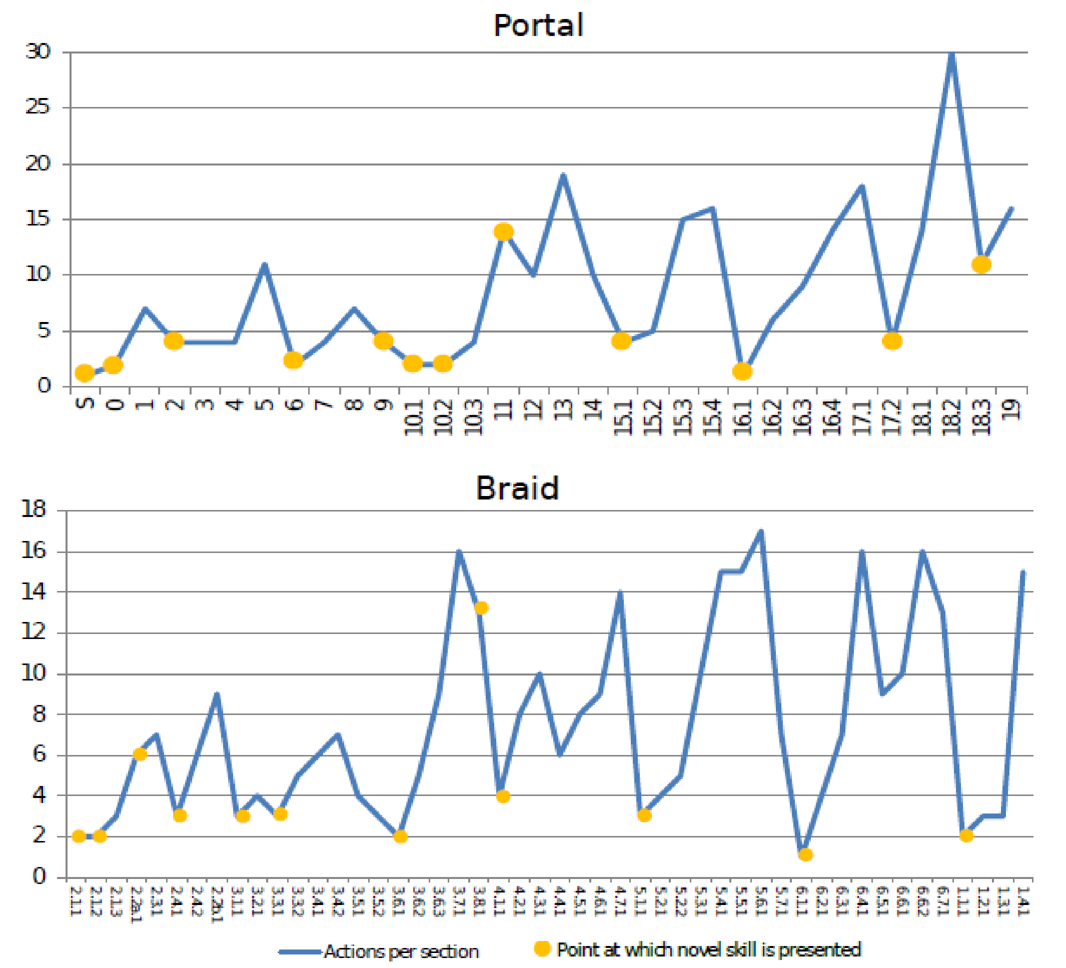
Transforming Game Difficulty Curves using Function Composition

Anurag Sarkar and **Seth Cooper**

Northeastern University

Difficulty Curve

- Defines how a game's difficulty changes over the course of gameplay

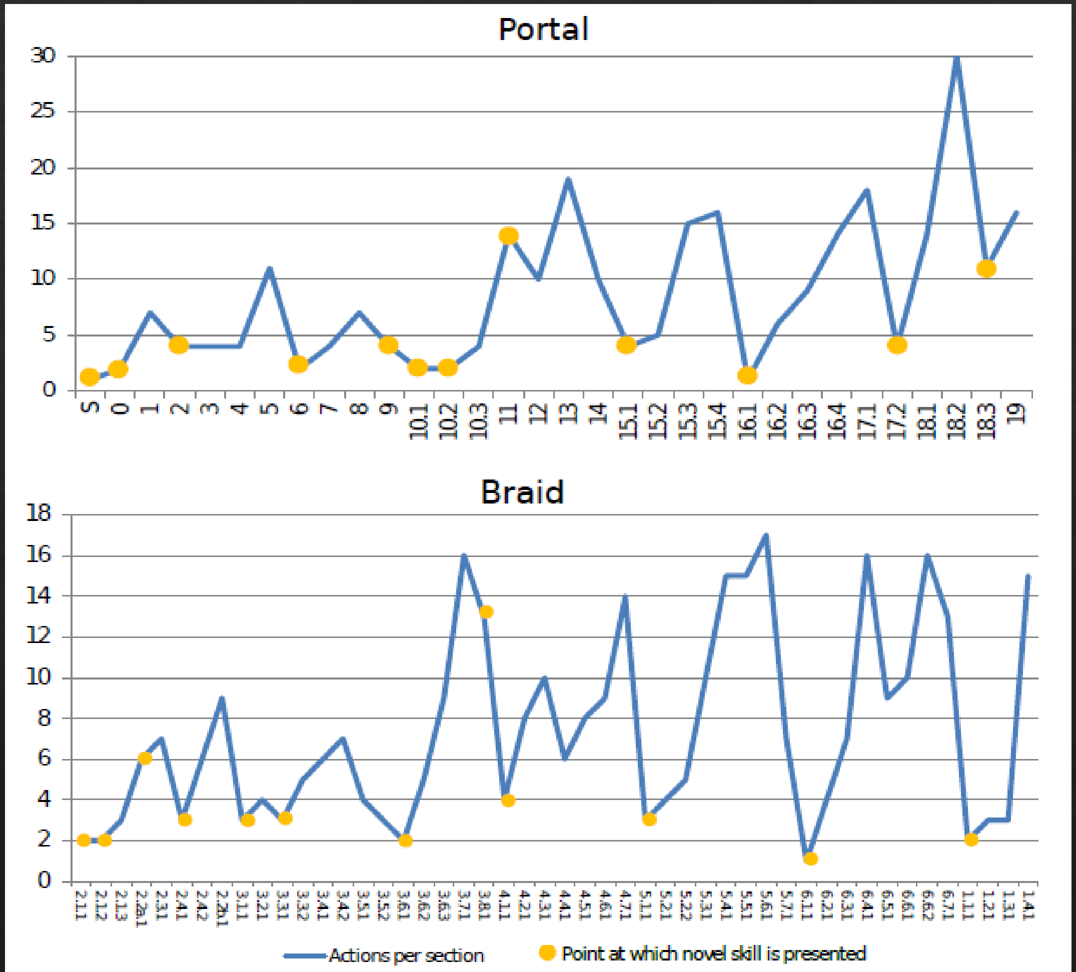


Linehan et al., 2014

Difficulty Curve

- Defines how a game’s difficulty changes over the course of gameplay

- Curves can be viewed as functions mapping from progression to difficulty



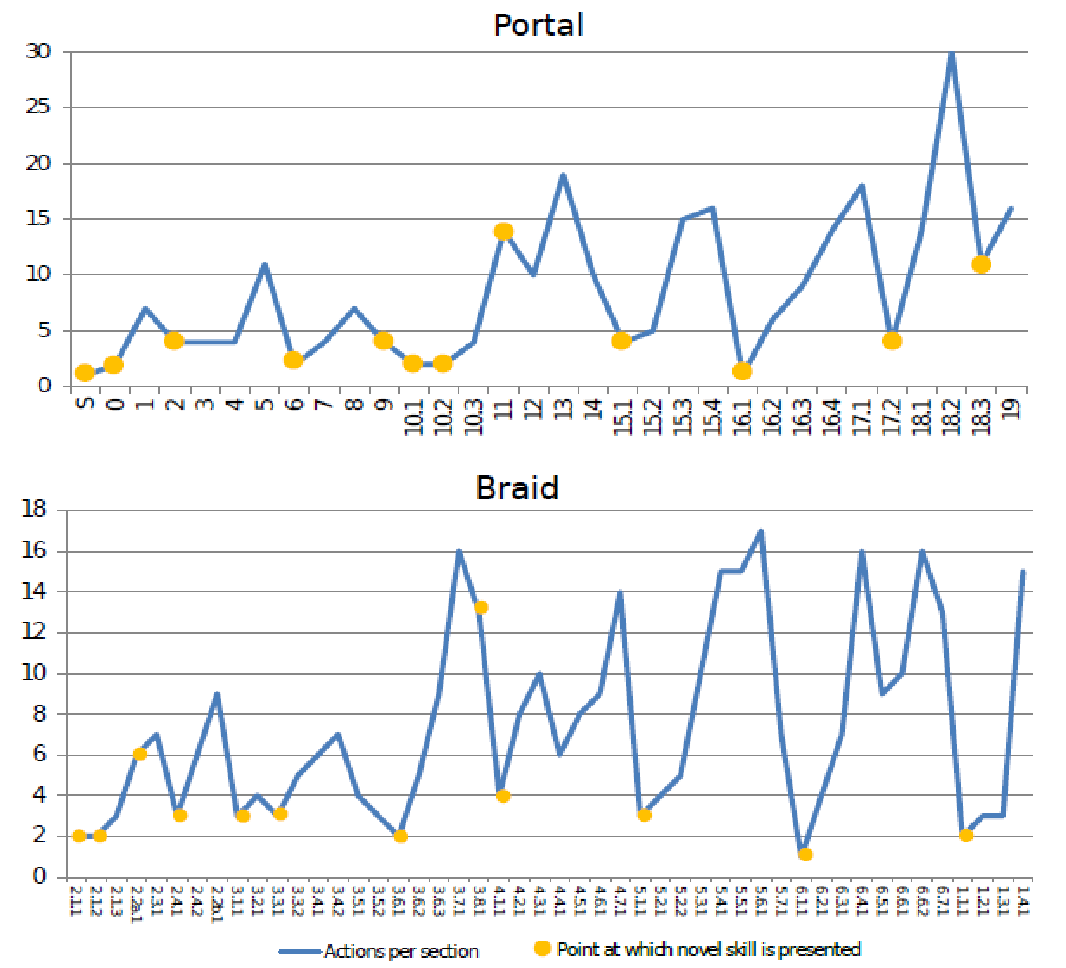
Linehan et al., 2014

Difficulty Curve

- Defines how a game's difficulty changes over the course of gameplay

- Curves can be viewed as functions mapping from progression to difficulty

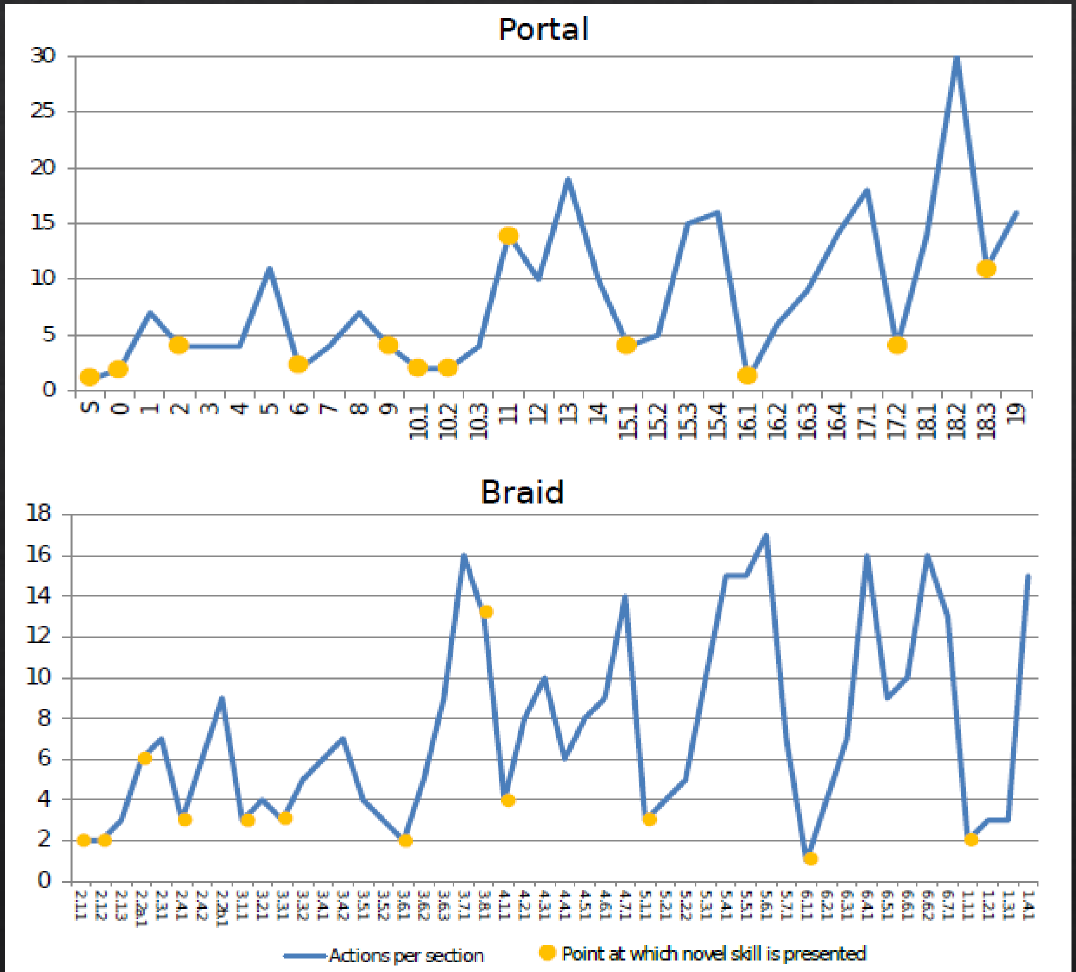
- In-game difficulty affects player engagement and so should be compatible with player skill



