Proceedings, The Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment

Evaluating Competitive Game Balance with Restricted Play

Alexander Jaffe, Alex Miller, Erik Andersen
Yun-En Liu, Anna Karlin, Zoran Popović

Department of Computer Science & Engineering
University of Washington
{ajaffe,amiller,eland,yunliu,karlin,zoran}@cs.washington.edu

Abstract

Game balancing is the fine-tuning phase in which a functioning game is adjusted to be deep, fair, and interesting. Balancing is difficult and time-consuming, as designers must repeatedly tweak parameters, and run lengthy playtests to evaluate the effects of these changes. If designers could receive immediate feedback on their designs, they could explore a vast space of variations, and select only the most promising games for playtesting. Such automated design feedback has been difficult to achieve, as there is no mathematical formulation of game balance that unifies many of its forms. We argue for a formulation in which carefully restricted agents are played against standard agents. We develop this *restricted-play balance framework*, and evaluate its utility by building a tool capable of calculating measures of balance for a large family of games. By applying this tool to an educational card game, we demonstrate how the framework and tool allow designers to rapidly evaluate and iterate on the balance of their games.

1 Introduction

Balancing is one of the most difficult and time-consuming phases of the game design process. In balancing, a playable game undergoes a tuning cycle of subtle adjustments and playtesting, in an effort to improve depth, pacing, fairness, randomness, and variety. Balance questions are diverse, from "What is the first player's advantage?", to "How much long-term strategy is there?", to "Does every action serve a useful role?" Such questions are particularly relevant to competitive games, which must remain interesting to a community, often for several years.

The difficulty of balancing arises from the emergent complexity of systems. Small tweaks can have unexpected consequences for a game's viable strategies, and are difficult to reason about. As such, evaluating balance usually requires extensive playtesting. While developing *Halo 3*, Bungie discovered that the sniper rifle was overpowered, subsuming several interesting strategies (Griesemer 2010). To resolve this imbalance while adhering to their qualitative design goals, they experimented with many adjustments, including reload time, clip size, zoom time, max ammo, and time between shots. Each tweak involves substantial playtesting and designer intuition, while producing inherently fuzzy answers.

A natural question is whether some portion of competitive game balance evaluation could be automated, through AI simulation by a backend reasoning tool (Nelson and Mateas 2009) rather than playtesting. One major roadblock to this goal is that the standard approach to evaluating balance relies on observation of typical human play, yet AI play is not currently predictive of human play (Hingston 2009). The diversity of viable strategies in complex games makes it unlikely for even a strong AI agent to behave like a human. A more basic roadblock is that many balance questions are not even well-defined.

One balance question that *is* relatively well-defined is the fairness of starting conditions. For example, the fairness of *Chess* is characterized by the win rates of the white and black players. In this paper, we propose an analogous quantitative formulation of other kinds of balance. We reduce them to questions about the win rates of asymmetric agents. This enables us to exploit similarities between human and AI *skill level* rather than behavior. Such similarities are now becoming a reality, as recently developed AI techniques, such as Monte-Carlo Tree Search, can play an unprecedented variety of games competitively with humans. (See §5.)

For example, suppose we wish to automatically gauge whether a game is focused more on long-term strategy, like *Risk*, or on short-term tactics, like a first-person shooter. To do so, we measure the win rate of a *greedy* agent, who optimizes only for a simple 'score' function of the subsequent state, versus a standard, proficient opponent who exploits the restricted agent's greediness. In *Chess*, the score might be number of pieces; in a fighting game, it might be health.

As another example, a designer could ask whether one action is more powerful than the others. To do so, we may measure the win rate of an agent who is restricted from ever (or often) taking that action, against an agent who exploits this restriction.

We propose that such formulations facilitate an algorithmic approach to answering competitive game balance questions, since win rates can be estimated with repeated games between reasonably strong AI agents. It allows us to build an 'early warning system', that illuminates certain flagrant imbalances upfront, saving playtesting time for the most promising iterations. An imbalance can match a designer's

goals, but she will always hope to be aware of it.

