riedl 2017), a *Super Mario Bros.*-style (Nintendo 1985) platformer for item collection. Since such games decouple game and task mechanics, a model that separately tracks player ability in performing game mechanics and executing tasks is necessary for proper DDA along both dimensions. However, such a model is not applicable to HCGs where level design centers around the underlying problem to be solved, causing game mechanics to be tightly coupled with the modeled task. For the hybrid skill chain and ratings-based model to be applicable to such HCGs, we simplify the previously disjoint model in this work.

Learner Models and Progression Design Learner models of skill and knowledge have long been applied in educational software and games (Desmarais and Baker 2012; Khenissi et al. 2015; Harpstead and Aleven 2015). Techniques include Bayesian Knowledge Tracing (Baker, Corbett, and Aleven 2008; Yudelson, Koedinger, and Gordon 2013), which attempts to probabilistically model learner acquisition of skills, and Item Response Theory (Baker and Kim 2004), a psychometric approach to testing. Many learner models incorporate hierarchical representations to relate concepts along with performance evaluation (Millán and Pérez-de-la Cruz 2002; Guzmán, Conejo, and Pérez-de-la Cruz 2007). While such models have long been applied in education, our work applies variations of related models to dynamic difficulty in the human computation domain.

In particular, the model of skill chain-based level progression and ratings-based DDA used in our work is similar to the system of Mu et al. (2018) that combines curriculum generation and adaptive problem selection, with the former and latter being analogous to progression generation and DDA respectively. Their work builds on that by Wang, He, and Andersen (2017) which describes a framework for progression design and knowledge assessment. These works are similar to our model but focus on foreign language learning rather than HCGs and utilize execution traces and reinforcement learning rather than skill chains and rating systems.

Games

We used two HCGs: *Iowa James: Hunter Collector Gatherer*, used to evaluate the disjoint skill model in Sarkar and Cooper (2019b), and *Paradox*, an HCG that has been used in much prior work involving rating-system based DDA but not used to test skill chains. They are described below.

Iowa James *Iowa James* is a 2D platformer HCG where players have to collect items while traversing the levels of the game, similar to the HCG *Gwario* (Siu, Guzdial, and Riedl 2017). The game used 75 levels with each level consists of hazards, collectible items and a treasure chest at the end. Items in a level are associated with a specific scenario as indicated by a banner on the top right. To progress to the next level, players have to unlock the treasure chest by collecting items relevant to the given scenario and then reach the chest by navigating the layout of the level and avoiding the hazards. Players have three lives for each level and lose a life each time they collect an irrelevant item or come into

contact with a hazard. If the player is killed by a hazard, they respawn at the beginning of the level. For matchmaking purposes, the player's score for a level was proportional to the number of lives they had upon moving to the next level. The skills in the game are based on typical platformer mechanics and are described below:

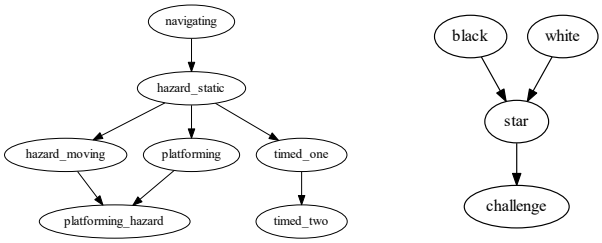
- *navigating (N)*: standard running and jumping-based movement in platformers
- *hazard-static (HS)*, *hazard-moving (HM)*: jumping over stationary and moving hazards respectively
- *platforming (P)*: traversing across platforms
- *platforming-hazard (PH)*: traversing across platforms containing hazards
- *timed-one (T1), timed-two (T2)*: crossing timed hazards spanning shorter and longer distances respectively