Linehan et al., 2014

Difficulty Curve

- Defines how a game's difficulty changes over the course of gameplay
- Curves can be viewed as functions mapping from progression to difficulty
- In-game difficulty affects player engagement and so should be compatible with player skill
- Traditional methods of defining curves involve manual refinement through iterative playtesting

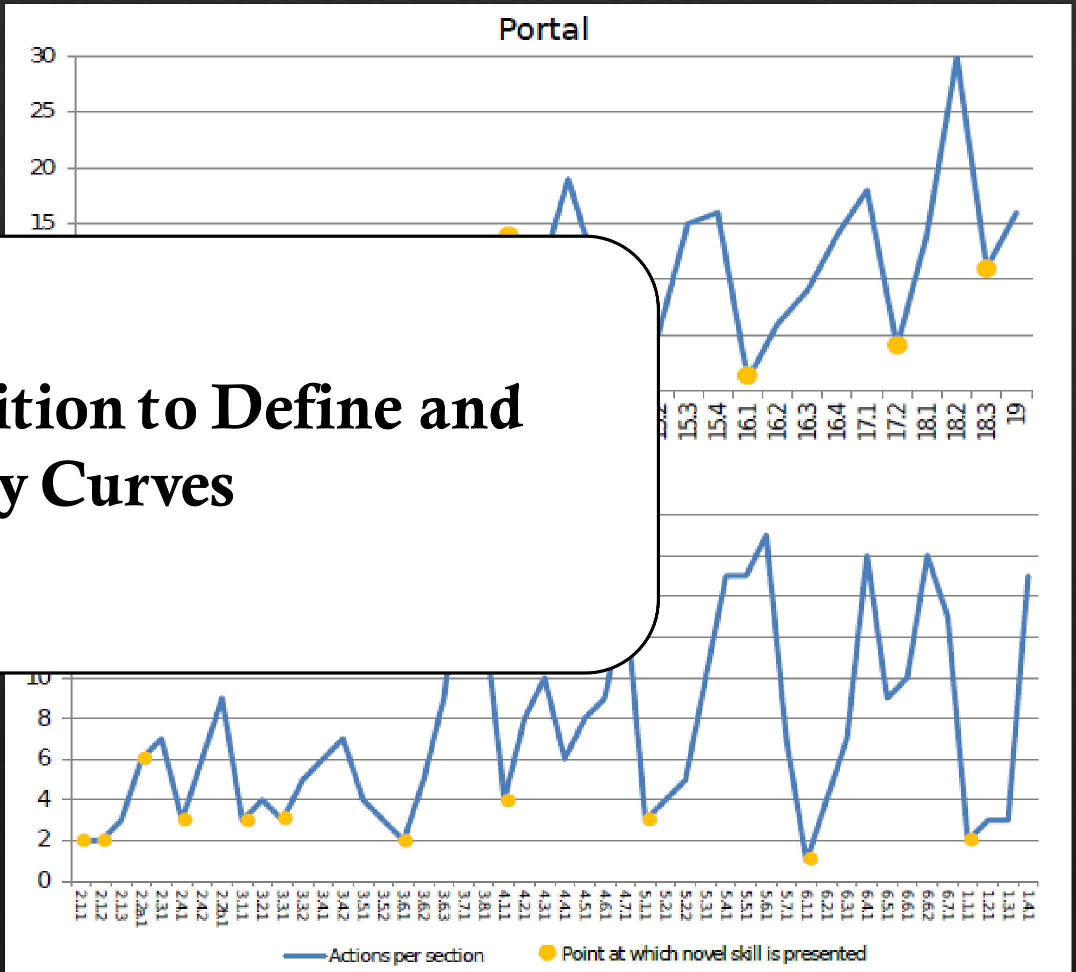


Linehan et al., 2014

Difficulty Curve

- Defines how a game's difficulty changes over the course of gameplay
- Curves can from progre
- In-game dif and so should be compatible with player skill
- Traditional methods of defining curves involve manual refinement through iterative playtesting

IDEA: Use Function Composition to Define and Transform Difficulty Curves

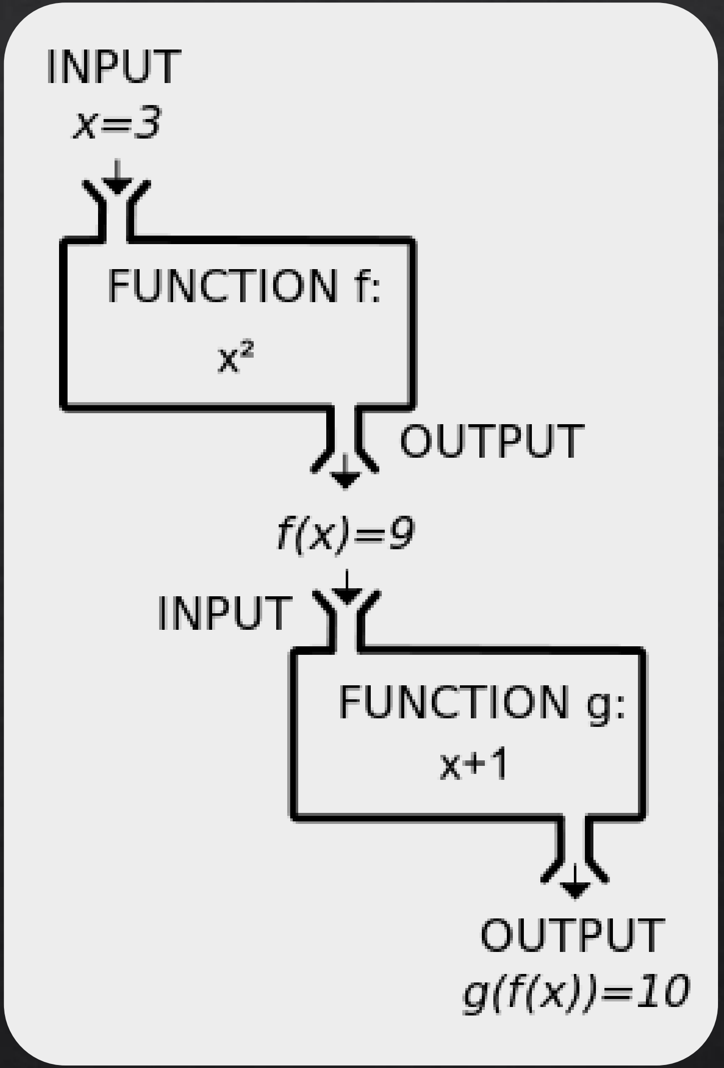


Linehan et al., 2014

Function Composition

- Multiple functions composed into one by applying one function to the output of another

- Given two functions $f(x)$ and $g(x)$, the composition of the functions $\boldsymbol{f \text{ o } g}$ is $f\big(g(x)\big)$ and $\boldsymbol{g \text{ o } f}$ is $g\big(f(x)\big)$
  - $f(x) = x^2,\ g(x) = x + 1$
  - $\boldsymbol{f \text{ o } g}\ (3) = f\big(g(3)\big) = (3{+}1)^2 = 16$
  - $\boldsymbol{g \text{ o } f}\ (3) = g\big(f(3)\big) = (3^2){+}1 = 10$



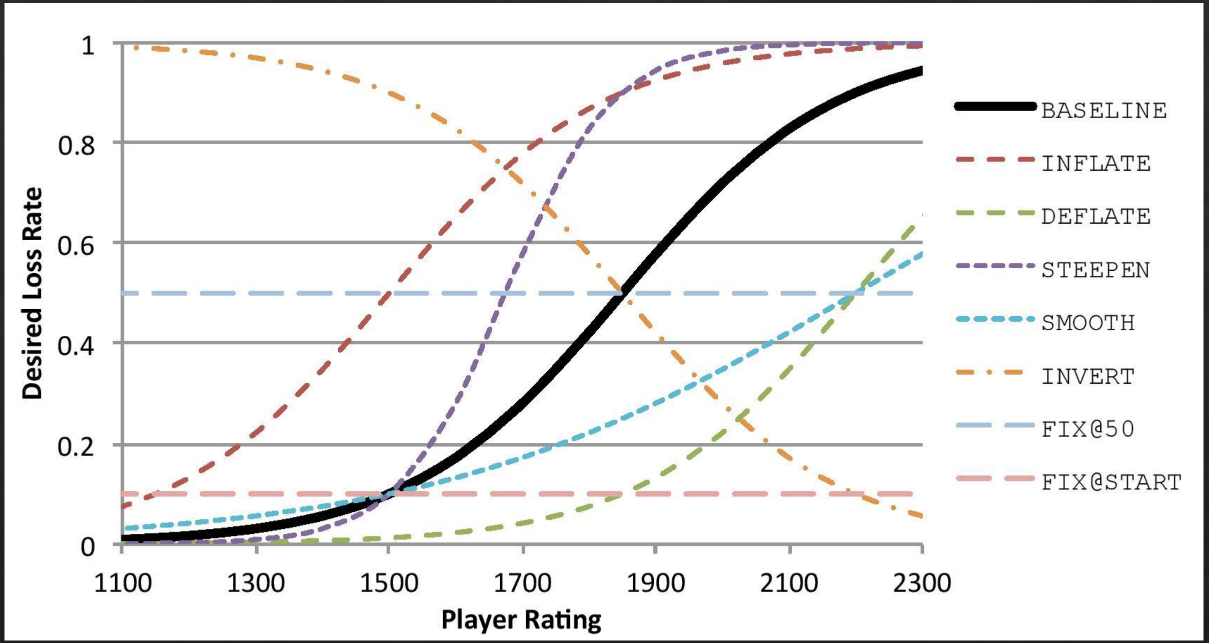
Motivation

- Precisely describe relative difficulty curves and transformations e.g. formalize a 'steep' curve

Motivation

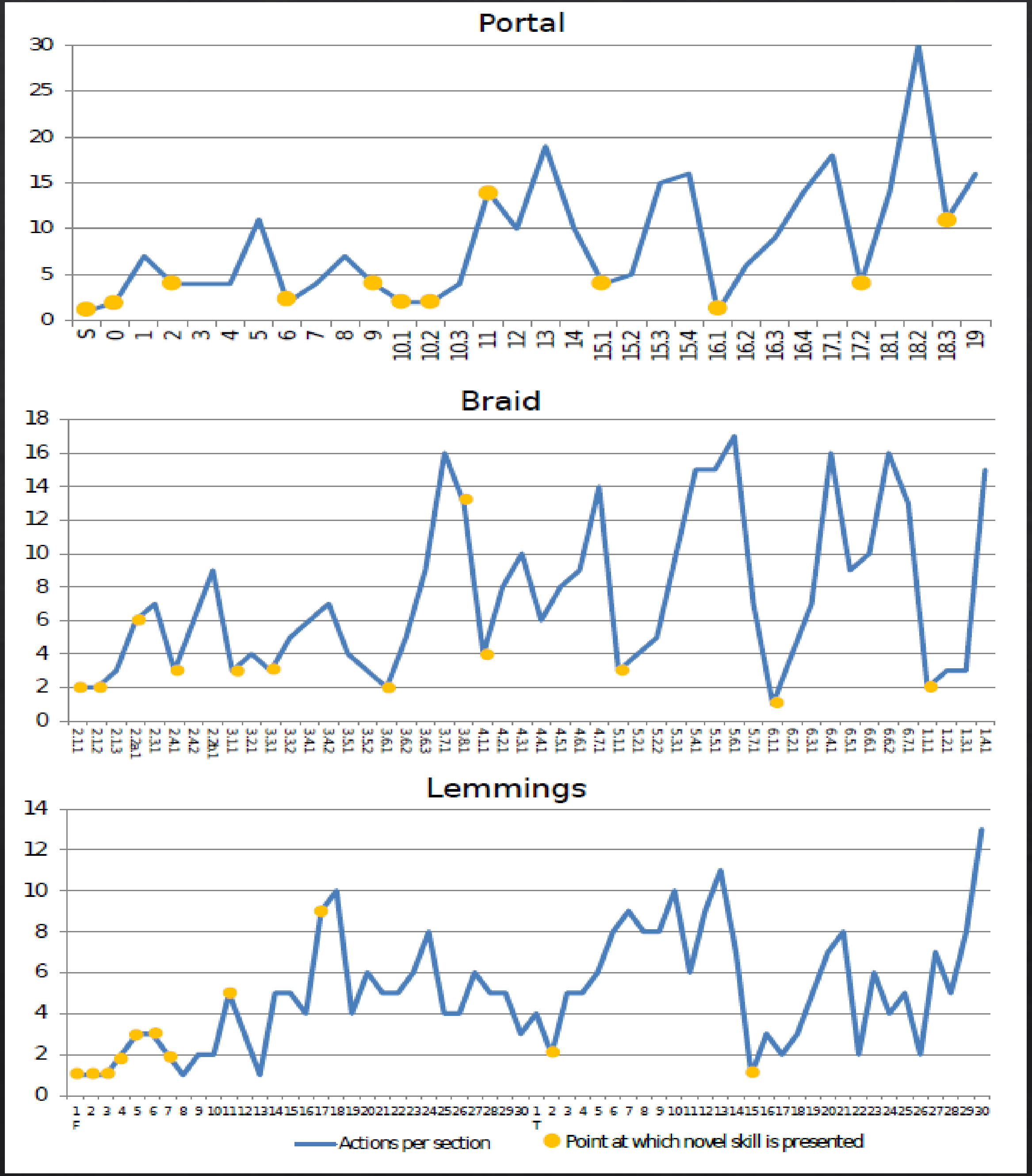
- Precisely describe relative difficulty curves and transformations e.g. formalize a ‘steep’ curve

- Functions (vs. manual refinement) capture a space of possible curves that can be explored