Consider a chess-like game, where one design goal is that all the pieces must work in unison. A designer might consider a rebalance in which the movement radius of one of the pieces p is extended. This has the potential to impact other aspects of the game indirectly, for example rendering useless some other piece q. This can be hard to predict using reason alone. Yet by observing that in the rebalance, restricting a player from moving q no longer diminishes her win rate, the designer is immediately directed to the flaw. She could then use this information to refine her intuition, and go back to the drawing board without playtesting this iteration at all. If, on the other hand, the reasons for this result remain unclear, she always has the option of playtesting.

To explore the promise of this approach, we have built a prototype balancing system for a large family of two-player competitive games. This system provides rapid design feedback, and allows a designer to iteratively modify game parameters and quantitatively observe the effects. We investigate the functionality and relevance of this system with a case study in which we balance an educational card game. In an effort to better understand the value of the balance formulations themselves, our prototype system works with optimal agents. In this way, our results are not affected by the secondary issue of how well particular AI agents play.

This paper is only a preliminary step toward a broader goal, and its limitations and future work are discussed in Section 5.

2 Balance Measures

In this paper, we consider zero-sum two-player games. A game is given by a set of states (some of which are win or tie states), a set of actions available to each player at each state, and a transition function mapping each state and pair of actions to a child state, or distribution over child states.

Let $\mathcal{R}$ be some behavior restriction - for instance, never playing a certain action, or avoiding certain outcome states. For some player or agent A, we let $A_{\mathcal{R}}$ be A modified with the restriction $\mathcal{R}$. If we consider some game element that $\mathcal{R}$ avoids, then we measure the impact of that game element by the probability that $A_{\mathcal{R}}$ beats A. [1] (Technically, we work with the *value* of the game (von Neumann 1928), by treating a tie as a 50% chance of a win.) We assume that A has full knowledge of its opponent's restriction, and is thus able to exploit it.

We now describe several initial categories of balance feature, and the restricted behaviors $\mathcal{R}$ to measure their impact.

How important is playing unpredictably? In real-time games such as fighting games, optimal play often requires players to randomize their strategies. However, the cost of playing predictably can vary across games. We measure the importance of unpredictability with a restricted player who must play low-entropy mixed strategies. In practice, this can be approximated by a player who can play any distribution, as long it has small support. $\mathcal{R} : |\mathbf{Support}| \leq \mathbf{k}$.

To what extent must players react to the current state of the game? Real-time strategy games often allow for fixed 'build orders', which are long-term strategies suggesting some actions to take at each timestep, roughly independently of the other player's actions (Brandy 2010). If a build-order is rigid yet effective, it may result in a strategically uninteresting game. We capture the importance of adaptation with an *oblivious* player: one who for the first k rounds knows only her own actions. She cannot query the current state of the game or her opponent's actions, and instead must simulate her opponent. $\mathcal{R} : \mathbf{Oblivious\text{-}until\text{-}round\text{-}k}$.

How powerful is a given action or combination of actions? In strategy games such as *Chess*, each piece is meant to act in unison: each should be essential to the game. In contrast, first-person shooters typically allow players to play according to their preference of weapon and play-style: no one weapon should be essential. We capture the importance of an action with a restricted player who is forbidden from using that action ever (or more than k times). Also informative is a restricted player who *must* use that action the first k times it is available. For action $\mathbf{a}$, $\mathcal{R} : |\mathbf{Plays(a)}| \leq \mathbf{k}$, or $\mathcal{R} : |\mathbf{Plays(a)}| \geq \mathbf{k}$.

How much long-term strategy is necessary? Actions in strategy games often have emergent consequences which are not observed until many turns later. In contrast, action and racing games, among others, tend to rely more on short-term tactical thinking. We capture the impact of long-term strategy with a player who is restricted to explore states at most k steps into the future. At one extreme, a player does not strategize at all, playing completely randomly. Depth-1 strategy produces a *greedy* player, who optimizes for some score function of the subsequent state, like health or piece count. At the other extreme, a player may search all the way to the leaves. The win rates of these varying search depths (against a strong, unrestricted opponent) gives a broad picture of the role of long-term strategy in the game. $\mathcal{R} : \mathbf{Depth} \leq \mathbf{k}$.