Paradox *Paradox* is a 2D puzzle HCG where levels consist of graph-like structures representing Boolean MAX-SAT problems with nodes and edges corresponding to variables and constraints respectively. Player score is based on the percentage of constraints of the underlying problem that players are able to satisfy. This is done by assigning Boolean values to variables using various tools. A player wins a level by reaching its target score. For this work, we used a new Unity version of the game based on a version used in prior research (Sarkar et al. 2017). Past versions used a fixed order for tutorial levels and a dynamic order for non-tutorial challenge levels. In this version, tutorial levels are also served dynamically. Game mechanics involve using two brushes—*white (W)* and *black (B)*—to assign true and false values to a single variable node as well as an advanced *star (S)* brush that assigns values to groups of nodes by running a MAX-SAT solver. We added a fourth *challenge (C)* skill to all non-tutorial levels so that in progressions using the skill chain, players would first encounter the *star* skill in the tutorial. Although the *challenge* skill may be somewhat artificial, it does demonstrate designers' ability to help shape progressions via the skill chain. *Paradox* used 47 levels, out of which 40 were non-tutorial challenge levels.

Method

The combined DDA model introduced in Sarkar and Cooper (2019b) involves 3 stages: 1) defining the game's skill chain 2) annotating levels with required skills and assigning levels an initial rating and 3) using the skill chain and assigned ratings to serve levels via matchmaking. In the following sections, we describe how each of these work, how we modified and extended the model beyond disjoint HCGs, and how we used it to define the different progressions that enable comparing skill chains and rating systems for DDA.

Skill Chain Definition The first step is to define the game's skill chain. This involves identifying the individual skills that players acquire during gameplay along with the dependencies between these skills. Conceptually, this takes the form of a directed graph where nodes and edges represent skills and dependencies respectively. An edge from skill A to skill B indicates that skill B is dependent on skill A, i.e. players must acquire skill A before they can acquire skill B. In other words, if a player is able to perform skill B, they can perform A as well. For this work, we manually defined the skill chain for each game based on our knowledge of their gameplay. In this study, we use only one skill chain per game. For the non-disjoint HCG *Paradox*, this is straightforward since mechanics are directly related to the task. For the disjoint HCG *Iowa James*, we opted to use the game skill chain from our prior work since that accounts for greater variation in gameplay and induces a more distinct sense of progression (e.g. the skill of *platforming* building on the skill of *navigating* as opposed to the skill of *collecting 5 items* building on that of *collecting 3 items* as was the case with the task skill chain). The skill chain we used for each game in this study is shown in Figure 2.



Iowa James Paradox

Figure 2: Skill chains for each game used in this work.

Skill Annotation and Rating Assignment for Levels The next step is informing the DDA system about the required skills and difficulty of each level in the game. For matchmaking, each level is associated with a list of skills needed to complete it and a Glicko-2 (Glickman 2001) rating indicating its difficulty. Thus, prior to using the system, we manually annotate each level with the skills required to finish it. These are taken from the skill chain defined in the previous step. In this work, each level was assigned a rating based on previous trials using the two games and the Glicko-2 rating system. For *Iowa James*, these ranged from 1496 to 1927 while for *Paradox* these ranged from 439 to 1783. These ratings are indicative of the difficulty of each level and were kept fixed over the course of gameplay while player ratings were updated.

Skill and Ratings-based Matchmaking Once the levels have been annotated and initialized with their required skills and Glicko-2 ratings, they can be used for matchmaking. Each player is assigned a starting rating of 1500 and an initial empty set of skills. To serve a level to a player, the system first determines the set of eligible levels based on the player's acquired skills, chooses which among those to serve based on the player's rating and then updates the player's skills and ratings based on the match outcome.

To determine the levels eligible to be served, we first filter out levels that the player has already beaten and the previous level that the player encountered. Among the remaining levels, we mark as eligible all levels that require one or fewer additional skills not yet acquired by the player. If such a level is not found, the system looks for levels with two or fewer additional skills and so on until an eligible level is found. If

the player has acquired all the skills but there are still unplayed levels, all such levels are eligible to be served.