Motivation

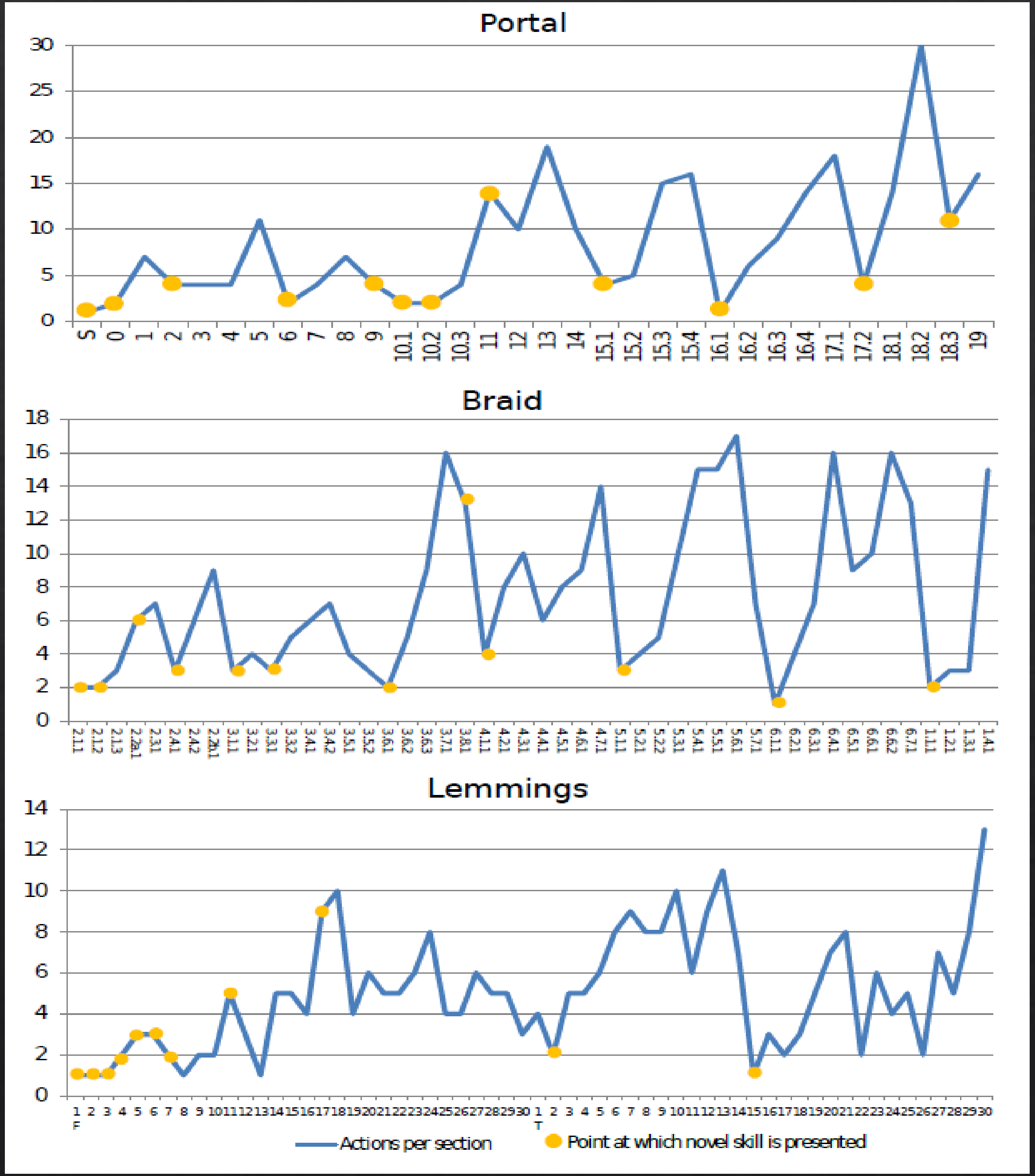
- ◈ Precisely describe relative difficulty curves and transformations e.g. formalize a ‘steep’ curve
- ◈ Functions (vs. manual refinement) capture a space of possible curves that can be explored
- ◈ Compare curves across games



Linehan et al., 2014

Motivation

- ◈ Precisely describe relative difficulty curves and transformations e.g. formalize a ‘steep’ curve
- ◈ Functions (vs. manual refinement) capture a space of possible curves that can be explored
- ◈ Compare curves across games
- ◈ Empirically evaluate impact of changing difficulty curves



Linehan et al., 2014

Experiment

- Applied function composition to transform the difficulty curve of the human computation puzzle game *Paradox* and tested:
    - if different transformations caused any changes in engagement/behavior
    - if such transformations could improve engagement benefits of the existing curve

Experiment

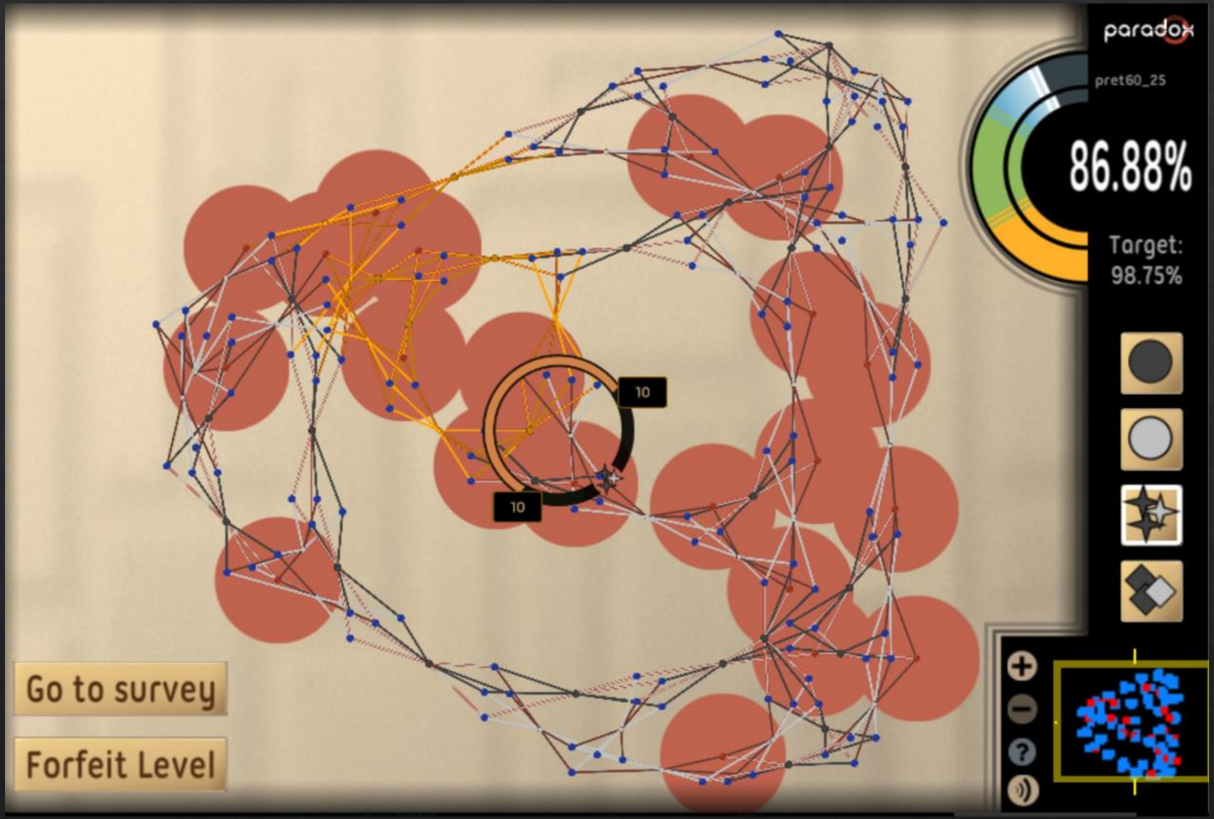
- Applied function composition to transform the difficulty curve of the human computation puzzle game *Paradox* and tested:
  - if different transformations caused any changes in engagement/behavior
  - if such transformations could improve engagement benefits of the existing curve

- *HYPOTHESIS*:

Transforming the difficulty curve using function composition impacts player behavior and experience with different transformations leading to different behavior and experience

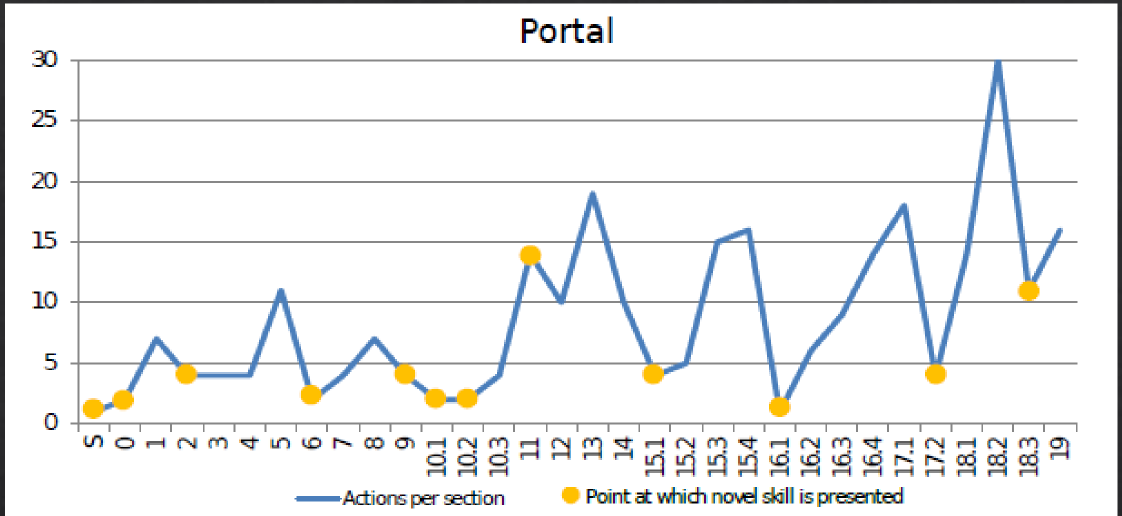
Paradox

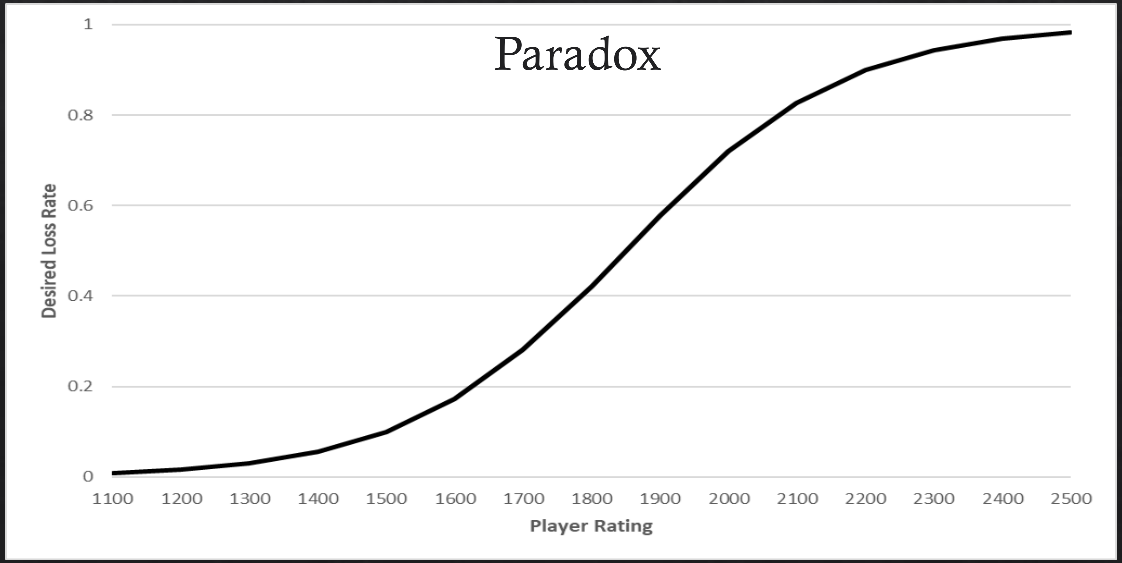
- 2D human computation puzzle game
- Each level is a boolean constraint satisfaction problem
- Players assign values to variables to solve constraints
- Score: percentage of satisfied constraints
- Target score reached → *Level Completed*



Player-vs-Level Matchmaking

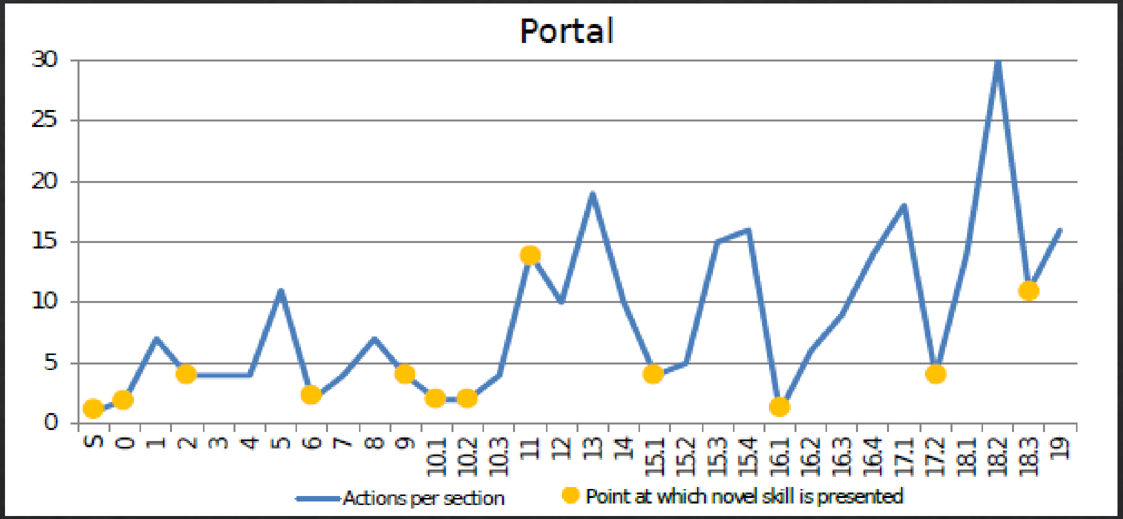
- Difficulty curve-based matchmaking system using Glicko-2 ratings to serve levels to players