Is the outcome known long before the game's end? Some games (such as *Risk*) are infamous for a 'lame duck' phase, in which the likely winner is known, but players must continue for some time to finish the game. This can decrease tension, or force players to concede early. On the other hand, other games (such as *Mario Kart* with 'rubber banding' opponents) leave the outcome uncertain until the end, making the majority of the game feel meaningless. To capture the extent to which outcomes are apparent early in the game, we study players who assign an additive value bonus ϵ to 'low' states, those at most k moves from a leaf. If this player cannot win frequently, it indicates that the game will tend to drag on, even once the winner is somewhat certain. $\mathcal{R} : \epsilon\text{-}\mathbf{bonus\text{-}to\text{-}height\text{-}k}$.

What is the effect of avoiding certain end states? Games that include multiple win conditions (such as *Civilization*) often do so to create an interesting strategic decision: players must hedge their goals, to prevent their opponent from acting to stop any one. Other games such as *Magic: The Gathering* feature multiple win conditions sim-

[1] The randomness in the outcome of a game can come from a number of sources. In games of simultaneous action, such as *Rock-Paper-Scissors*, optimal play requires randomness. However, even for non-simultaneous games, where randomness is not necessary, many heuristic agents use randomized strategies (Browne et al. 2012). Finally, a game may itself have random transitions.

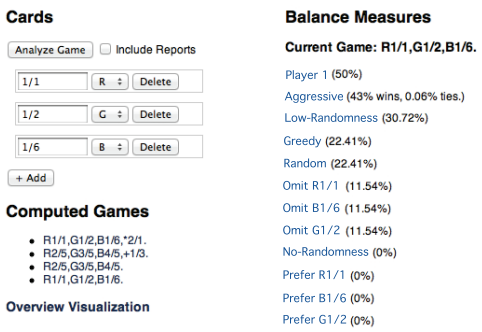

Figure 1: A basic analysis using our tool. The results show the performance of varieties of restricted play against a restriction-exploiting player. Players who never play Red 1/1, for example, win 11.54% of the time.

ply to let players choose their play style. To capture the importance of hedging goals, we study a restricted player with a modified payoff function, who treats certain win states as losses. (A related measure is for aggressive or passive players, who treat ties as losses or wins.) For a set of leaf states $\mathbf{S}$, $\mathcal{R} : \mathbf{Avoids\text{-}S}$.

Are the starting conditions of the game fair? This is the question most traditionally thought of as 'balance'. It is often essential that asymmetric starting conditions be somewhat fair, as players are 'locked into' them (Sirlin 2009). For example, one player must play black in *Chess*, and if she were at a tremendous disadvantage, the game could be frustrating. That said, even unfairness can be intentional. The *Street Fighter* series features an intentionally weak character, who players sometimes play as a challenge, a handicap, or a joke. To measure the fairness of starting conditions, we use an unrestricted player who always chooses a given condition. For a starting condition s $\mathcal{R} : \mathbf{Chooses\text{-}s}$.

3 Exploratory Study

We have built a prototype balance evaluation tool, to serve as an application of restricted play, and a testing ground for its validity. The tool measures the behavior of optimal agents, (subject to restrictions) playing two-player, perfect-information games in extensive form, with simultaneous moves and randomized transitions. Such games have optimal strategies that can be computed exactly in polynomial time (von Neumann 1928). It is straightforward to adapt such solvers to most restricted behaviors of §2.

A System to Support Rapid Iteration We describe an experimental tool for supporting rapid design iteration. This tool can be provided with code embodying a parameterized game tree. It then facilitates the following interaction cycle.

A designer inputs a set of game parameters into a web-based interface, (Figure 1, left side) and is presented with a set of balance measures for the specified game, derived by computing the value of several restricted behaviors. These measures are ranked by their intensity, so that the restricted behaviors that are most effective or ineffective are highlighted. (Figure 1, right side). The designer forms a hypothesis about how modifying the parameters could resolve the

Figure 2: *Monsters Divided*. $G1/1$ and $B5/6$ are played, with $+3/2$ and $+1/3$ power cards applied to them, respectively. G's rule is in play: closest to 1/2 wins. $5/6+1/3$ beats $1/1 + 3/2$, so P2 wins a B trophy.

balance features she finds most problematic, and repeats.

Of the balance features in §2, our prototype tool implements all but two of the proposed balance measures.