From the eligible levels, the specific level to serve is chosen using the current player rating. Since ratings correspond to competence and difficulty for player and level respectively, comparing them gives an estimate of how hard that level will be for that player. Using the player's rating, we compute their desired loss rate (DLR) which is the desired probability of losing that we want them to have based on their current rating. As the player's rating goes up, so does their DLR and hence they are matched with harder levels, thus achieving DDA. From among the eligible levels determined using the player's skills, we serve the one against which the player's loss probability is closest to their DLR. This loss probability is computed using the Glicko-2 expectation function (Glickman 2001). The DLR is given by the equation $\mathrm{DLR}(r) \approx 1/(1 + e^{0.00628(1850-r)})$, which gives starting players a low 10% chance of losing. More details of this method are given in (Sarkar and Cooper 2019a). Finally, the player plays the chosen level. Each such instance is treated as a plaver-vs-level match and the results are used to update the player skills and ratings. If the player completes the level, their list of acquired skills is updated with the new skill(s) required by that level. Additionally, the player's rating is updated based on their score on that level.

Progressions To evaluate the relative merits of skill chains and rating systems, we defined four level progressions. In all cases, previously beaten levels and the immediately preceding level played are not eligible for serving.

- SKILL_RAT: use skill chains to determine eligible levels and choose the level to serve using the rating system
- SKILL_ONLY: use skill chains to determine eligible levels but then randomly choose the level to serve rather than using ratings
- RAT_ONLY: use only ratings to pick level to serve, rather than determine eligible levels using skill chains
- RANDOM: randomly pick level from among all eligible levels, thus ignoring both skill chains and rating systems

The progressions thus respectively use both skill chains and ratings, only skill chains, only ratings and neither.

Evaluation and Discussion

We ran a Human Intelligence Task (HIT) on Amazon Mechanical Turk for each game. The HITs paid $1 but payment was upfront and playing the game was completely optional. This was motivated by past work (Sarkar and Cooper 2018) to give players flexibility in how long they played. We recruited approximately 330 players for each game. After filtering for errors in data and players who accepted payment but didn't play, we had data for 293 players for *Iowa James* and 230 for *Paradox*. In each game, each player was randomly assigned to one of the four progressions described above, leading to 76, 97, 68 and 52 players in SKILL_RAT, RAT_ONLY, SKILL_ONLY and RANDOM respectively for *Iowa James* and 52, 81, 54 and 43 respectively for *Paradox*.

For each progression, we looked at:

- *Play Time*: the time in seconds that a player in that progression played the game

Variable	SKILL_RAT	SKILL_ONLY	RAT_ONLY	RANDOM
Play Time ($p = .29$)	355	489	419	269
Final Player Rating ($p = .19$)	1406	1401	1353	1358
Max Level Rating† ($p < .001$)	1669^a	1839^b	1662^a	1517^a
Levels Completed† ($p < .001$)	3^a	2^b	3^{ab}	1^c
Levels Failed ($p = .1$)	2.5	4	3	4
Max Skillset Size ($p = .14$)	2	2	2	1

Table 1: Analysis for *Iowa James* showing median values. Variables with daggers† had significant differences in omnibus tests. Values with shared letter superscripts^{abc} were not found to be different in pairwise post-hoc comparisons.

Variable	SKILL_RAT	SKILL_ONLY	RAT_ONLY	RANDOM
Play Time ($p = .81$)	443	481	466	395
Final Player Rating ($p = .09$)	1069	1122	1075	1395
Max Level Rating† ($p < .001$)	758^a	758^a	602^b	0^b
Levels Completed† ($p < .001$)	3^{ab}	3^a	2^b	0^c
Levels Failed† ($p = .03$)	1^a	2^{ab}	4^b	2^{ab}
Max Skillset Size† ($p < .001$)	2^{ab}	3^a	2^{bc}	0^c

Table 2: Analysis for *Paradox* showing median values. Variables with daggers† had significant differences in omnibus tests. Values with shared letter superscripts^{abc} were not found to be different in pairwise post-hoc comparisons.