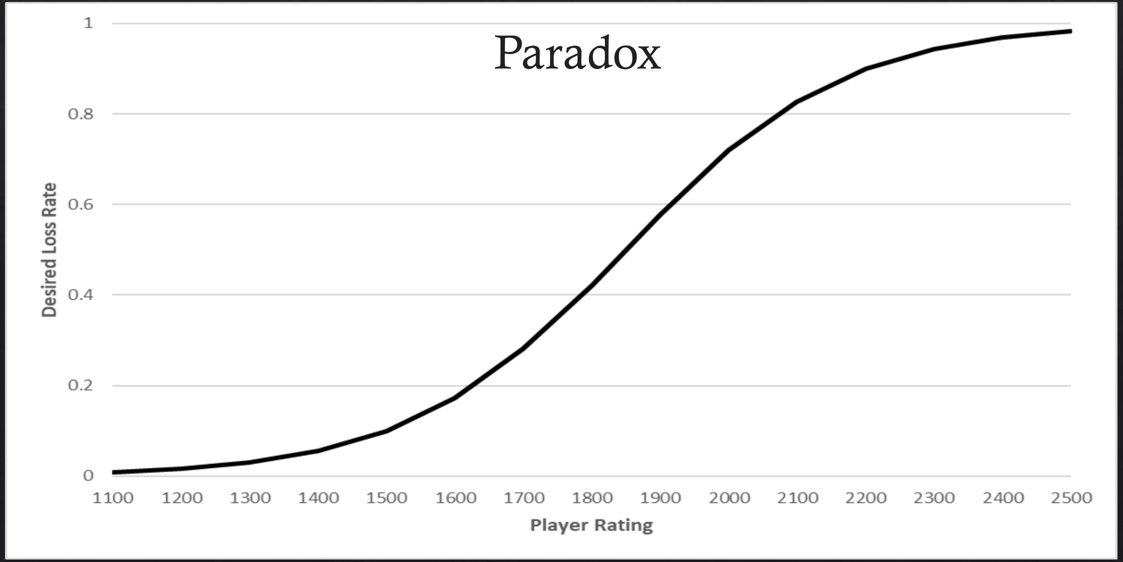




Player-vs-Level Matchmaking

- Difficulty curve-based matchmaking system using Glicko-2 ratings to serve levels to players
- Player and levels assigned ratings
  - Player rating → Skill
  - Level rating → Difficulty



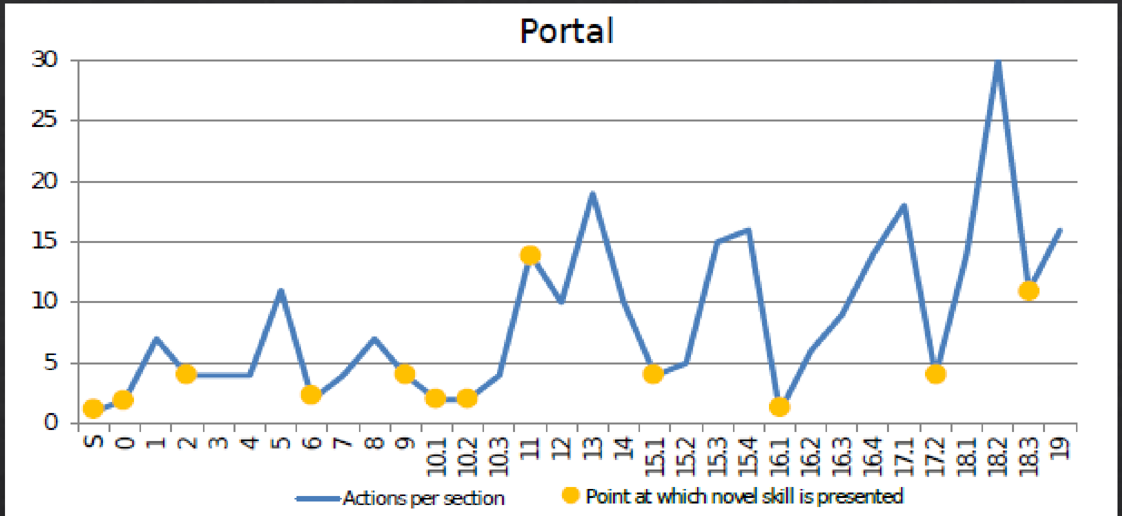


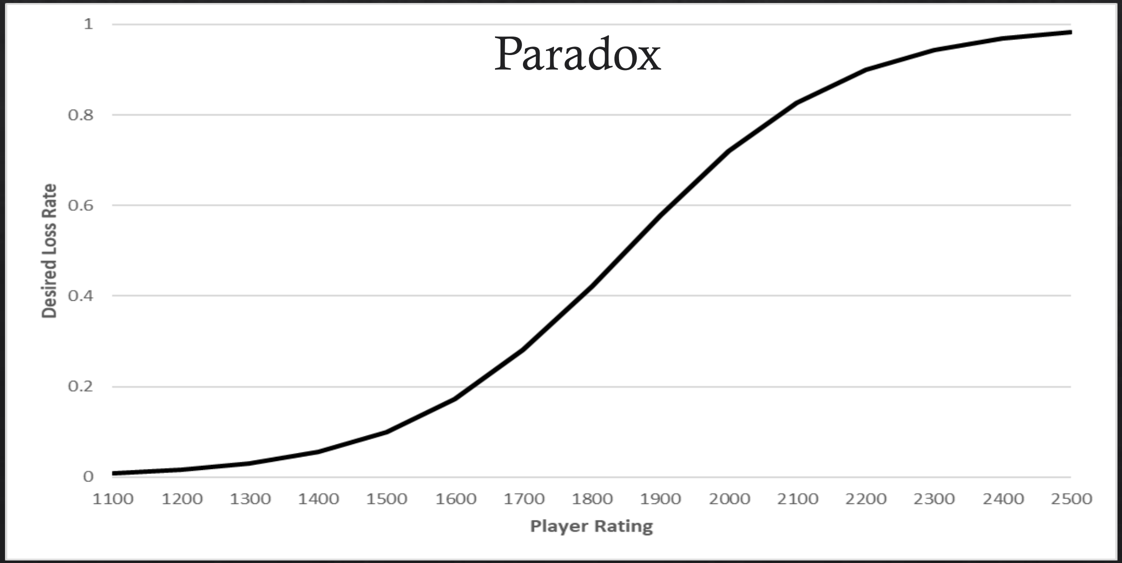
Player-vs-Level Matchmaking

- Difficulty curve-based matchmaking system using Glicko-2 ratings to serve levels to players

- Player and levels assigned ratings
  - Player rating → Skill
  - Level rating → Difficulty

- Compare ratings to compute player's chance of losing level i.e. level difficulty for that player





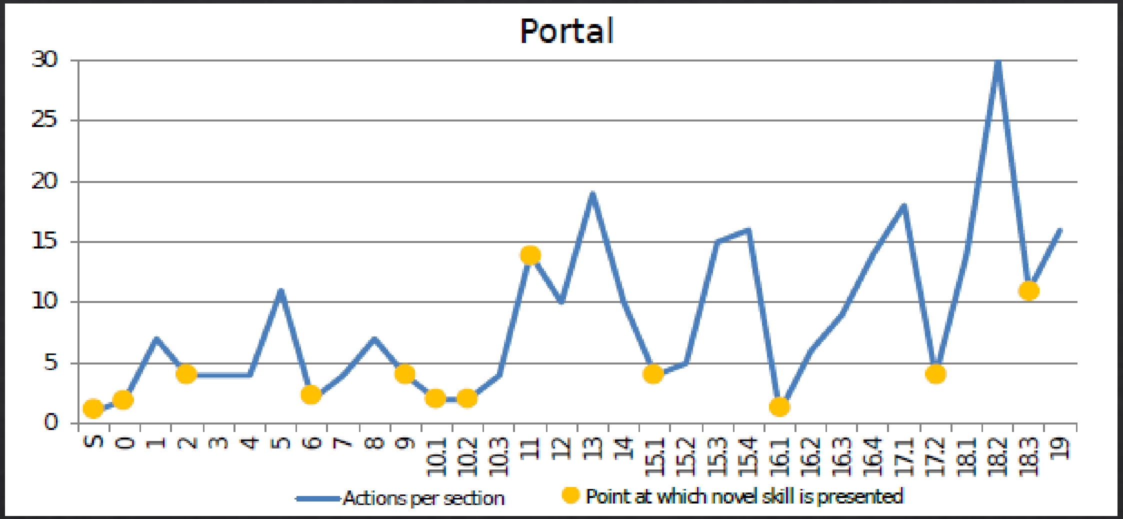
Player-vs-Level Matchmaking

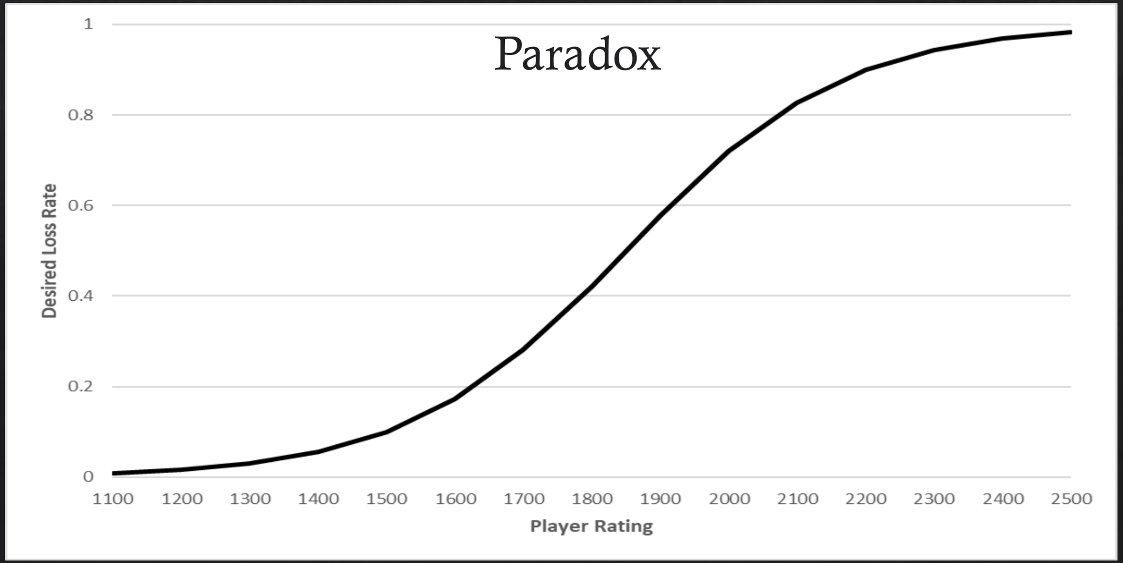
- Difficulty curve-based matchmaking system using Glicko-2 ratings to serve levels to players

- Player and levels assigned ratings
  - Player rating → Skill
  - Level rating → Difficulty

- Compare ratings to compute player's chance of losing level i.e. level difficulty for that player

- Use ratings-based loss estimates to determine next level as given by curve





Player-vs-Level Matchmaking

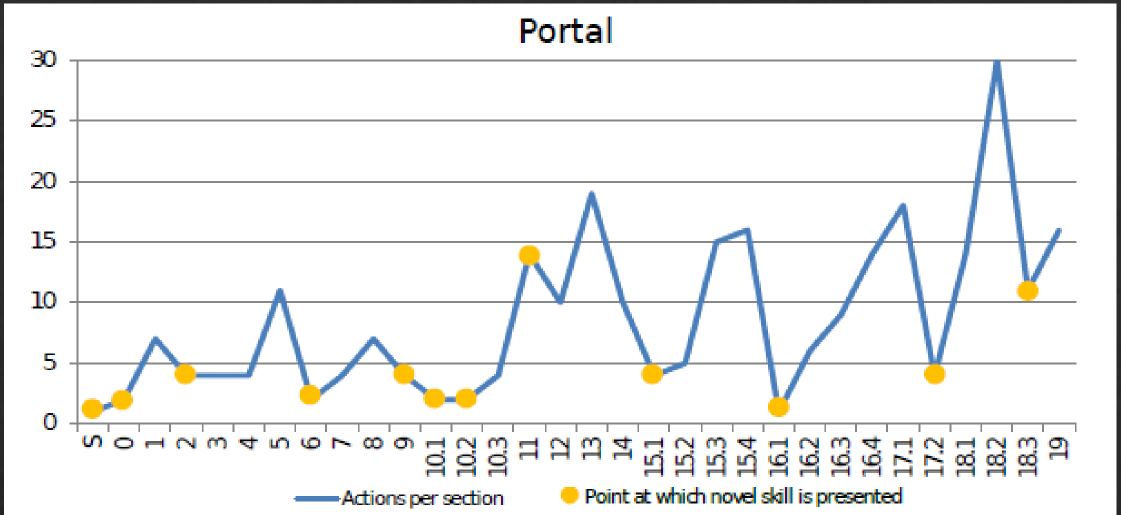
- Difficulty curve-based matchmaking system using Glicko-2 ratings to serve levels to players