- 'Omit $\mathbf{a}$': $|\mathbf{Plays(a)}| \leq \mathbf{0}$.
- 'Prefer $\mathbf{a}$': $|\mathbf{Plays(a)}| \geq \infty$.
- 'Random': $\mathbf{Depth} \leq \mathbf{0}$.
- 'Greedy': $\mathbf{Depth} \leq \mathbf{1}$.
- 'Agressive': $\mathbf{Avoids\text{-}ties}$.
- 'Low-randomness': $|\mathbf{Support}| \leq \mathbf{2}$.
- 'No-randomness': $|\mathbf{Support}| \leq \mathbf{1}$.
- 'Player 1': $\mathbf{Chooses\text{-}first\text{-}player}$.

Monsters Divided To evaluate the relevance of our tool, we used it to help balance *Monsters Divided* - a card game developed by our lab to help teach fractions to elementary school students. (Figure 2.) The game consists of *monster cards* and *power cards*. Each monster card has a fraction and a color: red (R), green (G), or blue (B). Each power card represents a function which modifies a fraction through either addition or multiplication. For example, a power card may be labeled $+1/3$ or $*2/3$. Each player is given three monster cards and two power cards, visible to both players.

In each round, players simultaneously play a single monster card, trying for the 'best' fraction. The color of the played monster cards determine the 'color of the round'. Blue overrides red, red overrides green, and green overrides blue. *In a red round, the highest fraction wins. In a blue round, the lowest fraction wins. In a green round, the fraction closest to* $\frac{1}{2}$ *wins.* After playing monster cards, players can also choose to play a single power card, or pass. Power cards can be applied to either player's fraction. At the end of the round, the winning player gets a trophy of her monster's color. If there is a tie, both players get their monsters' respective trophies. The played power cards are discarded, and the monster cards are returned to the players. To win the game, a player needs either one trophy of each color or three trophies of one color. The game can end in a tie.

Balancing the Cards The rules above leave unspecified

	Power Cards	Noteworthy Balance Measures	Interpretation
A	$+1/2$	Omit Green: 13.54%, Omit Blue: 7.44%, Omit Red: 7.83%.	Green is too weak. (Green cannot beat Red, even by playing +1/2.)
B	$+2/3$	Omit Blue: 6.76%, Omit Red: 11.62%, Omit Green: 12.38%.	Blue is too strong. (Red cannot beat Blue, even by playing +2/3.)
C	$+2/1$	Omit Blue: 10.77%, Omit Red: 10.83%, Omit Green: 10.78%.	All monsters are about as strong. The slight differences point to interesting variety in how they are played.
D	$*1/2$	Random: 8.04%.	Random play is unreasonably effective.
E	$*2/3$	Omit $Self * 2/3$: 47.41%, Random: 11.99%.	Power card on self unhelpful; random play too good.
F	$*4/1$	Omit Blue: 6.76%, Omit Green: 12.38%, Omit Red: 11.62%, Random: 3.26%.	Blue is too strong, but now random performs sufficiently poorly.
G	$*2/3$, $*4/1$	Greedy: 3.41%, Prefer $Self * 4/1$: 3.83%.	Too harsh to simple players, and one direction of $* 4/1$ is too self-destructive.
H	$*2/3$, $+2/3$	Prefer $Self * 2/3$: 5.41%, Omit Blue: 4.31%.	Multiplier power card is now less self-destructive, but Blue is now too strong.
I	$*2/3$, $+1/2$	Prefer $Self * 2/3$: 23.73%, Greedy: 7.95%.	All values appear reasonable, and there is nice variety among the actions. Ready for playtesting.

Table 1: Balancing iterations of the set of power cards in *Monsters Divided*. We give noteworthy balance measures for each iteration, and the interpretation that led to the next iteration. For example, in iteration E, playing the power card $*2/3$ on one's self is mostly useless, as the inability to do so barely hurts chances of winning. Also in E, playing randomly performs better than intended. We address these problems in F, with a larger power card that may have a stronger effect.

the cards used by each player. Hence our next design goal was to choose a small, balanced set of starting cards.

We knew that a simple set of monster cards (*Red 1/1*, *Green 1/2*, and *Blue 1/6*) created a reasonably interesting game of hedging against win states and prediction. To add complexity, we wished to add a pair of power cards to the game. Rather than generate a single plan and immediately playtest it, we used our tool to generate interactive feedback on many designs, build hypotheses to explain the feedback, and invent new variations to address the feedback. In this way we threw out many inarguably bad card sets, until a promising set could be promoted to playtesting.