- *Final Player Rating*: the Glicko-2 rating that a player ended up with after finishing playing
- *Max Level Rating*: the Glicko-2 rating of the most difficult level that a player was able to complete
- *Levels Completed*: number of levels completed by the player
- *Levels Failed*: number of levels attempted but not completed by the player
- *Max Skillset Size*: highest number of skills in the skill chain of any level that the player managed to complete

For each metric, we ran an omnibus Kruskal-Wallis test across progressions. If significant, we ran pairwise post-hoc Wilcoxon Rank-Sum tests with the Holm correction comparing all pairs of progressions. For *Iowa James*, we found significant omnibus differences across progressions for *Max Level Rating* and *Levels Completed* while for *Paradox*, in addition to these metrics, *Levels Failed* and *Max Skillset Size* were also significantly different across progressions. Significant post-hoc differences were also observed for each of the above metrics. Results of all statistical analyses are shown for *Iowa James* and *Paradox* in Tables 1 and 2 respectively.

Quantity and Difficulty of Completed Levels The goal for DDA in HCGs is optimizing both the quantity and difficulty of completed levels since harder levels correspond to the harder underlying problems which would most benefit from HCG modeling. Thus high values for *Levels Completed* and *Max Level Rating* are preferred.

For *Iowa James*, the progression SKILL_ONLY does best in terms of completing more difficult levels. The number of levels completed is more nuanced, with SKILL_RAT completing more than SKILL_ONLY, but RAT_ONLY not found different from either. Though not significant, players on average failed more levels under SKILL_ONLY than both

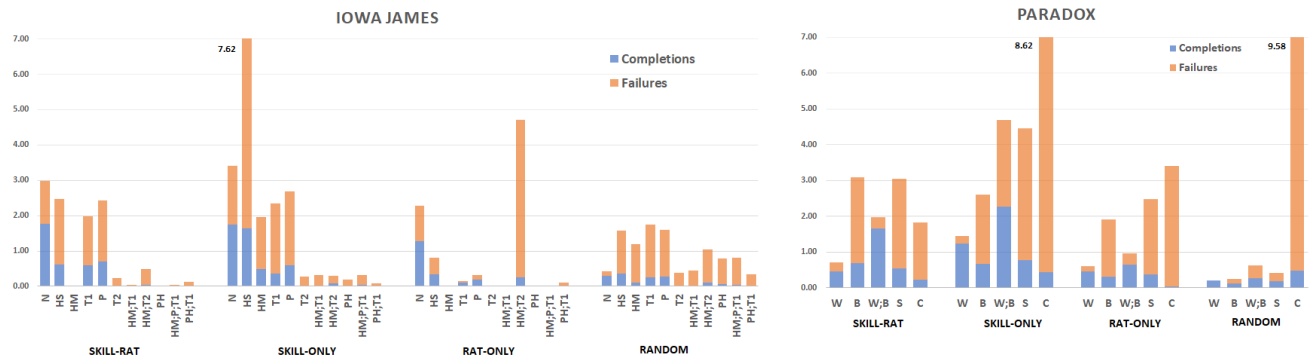


Figure 3: (left) Number of completions and failures for the 11 unique skillsets in *Iowa James*, for each progression. Number of levels for each skillset were: N - 3, HS - 12, HM - 9, T1 - 12, P - 12, T2 - 3, HM;T1 - 3, HM;T2 - 6, PH - 6, HM;P;T1 - 6, PH;T1 - 3. Letters refer to skills as defined previously. (right) Number of completions and failures for the 5 unique skillsets in *Paradox*, for each progression. Number of levels for each skillset were: W - 1, B - 1, W;B - 3, S - 2, C - 40. Letters refer to skills as defined previously.