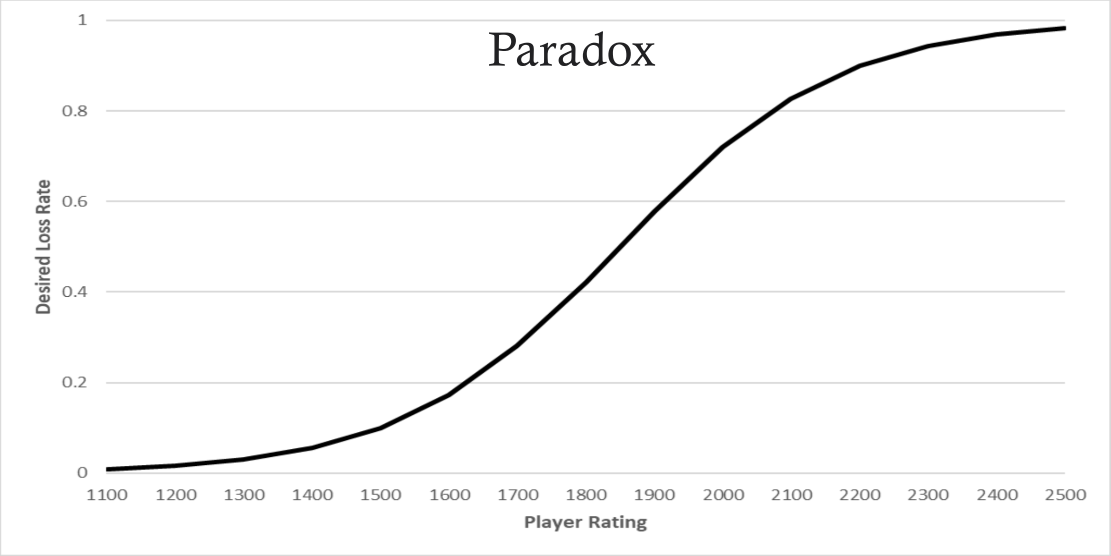
- Player and levels assigned ratings
  - Player rating → Skill
  - Level rating → Difficulty

- Compare ratings to compute player’s chance of losing level i.e. level difficulty for that player

- Use ratings-based loss estimates to determine next level as given by curve

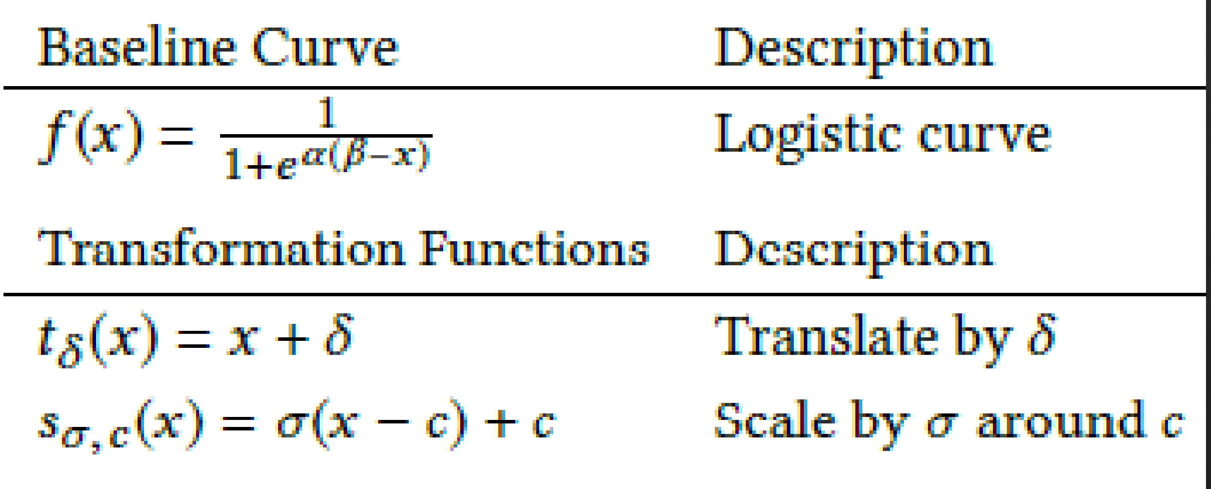
- Difficulty of game adapts to player’s skill



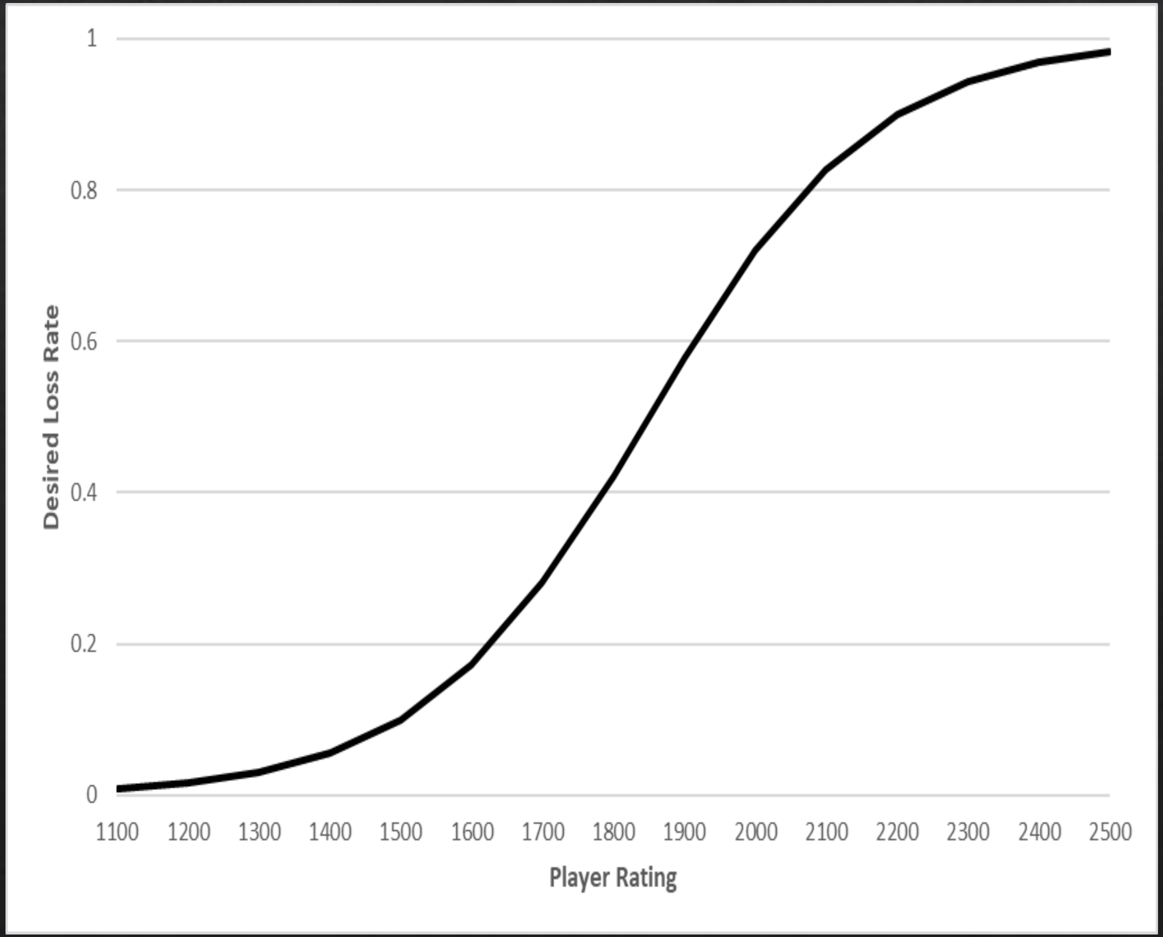


Curve Functions

Difficulty curve is a function mapping player skill (Glicko-2 rating) to difficulty (desired loss rate)

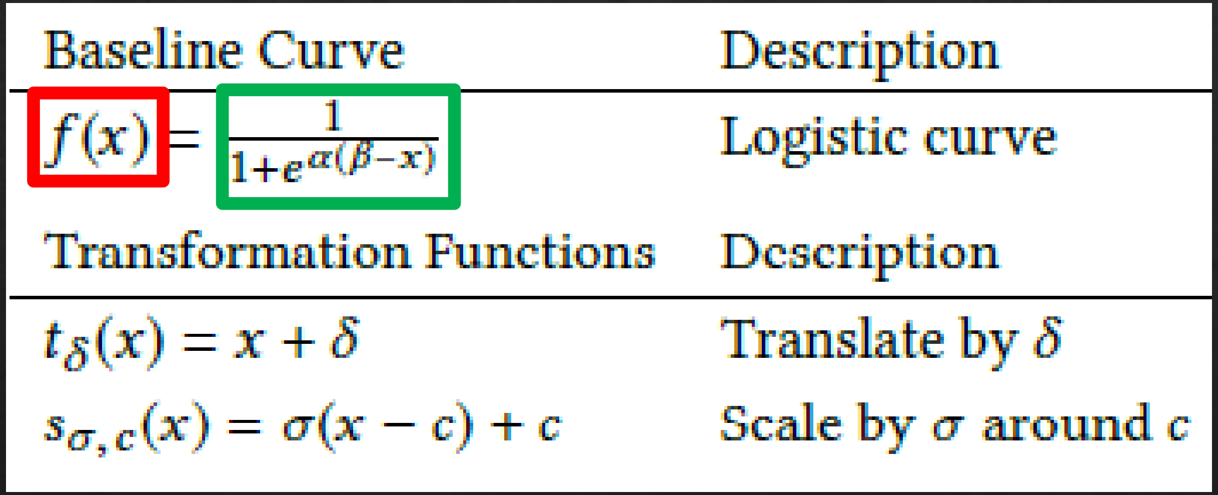


Baseline Curve	Description
$f(x) = \frac{1}{1+e^{\alpha(\beta-x)}}$	Logistic curve
Transformation Functions	Description
$t_\delta(x) = x + \delta$	Translate by δ
$s_{\sigma,c}(x) = \sigma(x - c) + c$	Scale by σ around c