To select a strong pair of power cards, we first investigated the properties of single power cards in isolation. We then moved on to select a pair of these cards. Note that at all stages, we relied on some intuition for the values of a balance measure that should be considered a success. This comes partly from common sense, but also from practice working with the tool.

An abridged history of the design session is found in Table 1. The variations shown constitute half of the evaluated variations - only the most informative. By applying the tool, we were able to quickly discover design flaws such as:

An overpowered or underpowered monster card. We intended all three monster cards to be roughly comparable in usefulness. This balance was frequently broken by power cards, and the tool immediately informed us, as in variations A, B, and F. For example, in variation A we see that restricting the player from playing Green has a smaller impact on the win-rate than restricting Red or Blue. This suggests Green as under-powered. We corrected such variations in two ways. Sometimes we reasoned about the ways a power card could help one monster more than others, and tried to weaken that effect. Other times, we simply searched the surrounding space of cards, evaluating the tool on several similar variations. In this way we formed intuition for the effects of differing power cards.

Leniency or strictness toward simple players. We wished to create a strategic game that was nonetheless accessible to players with a basic understanding of the rules. Hence we were wary of variations in which greedy or random behaviors performed extremely well or poorly. Variations D and E suffered from the former problem, G from the latter problem. In variation G, a greedy player wins only 3.4% of games, which probably indicates too harsh a punishment for mediocre play. To address these problems, we focused on tempering the power card that was most effective or harmful when 'preferred' (played as soon as possible).

A self-destructive or irrelevant power card. In some variations, a power card was unhelpful, or self-destructive if used at the wrong time. These diagnoses came from observing that an 'omit' behavior performs nearly perfectly, while a 'prefer' behavior performs terribly. This occurred in variation G, where preferring the power card '$Self * 4/1$' often lead to losing the game, and in variation H. We decided that a 'decoy' strategy - which players must learn to avoid - was harmful to the accessibility of our game.

After twenty or so iterations, we arrived at a design which passed most of the quantitative tests we were concerned with. Not all values were perfect, but they were satisfactory enough to feel confident going into playtesting. The design problems we identified would have been definitively incompatible with our goals for the game. In using this tool, we saved the time and effort of up to 20 phases of manual evaluation, many of which would have required playtesting. Moreover, by tracking the graph of balance features across iterations (Figure 3) we gained a novel form of intuition for the dependencies of the game on parameter choices.

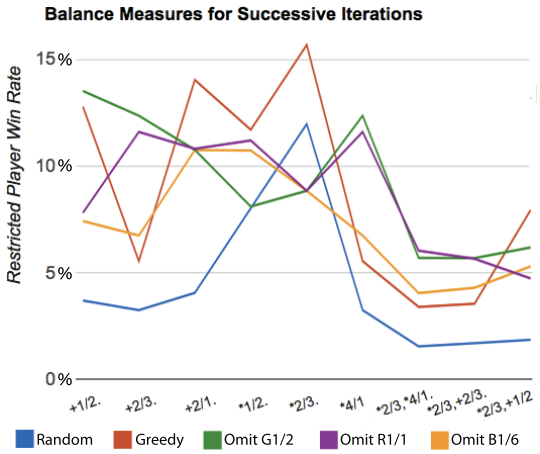

Figure 3: Balance parameters across variations. The gaps between *Greedy* and *Random* convey the relative roles of long-term strategy and short-term tactics. The designer can easily discover trends such as the power gap between G1/2 and R1/1 due to a 1/2 power card.

4 Related Work

Playtesting Improvements Developers have recently made strong improvements to the balancing process while maintaining the same basic cycle, using data analytics. Valve, for instance, collects many types of data, such as player statistics, survey information, and biometrics (Ambinder 2009). Heatmaps have been used by developers to analyze games such as *Halo 3* (Romero 2008), and even in games without virtual environments, by using Playtracer (Andersen et al. 2010). All of this supplements traditional interviews and observations, which continue to be an integral part of the balancing cycle.

One alternative approach to balancing is mathematical modeling. Here designers express the power of game elements as a quantitative formula of their definitional characteristics. For instance, the power of a card could be expressed as its fraction minus some constant times its cost. Such approaches offer a helpful additional perspective, but are of questionable validity, due to the frequent arbitrariness of the cost function.