SKILL_RAT and RAT_ONLY, with players failing the fewest levels in SKILL_RAT. This may be due to serving levels randomly from a given skill-based hierarchy rather than using ratings. It is possible for levels requiring the same skillset to vary in difficulty. Foregoing ratings to randomly select which skill-based eligible level to serve probably resulted in players playing levels too hard for them even if they had all the required skills, leading to more failed attempts at levels, but also potentially a chance of completing a level with a higher rating. Thus, incorporating ratings addresses this as shown by the results for SKILL_RAT albeit at the cost of lower values for *Max Level Rating* (similar to RAT_ONLY). We see that when using a skill chain, incorporating or excluding ratings may allow a trade-off between a higher number of level completions and a higher difficulty of levels completed, respectively.

For *Paradox*, incorporating skill chains seems more beneficial with either SKILL_RAT or SKILL_ONLY in the best group in terms of *Max Level Rating*, *Levels Completed*, *Levels Failed* and *Max Skillset Size*. We don't notice the trade-offs between SKILL_RAT and SKILL_ONLY as we did for *Iowa James* and these conditions did not exhibit significant post-hoc differences for any variable.

Overall, these results demonstrate the potential benefits of incorporating skill chains into DDA systems for HCGs with the impact of additionally using the rating system varying by game. Also, as expected RANDOM does worst for most measures across progressions for both games and is in the worst group of all statistical comparisons, validating the need for using either skill chains or rating systems, or both, for DDA.

Level Skillsets We also looked at how the progressions varied in terms of the types of levels played. We differentiate types of levels based on the set of skills needed to solve them. Thus, each unique skillset defines a level type. A skillset is defined as a set of one or more skill nodes that are not dependent on another in the skill chain. For each skillset, we looked at the number of times players failed and completed levels requiring that set, in each progression. Results are shown in Figure 3. The chart for each game consists of four clusters of histograms representing the four progressions with the skillsets for each progression in increasing order of difficulty from left to right. The number of completions and failures for each progression was normalized by the number of players in that progression. These figures confirm the previous findings and reveal some points of interest. For *Iowa James*, RAT_ONLY does better than RANDOM in serving players levels that they can complete but worse than the two skill-based progressions. The high failure values for the harder *HM;T2* skillset suggests that RAT_ONLY served levels requiring this skillset, based solely on ratings, multiple times to players that hadn't acquired some of the easier skills as evidenced by the values for these being much lower for RAT_ONLY. Without acquiring dependent skills, players failed to complete levels that required this particular skillset. Both the skill-based progressions show more desirable player behavior. As stated before, players under SKILL_ONLY played harder levels but lost a lot more than players under SKILL_RAT. Most of this also holds true for *Paradox* with the differences less stark due to the game having fewer unique skillsets.

Skill Acquisition Finally, we examined the percentage of players in each progression that acquired the different skills, shown in Tables 3 and 4. For both games, the differences in these percentages were statistically significant across progressions based on an omnibus chi-square test. For *Iowa James*, the percentage of players acquiring the two easiest skills is similar for all progressions except RANDOM. The medium difficulty skills show higher percentages for skill-based progressions though both RANDOM and RAT_ONLY do better for *hazard_moving* which may be due to few players in the SKILL_RAT progression playing levels requiring that skill. This may also explain the higher numbers for RAT_ONLY and RANDOM for the two hardest skills. The skill-based progressions served levels requiring these skills only if players had acquired all of the previous skills while the other two progressions served them regardless of this fact

Skill	SKILL_RAT	SKILL_ONLY	RAT_ONLY	RANDOM
navigating	96	91	97	58
hazard_static	57.9	55.7	52.9	46.2
hazard_moving	4	18.6	19.1	21.2
timed_one	34.2	13.4	20.6	19.2
platforming	38.2	19.6	17.7	21.2
timed_two	2.6	4.1	19.1	7.7
platforming_hazard	1.3	3.1	4.4	5.8

Table 3: Percentage of players that acquired the given skill in *Iowa James* ($\chi^2(12) = 34.5, p < .001$).