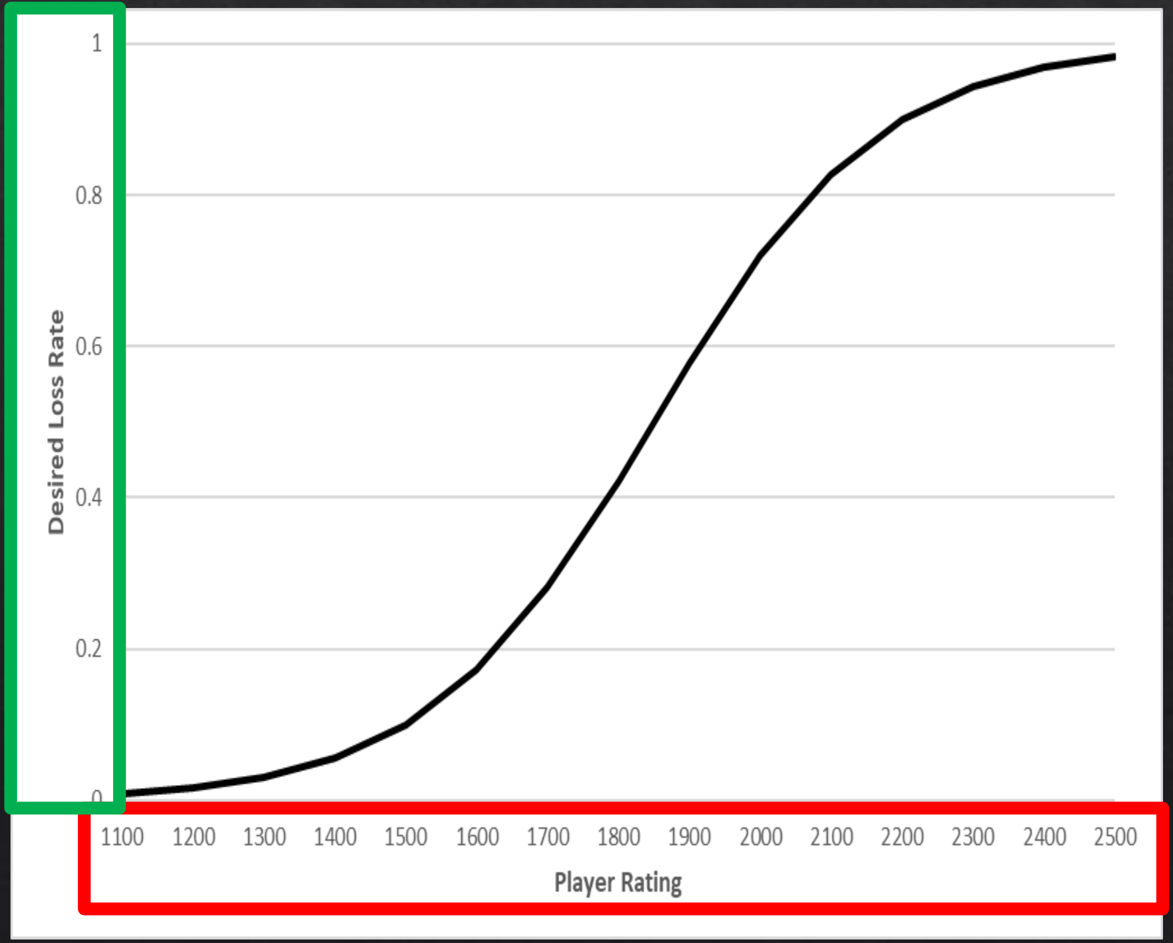


Curve Functions

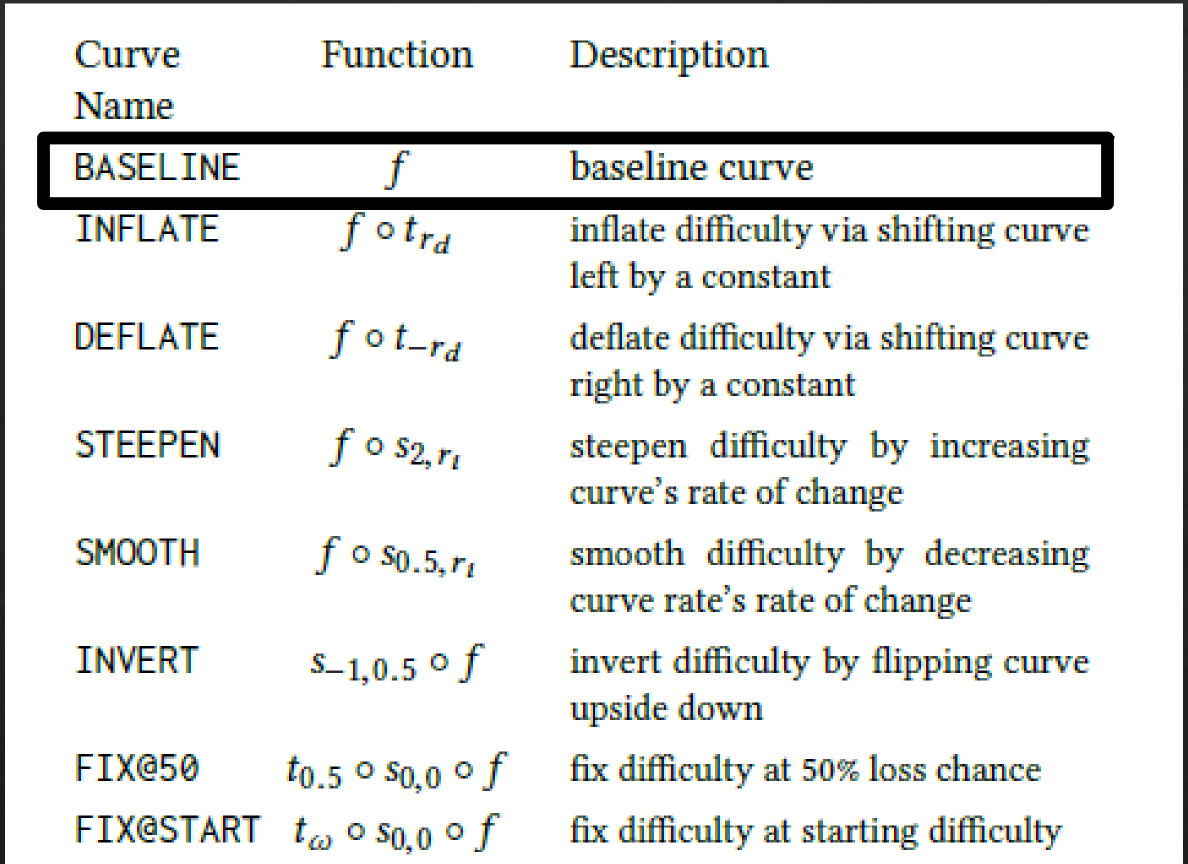
Difficulty curve is a function mapping player skill (Glicko-2 rating) to difficulty (desired loss rate)



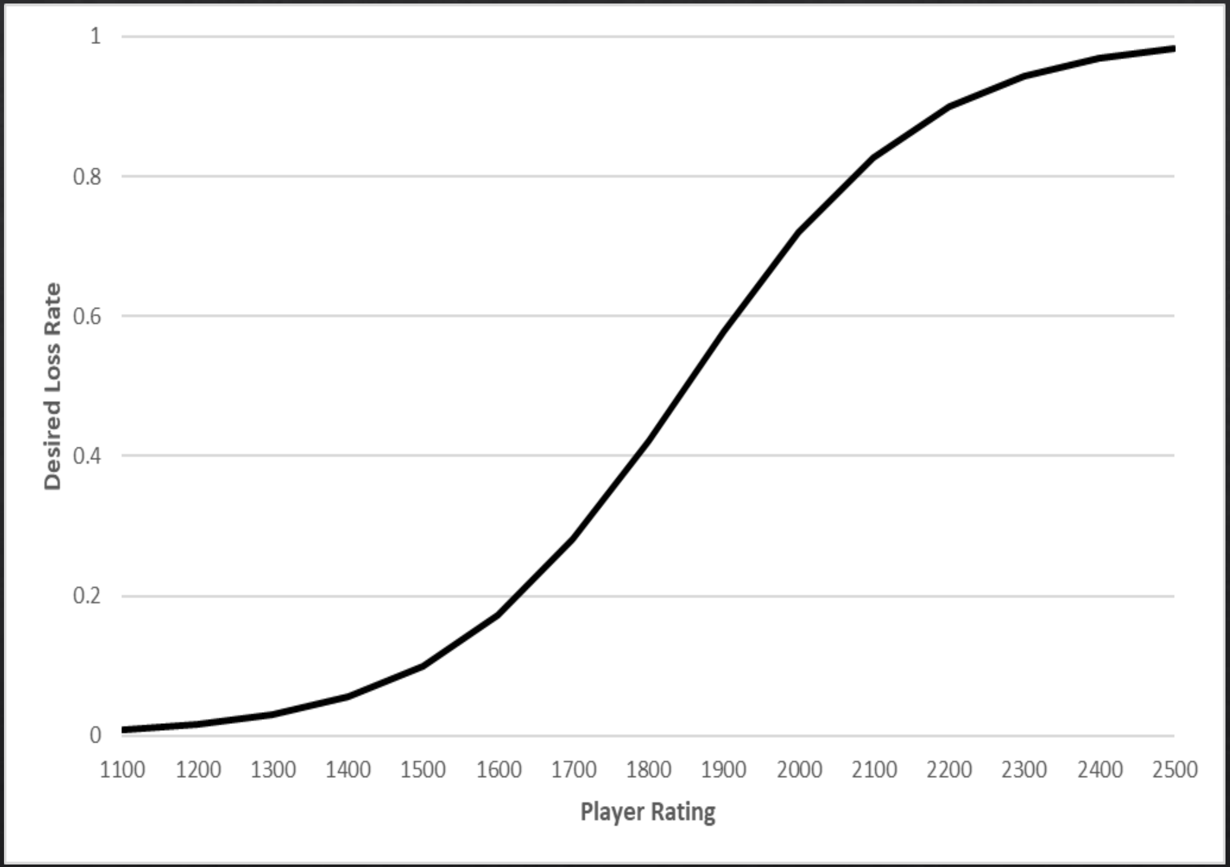
Baseline Curve	Description
$\boxed{f(x)} = \boxed{\frac{1}{1+e^{\alpha(\beta-x)}}}$	Logistic curve
Transformation Functions	**Description**
$t_\delta(x) = x + \delta$	Translate by δ
$s_{\sigma,c}(x) = \sigma(x - c) + c$	Scale by σ around c



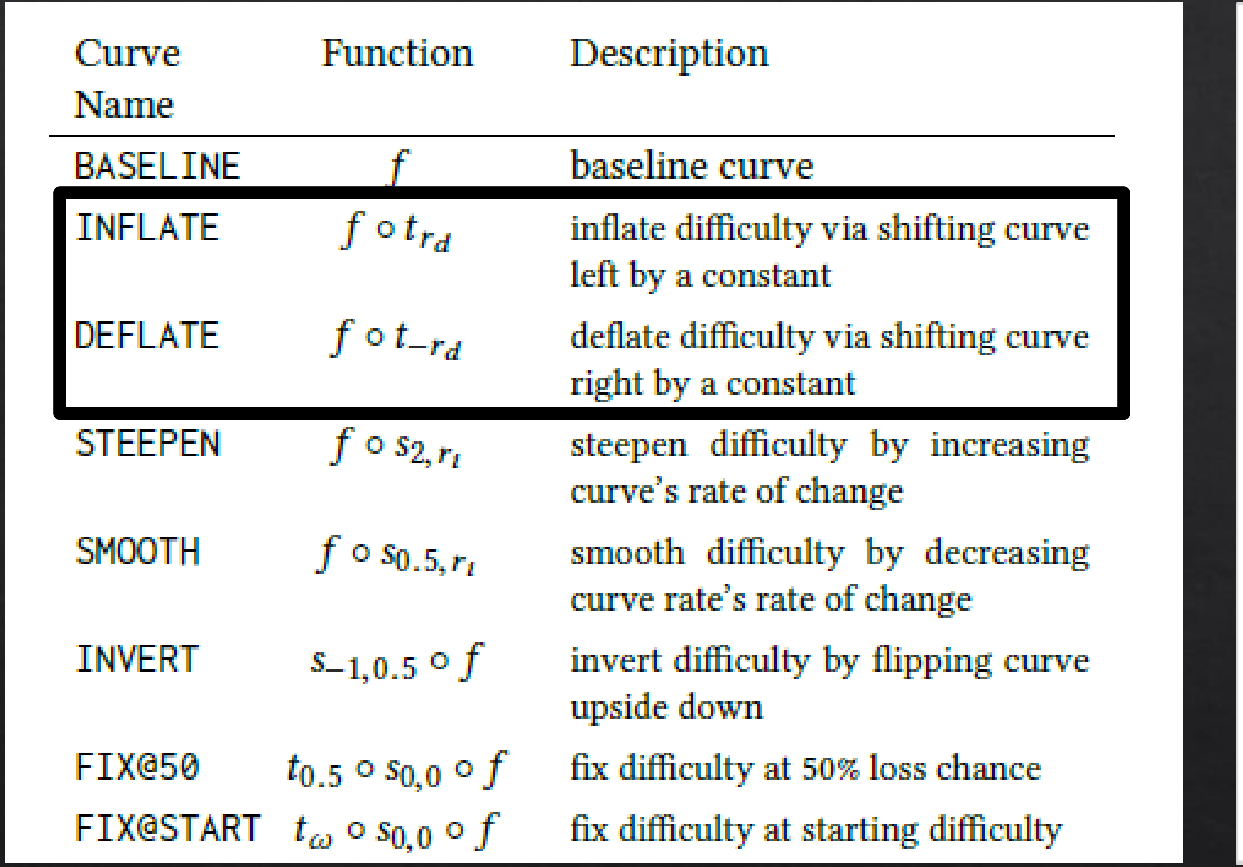
Curve Transformations



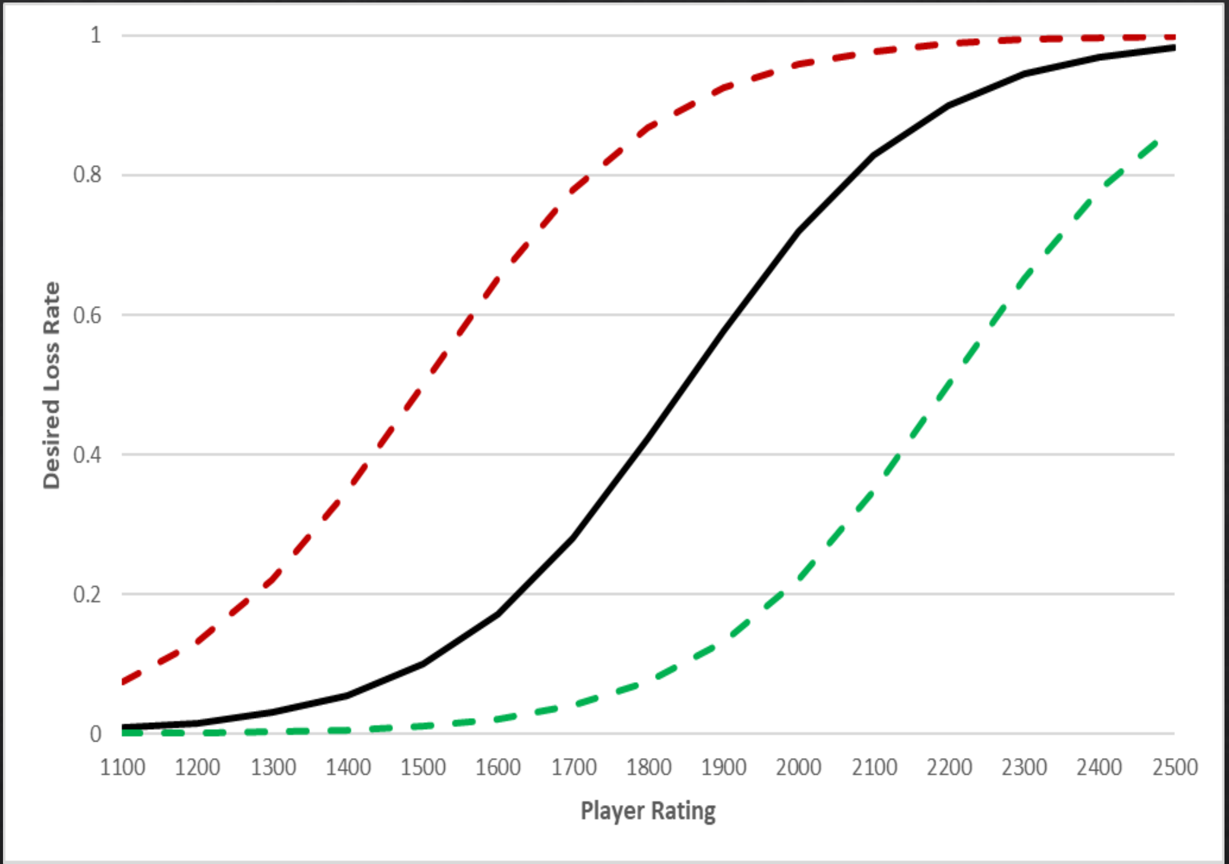
Curve Name	Function	Description
BASELINE	f	baseline curve
INFLATE	$f \circ t_{r_d}$	inflate difficulty via shifting curve left by a constant
DEFLATE	$f \circ t_{-r_d}$	deflate difficulty via shifting curve right by a constant
STEEPEN	$f \circ s_{2,r_t}$	steepen difficulty by increasing curve's rate of change
SMOOTH	$f \circ s_{0.5,r_t}$	smooth difficulty by decreasing curve rate's rate of change
INVERT	$s_{-1,0.5} \circ f$	invert difficulty by flipping curve upside down
FIX@50	$t_{0.5} \circ s_{0,0} \circ f$	fix difficulty at 50% loss chance
FIX@START	$t_\omega \circ s_{0,0} \circ f$	fix difficulty at starting difficulty