Automated Balance Evaluation *Ludi* (Browne 2008) is a system for describing and automatically playing general games. Browne computes several statistics of this play, measures their correlation to human rankings, and uses them to generate proposed games. Among the many design measures that Browne proposes, there are four that measure the win rate of a restricted player. These measure players who search at a limited depth, play randomly, or optimize only over one player's pieces. However, Browne does not address the unique advantages of these measures, over those based on observation of AI behavior.

(Marks and Hom 2007) aim to actually design 'balanced' board games, where 'balanced' means that the game's starting conditions are fair. They measure this property by playing AI agents against themselves and checking that one side does not win a disproprortionately large amount of the time, a special case of our framework.

Several projects described below illuminate game designs using simulated players with custom properties. However, all focus primarily on producing example play traces, in favor of quantitative formulation of balance features.

(Nelson 2011) conceptually explores a variety of automatically-generated game metrics, including (briefly) studying the behavior of agents from interesting simple families. This idea is in essence restricted-play. Machinations (Dormans 2011) is a diagramming tool for describing in-game resource dependencies. It is intended to help designers reason about resources through examination, and through simulation. Users write scripts to simulate interesting player strategies, and observe their play in the model. (Chan et al. 2004; Denzinger et al. 2005) have studied similar questions in a series of papers, but are concerned only with the special case of discovering short but powerful action sequences. For instance, they discover that in *FIFA* one can consistently score with corner kicks. (Smith, Nelson, and Mateas 2009; 2010) built *BIPED*, a game prototyping system, on top of their logical engine, *Ludocore*. *BIPED* generates play traces obeying designer-specified constraints. These traces can be examined to gain early qualitative insight into a design.

5 Limitations and Future Work

This work has been the first exploratory step in a much larger project: a system to guide, inform, and accelerate the balancing process of a variety of games. Here we overview limitations of our approach, and discuss ways to circumvent them.

Limitations of Restricted-Play While restricted play is quite expressive, many kinds of balance simply cannot be captured through this formulation. Any property that is deeply rooted in human psychology, or the look and feel of a game, is untenable. Conversely, although the current framework does not model the difficulty of successfully executing actions, (as when entering a special move in a fighting game) it may be possible to do so by measuring the frequency of successful execution.

On a practical level, it can be time-consuming to code a game's transition function, though libraries or logic programming can make this easier. For games involving complex physics or simulation, coding a separate transition function is likely to be prohibitive. In this case we must rely on telemetry from within the game code, which fortunately is now built into many modern games.

Limitations of The Prototype The current tool runs only on perfect information discrete games. These include many board games, strategy games, puzzle games, and other twitch-free games. We believe the tool could be generalized to games of imperfect information using the techniques of (Koller, Megiddo, and von Stengel 1996), or to continuous state spaces that are sufficiently smooth to admit discretization (Pfeifer, Brewer, and Dawe 2012). For real-time games, AI agents typically require a hand-coded set of custom rules, and adjusting them for each balancing iteration is likely infeasible. Fortunately, we need not run a game in real-time to analyze its balance, and hence can utilize slower, more advanced AI techniques.

Utilizing Heuristic Agents Our prototype uses optimal agents; it will be worthwhile to examine the usefulness of restricted play for strong but imperfect agents. This may be more realistic, and will be necessary for more complex games. We propose to exploit recent advances in heuristic AI - in particular, we are experimenting with Monte-Carlo Tree Search (MCTS) (Kearns, Mansour, and Ng 2002).

MCTS is the first technique to compete at an expert level in *Go*, and has been been used for a large variety of games (Browne et al. 2012). MCTS has several advantages over traditional tree search methods. MCTS in its purest form does not need hand-coded heuristics, and hence could be used to evaluate new iterations without designer intervention. As a sampling-based bandit method, MCTS is robust, and well-suited to adapt to player restrictions, such as removing tree edges on the fly or modifying state payoffs. Moreover, since MCTS is in essence a recursive simulation, it is not hard to make an agent model and exploit an opponent's restriction.

Note that obtaining balance measures from heuristic play may require an estimate of agent strength. This can be obtained by playing them against known humans or agents. Although AI agents are unlikely to reach top human skill levels, moderate play is informative for an initial understanding of balance. What's more, there is anecdotal evidence that game balance can be somewhat robust to skill level. Equally matched *Chess* games give white between a 54% and 56% chance of winning, depending on skill (Sonas 2012). We only wish to detect extreme violations of a designer's balance goals, so small variations such as these are perfectly acceptable. In fact, the typical handicap for the first player in *Go* is identical at every skill level (Benson 2012).