Skill	SKILL_RAT	SKILL_ONLY	RAT_ONLY	RANDOM
white	86.5	93.8	79.6	46.5
black	88.5	87.7	61.1	46.5
star	48.1	50.6	42.6	30.2
challenge	9.6	17.3	3.7	20.9

Table 4: Percentage of players that acquired the given skill in *Paradox* ($\chi^2(5) = 25.9, p = .002$).

Skill	SKILL_RAT	SKILL_ONLY	RAT_ONLY	RANDOM
navigating	1	2	1	3
hazard_static	4	4	4	3.5
hazard_moving	15	6.5	5	5
timed_one	5.5	8	5	5
platforming	5	5	6.5	4
timed_two	36.5	24.5	5	7
platforming_hazard	67	19	13	14

Table 5: Median number of matches to acquire each skill in *Iowa James*, using only players who acquired the skills.

Skill	SKILL_RAT	SKILL_ONLY	RAT_ONLY	RANDOM
white	2	1	2	2
black	1	2	3	2
star	5	5	5	2
challenge	9	7.5	6	2

Table 6: Median number of matches to acquire each skill in *Paradox*, using only players who acquired the skills.

so more players are likely to have attempted levels requiring the hardest skills. For *Paradox* however, the trends are more straightforward with both skill progressions having higher percentages than the other progressions for all skills except the hardest where again RANDOM has the highest percentage. RANDOM may thus be useful if the main goal is to get as many people as possible to play the hardest levels and other learning and DDA goals need not be optimized. Lastly, we looked at the median number of matches neede dby players to acquire different skills. These are given in Tables 5 and 6. Notably, less than around half the *Paradox* players acquired the *star* skill, which is effectively the end of its tutorial. This may indicate that most players didn't learn the skills needed to effectively complete the challenge levels, which is where the bulk of the *Paradox* levels were, and where we expect ratings would be the most useful based on previous work.

Conclusion and Future Work

We studied how a DDA model's two components—skill chains and rating systems—affect difficulty balancing and player behavior both separately and taken together, in two different HCGs. We found that skill chains could be useful for getting players to solve both a higher number of tasks as well as tasks of greater difficulty than when using only rating systems as in prior work. Moreover, in certain cases, introducing rating systems to a skill chain-only model helped better balance the difficulty of the game.

There are several avenues for future work. Currently, skill chains were defined manually but in the future, they could be derived from playtesting analysis (Horn, Cooper, and Deterding 2017), or inferred automatically. Future work could also explore automatically annotating levels with the skills needed to complete them. A limitation of our work is how skills are acquired. Once players complete a level, they are considered to have acquired all skills required by that level. However, it is possible that some skills are more instrumental to level completion and players may not have used each listed skill to finish the level. Thus, it may be helpful to treat skill acquisition more granularly and update skill mastery relative to individual skills rather than skillsets. Moreover, skill mastery in the current model is binary—either players have a skill or they do not. In the future, we could incorporate Bayesian Knowledge Tracing (Baker, Corbett, and Aleven 2008; Yudelson, Koedinger, and Gordon 2013), used in educational data mining, to facilitate more nuanced mastery tracking and inform the confidence we have in players to exercise skills rather than treat it as a binary. A final limitation is that currently, once players acquire a skill, they cannot lose mastery of it even if they never use it again or go on to do so incorrectly. For this, it could be useful to incorporate a forgetting mechanism, perhaps inspired by the curve of Ebbinghaus (2013) as used by Wang et al. (2019).

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1652537.

References

Baker, F. B., and Kim, S.-H., eds. 2004. *Item response theory: parameter estimation techniques, second edition*. New York: CRC Press, 2nd edition.