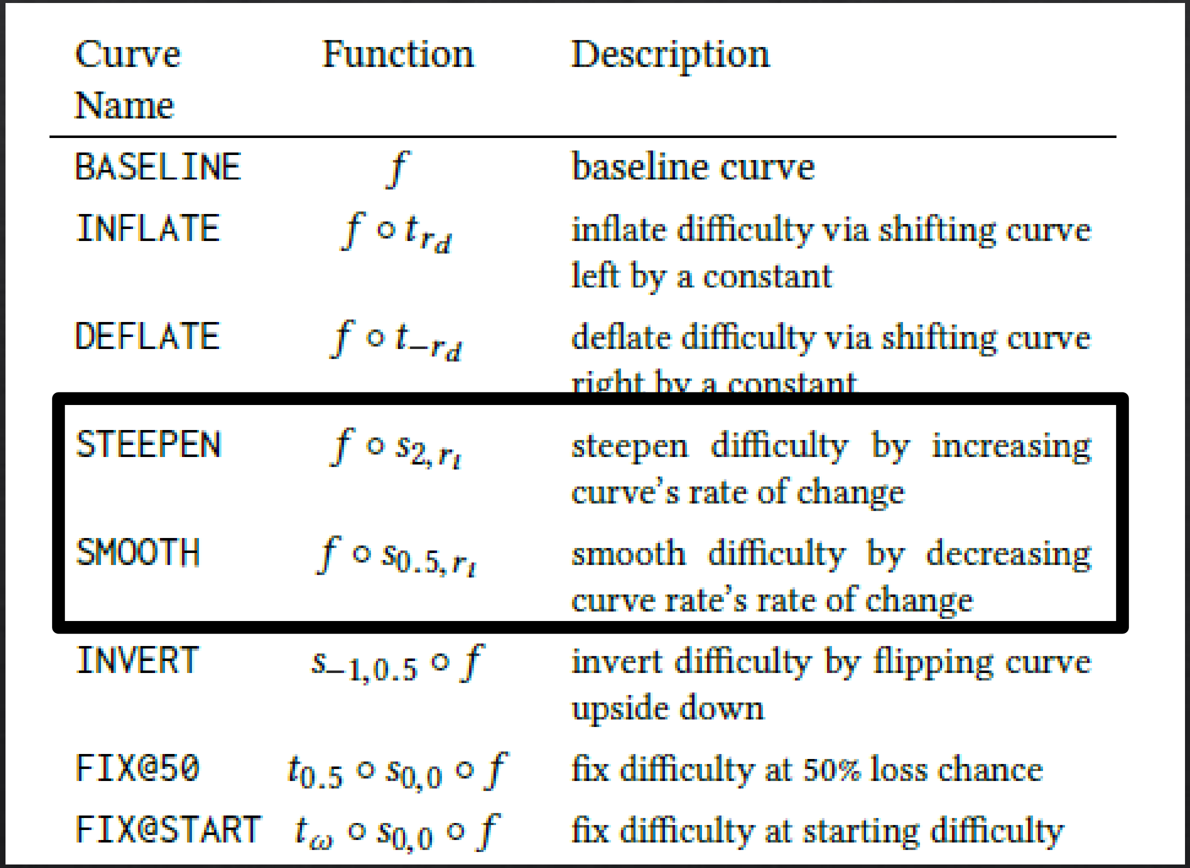
Curve Transformations



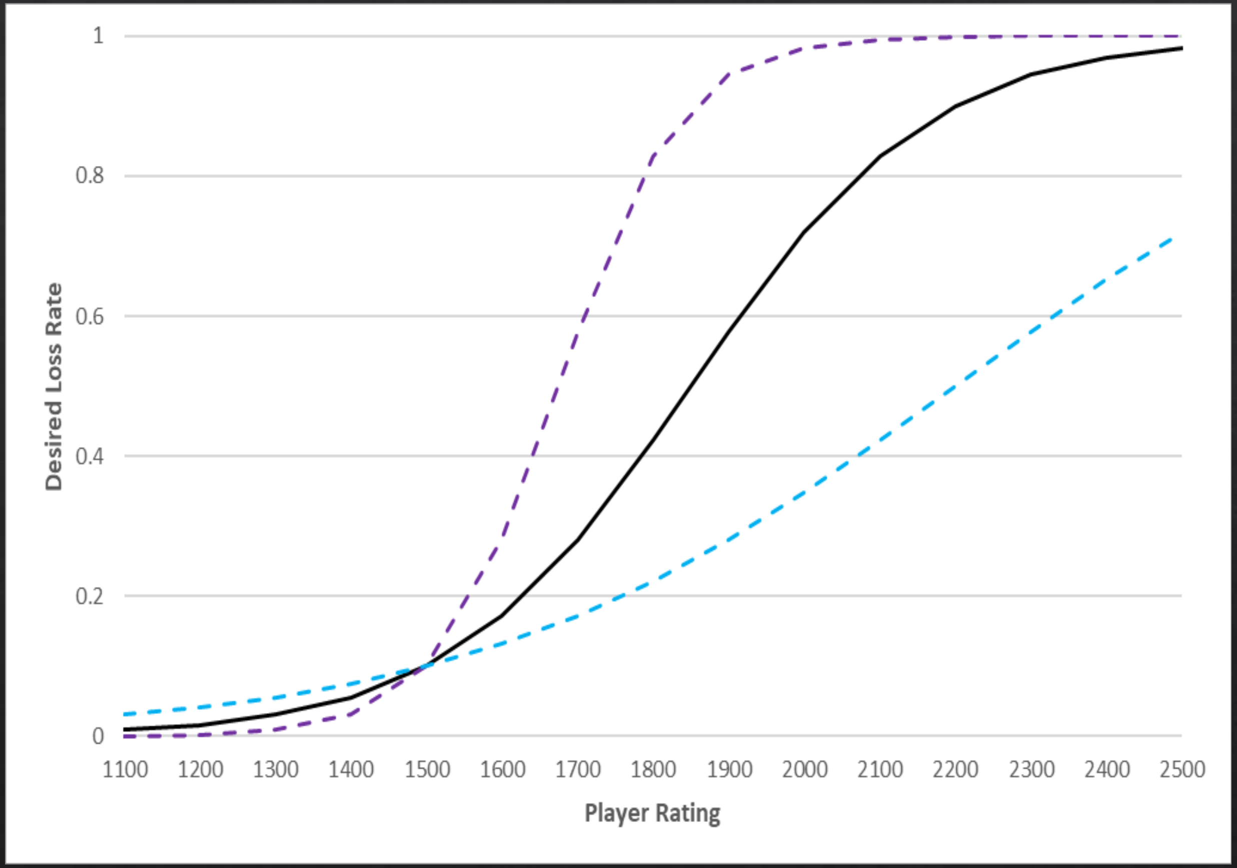
Curve Name	Function	Description
BASELINE	f	baseline curve
INFLATE	$f \circ t_{r_d}$	inflate difficulty via shifting curve left by a constant
DEFLATE	$f \circ t_{-r_d}$	deflate difficulty via shifting curve right by a constant
STEEPEN	$f \circ s_{2,r_t}$	steepen difficulty by increasing curve's rate of change
SMOOTH	$f \circ s_{0.5,r_t}$	smooth difficulty by decreasing curve rate's rate of change
INVERT	$s_{-1,0.5} \circ f$	invert difficulty by flipping curve upside down
FIX@50	$t_{0.5} \circ s_{0,0} \circ f$	fix difficulty at 50% loss chance
FIX@START	$t_{\omega} \circ s_{0,0} \circ f$	fix difficulty at starting difficulty



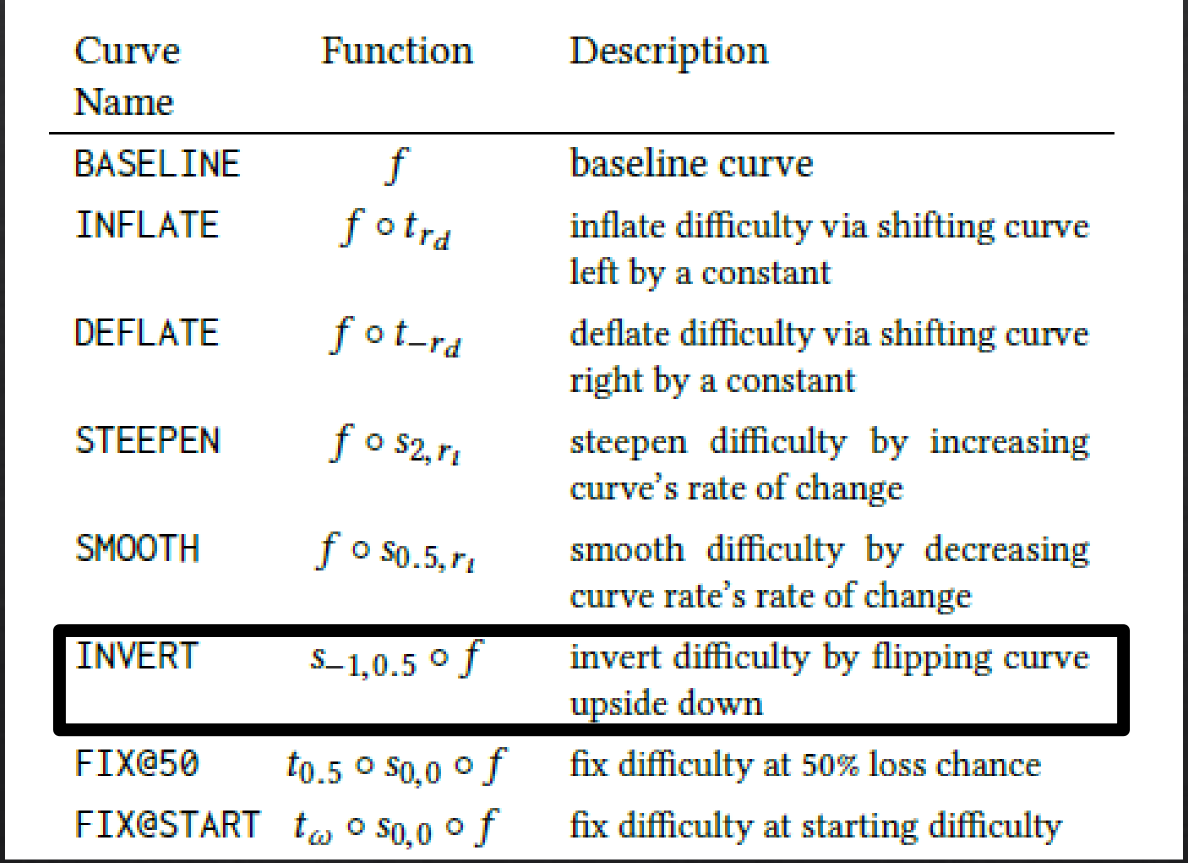
Curve Transformations



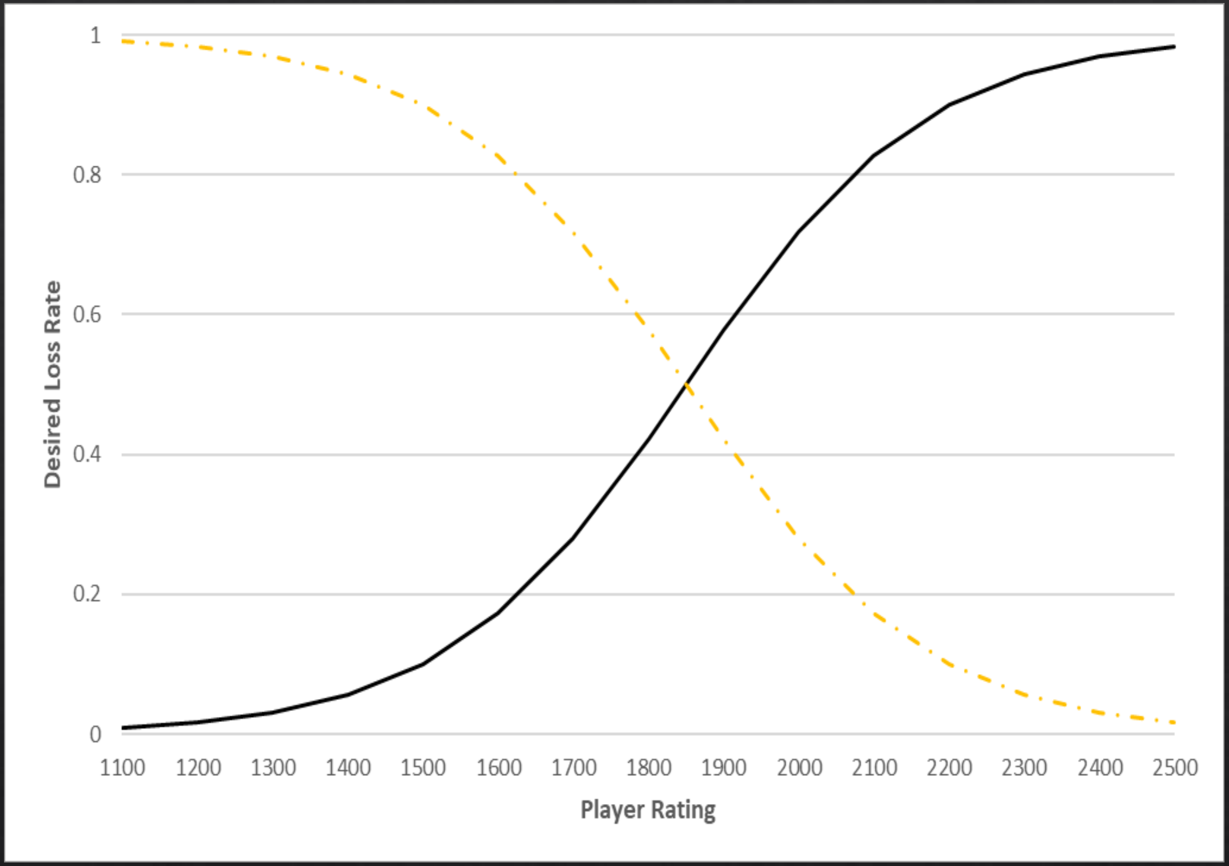
Curve Name	Function	Description
BASELINE	f	baseline curve
INFLATE	$f \circ t_{r_d}$	inflate difficulty via shifting curve left by a constant
DEFLATE	$f \circ t_{-r_d}$	deflate difficulty via shifting curve right by a constant
STEEPEN	$f \circ s_{2, r_\iota}$	steepen difficulty by increasing curve's rate of change
SMOOTH	$f \circ s_{0.5, r_\iota}$	smooth difficulty by decreasing curve rate's rate of change
INVERT	$s_{-1,0.5} \circ f$	invert difficulty by flipping curve upside down
FIX@50	$t_{0.5} \circ s_{0,0} \circ f$	fix difficulty at 50% loss chance
FIX@START	$t_\omega \circ s_{0,0} \circ f$	fix difficulty at starting difficulty