6 Conclusion

Games are design artifacts of tremendous complexity. With rare exception, they have been treated as impenetrable subjects of intuition. We have made a case in this paper for how we believe this can change. By capturing select design features in terms of the performance of agents, we provide a rigorous language for tracking, discussing and automatically measuring these features. We have explored the potential usefulness of this formulation with a new kind of game balancing tool, one which allowed us to make quick progress on a real game. We are enthusiastic about restricted play's ability to subtly reshape the design process, be it through early warning systems like the one prototyped here, or entirely new design tools.

References

Ambinder, M. 2009. Valve's approach to playtesting: The application of empiricism. Game Developer's Conference.

Andersen, E.; Liu, Y.-E.; Apter, E.; Boucher-Genesse, F.; and Popović, Z. 2010. Gameplay analysis through state projection. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*.

Benson, T. 2012. A change in komi. http://www.usgo.org/org/komi.html.

Brandy, L. 2010. Using genetic algorithms to find starcraft 2 build orders. http://lbrandy.com/blog/2010/11/using-genetic-algorithms-to-find-starcraft-2-build-orders/.

Browne, C.; Powley, E.; Whitehouse, D.; Lucas, S.; Cowling, P.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; and Colton, S. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*.

Browne, C. B. 2008. *Automatic generation and evaluation of recombination games*. Ph.D. Dissertation, Queensland University of Technology.

Chan, B.; Denzinger, J.; Gates, D.; Loose, K.; and Buchanan, J. 2004. Evolutionary behavior testing of commercial computer games. In *Evolutionary Computation, 2004. CEC2004.*

Denzinger, J.; Loose, K.; Gates, D.; and Buchanan, J. 2005. Dealing with parameterized actions in behavior testing of commercial computer games. In *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG)*, 37–43.

Dormans, J. 2011. Simulating mechanics to study emergence in games. In *Workshop on Artificial Intelligence in the Game Design Process (IDP11) at the 7th Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE)*.

Griesemer, J. 2010. Design in Detail: Changing the Time Between Shots for the Sniper Rifle from 0.5 to 0.7 Seconds for Halo 3. Game Developer's Conference.

Hingston, P. 2009. A turing test for computer game bots. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm? arnumber=5247069.

Kearns, M.; Mansour, Y.; and Ng, A. Y. 2002. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine Learning* 49:193–208.

Koller, D.; Megiddo, N.; and von Stengel, B. 1996. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior* 14(2).

Marks, J., and Hom, V. 2007. Automatic design of balanced board games. In Schaeffer, J., and Mateas, M., eds., *AIIDE*, 25–30. The AAAI Press.

Nelson, M. J., and Mateas, M. 2009. A requirements analysis for videogame design support tools. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, FDG '09, 137–144. New York, NY, USA: ACM.

Nelson, M. 2011. Game metrics without players: Strategies for understanding game artifacts. In *Workshop on Artificial Intelligence in the Game Design Process (IDP11) at the 7th Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE)*.

Pfeifer, B.; Brewer, D.; and Dawe, M. 2012. AI Postmortems: Kingdoms of Amalur: Reckoning, Darkness II and Skulls of the Shogun. Game Developers Conference.

Romero, R. 2008. Successful instrumentation: Tracking attitudes and behviors to improve games. Game Developer's Conference.

Sirlin, D. 2009. Balancing multiplayer competitive games. Game Developer's Conference.

Smith, A.; Nelson, M.; and Mateas, M. 2009. Computational support for playtesting game sketches. In *Proceedings of the Fifth Artificial Intelligence and Interactive Digital Entertainment Conference*, AIIDE '09.

Smith, A. M.; Nelson, M. J.; and Mateas, M. 2010. Ludocore: A logical game engine for modeling videogames. In *IEEE Conference on Computational Intelligence and Games (CIG)*.

Sonas, J. 2012. The sonas rating formula - better than elo? http://www.chessbase.com/newsdetail.asp?newsid=562.

von Neumann, J. 1928. Zur theorie der gesellschaftsspiele. *Mathematische Annalen* 100(1):295–320.