Baker, R. S.; Corbett, A. T.; and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on intelligent tutoring systems*, 406–415. Springer.

Cook, D. 2007. The chemistry of game design. *Gamasutra*. http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php.

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 1st International Joint Conference of DiGRA and FDG*.

Desmarais, M. C., and Baker, R. S. J. d. 2012. A review of recent advances in learner and skill modeling in intelligent

learning environments. *User Modeling and User-Adapted Interaction* 22(1):9–38.

Ebbinghaus, H. 2013. Memory: A contribution to experimental psychology. *Annals of Neurosciences* 20(4).

Echeverria, A.; Barrios, E.; Nussbaum, M.; Amestica, M.; and Leclerc, S. 2012. The atomic intrinsic integration approach: A structured methodology for the design of games for the conceptual understanding of physics. *Computers & Education* 59(2):806–816.

Elo, A. E. 1978. *The rating of chessplayers, past and present*. Arco.

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

Guzmán, E.; Conejo, R.; and Pérez-de-la Cruz, J.-L. 2007. Adaptive testing for hierarchical student models. *User Modeling and User-Adapted Interaction* 17(1):119–157.

Harpstead, E., and Aleven, V. 2015. Using empirical learning curve analysis to inform design in an educational game. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*, 197–207.

Horn, B.; Miller, J. A.; Smith, G.; and Cooper, S. 2018. A Monte Carlo approach to skill-based automated playtesting. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Horn, B.; Cooper, S.; and Deterding, S. 2017. Adapting cognitive task analysis to elicit the skill chain of a game. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, 277–289.

Hunicke, R. 2005. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 429–433.

Jennings-Teats, M.; Smith, G.; and Wardrip-Fruin, N. 2010. Polymorph: dynamic difficulty adjustment through level generation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, 11:1–11:4.

Khenissi, M. A.; Essalmi, F.; Jemni, M.; and Kinshuk. 2015. Learner modeling using educational games: a review of the literature. *Smart Learning Environments* 2(1):6.

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.; Popovic, Z.; Khatib, F.; Cooper, S.; and Baker, D. 2019. De novo protein design by citizen scientists. *Nature* 570(7761):390–394.

Krause, M.; Takhtamysheva, A.; Wittstock, M.; and Malaka, R. 2010. Frontiers of a paradigm - exploring human computation with a digital game. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*.

Millán, E., and Pérez-de-la Cruz, J. L. 2002. A Bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction* 12(2):281–330.

Mu, T.; Wang, S.; Andersen, E.; and Brunskill, E. 2018. Combining adaptivity with progression ordering for intelligent tutoring systems. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*.

Nintendo. 1985. *Super Mario Bros.* Game [NES].

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. 2019a. Transforming game difficulty curves using function composition. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*.

Sarkar, A., and Cooper, S. 2019b. Using a disjoint skill model for game and task difficulty in human computation games. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*.

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*.

Siu, K.; Guzdial, M.; and Riedl, M. 2017. Evaluating single player and multiplayer in human computation games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*.

Sturn, T.; Wimmer, M.; Purgathofer, P.; and Fritz, S. 2013. Landspotting – games for improving global land cover. In *Proceedings of the 8th International Conference on the Foundations of Digital Games*.

Valve Corporation. 2008. *Left 4 Dead*. Game.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computer Systems*.

Wang, S.; Cohen, B.; Yi, S.; Park, J. Y.; Teo, N.; and Andersen, E. 2019. Goal-based progression synthesis in a Korean learning game. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*.

Wang, S.; He, F.; and Andersen, E. 2017. A unified framework for knowledge assessment and progression analysis and design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*.

Yudelson, M. V.; Koedinger, K. R.; and Gordon, G. J. 2013. Individualized Bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*.

Zook, A., and Riedl, M. 2012. A temporal data-driven player model for dynamic difficulty adjustment. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.