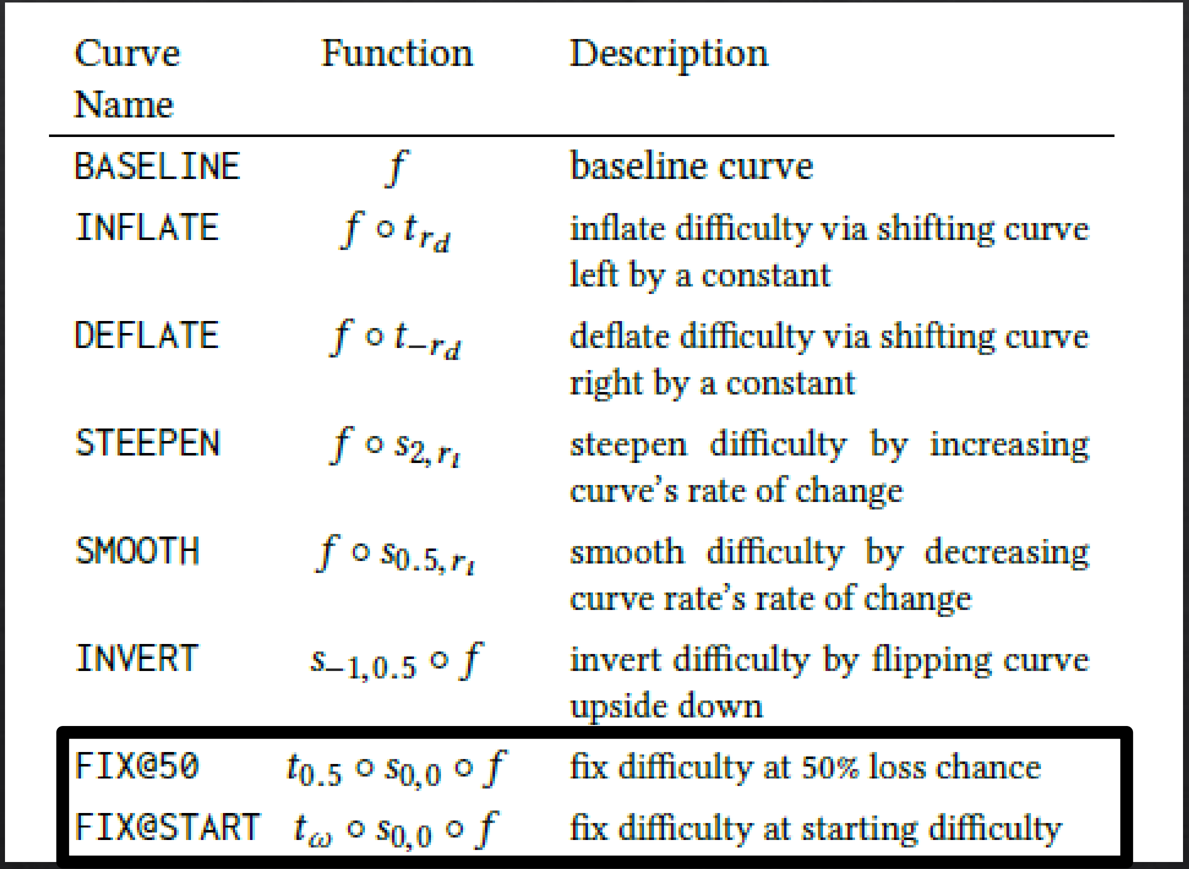
Curve Transformations



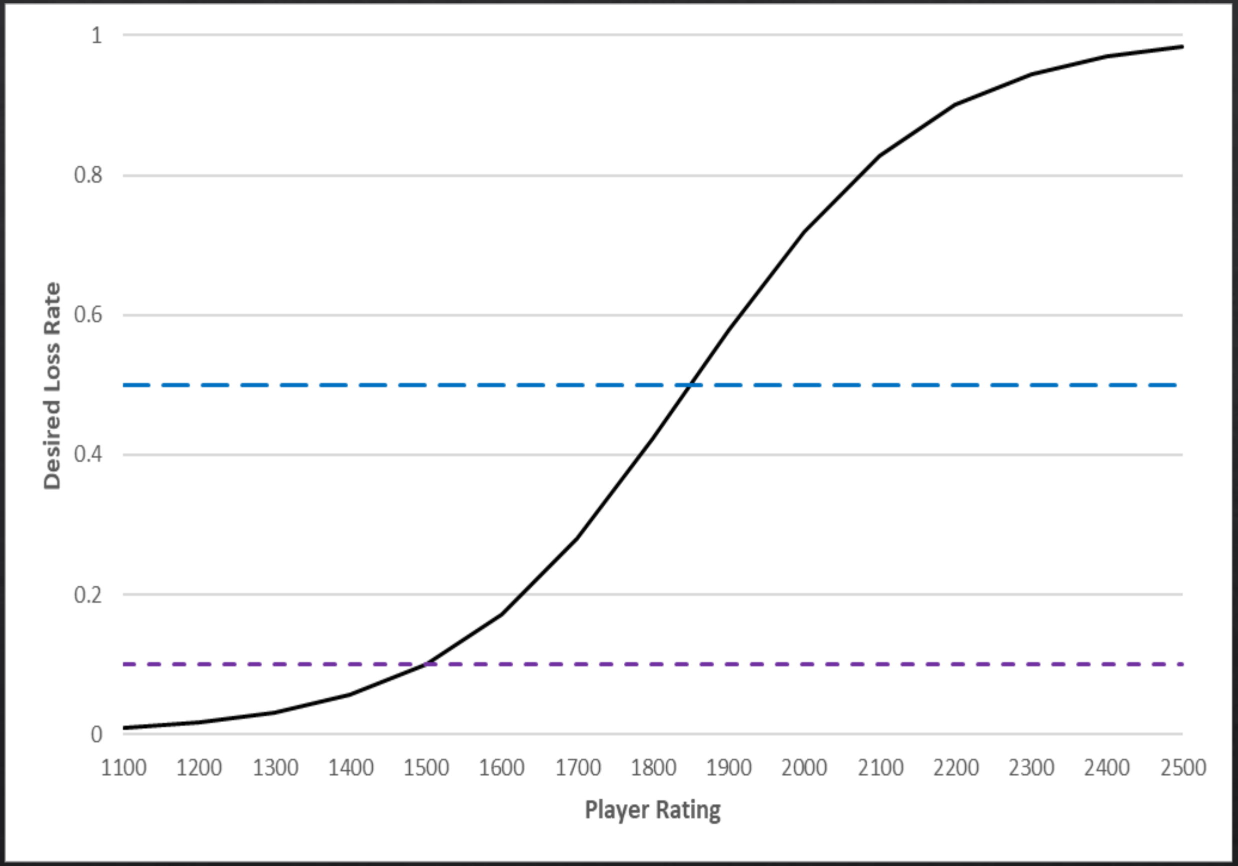
Curve Name	Function	Description
BASELINE	f	baseline curve
INFLATE	$f \circ t_{r_d}$	inflate difficulty via shifting curve left by a constant
DEFLATE	$f \circ t_{-r_d}$	deflate difficulty via shifting curve right by a constant
STEEPEN	$f \circ s_{2,r_t}$	steepen difficulty by increasing curve's rate of change
SMOOTH	$f \circ s_{0.5,r_t}$	smooth difficulty by decreasing curve rate's rate of change
INVERT	$s_{-1,0.5} \circ f$	invert difficulty by flipping curve upside down
FIX@50	$t_{0.5} \circ s_{0,0} \circ f$	fix difficulty at 50% loss chance
FIX@START	$t_{\omega} \circ s_{0,0} \circ f$	fix difficulty at starting difficulty



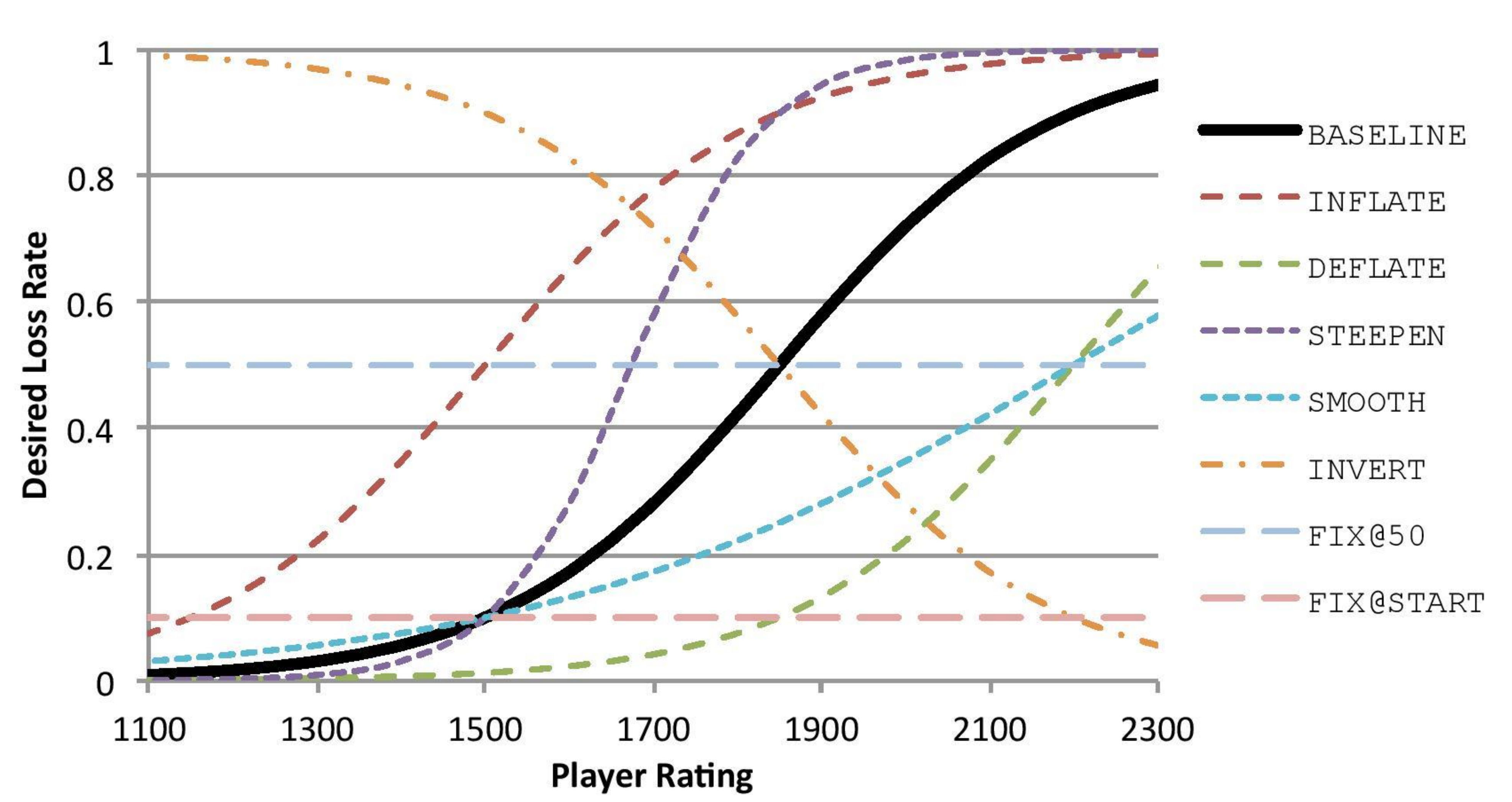
Curve Transformations



Curve Name	Function	Description
BASELINE	f	baseline curve
INFLATE	$f \circ t_{r_d}$	inflate difficulty via shifting curve left by a constant
DEFLATE	$f \circ t_{-r_d}$	deflate difficulty via shifting curve right by a constant
STEEPEN	$f \circ s_{2,r_I}$	steepen difficulty by increasing curve's rate of change
SMOOTH	$f \circ s_{0.5,r_I}$	smooth difficulty by decreasing curve rate's rate of change
INVERT	$s_{-1,0.5} \circ f$	invert difficulty by flipping curve upside down
FIX@50	$t_{0.5} \circ s_{0,0} \circ f$	fix difficulty at 50% loss chance
FIX@START	$t_{\omega} \circ s_{0,0} \circ f$	fix difficulty at starting difficulty



Curve Transformations

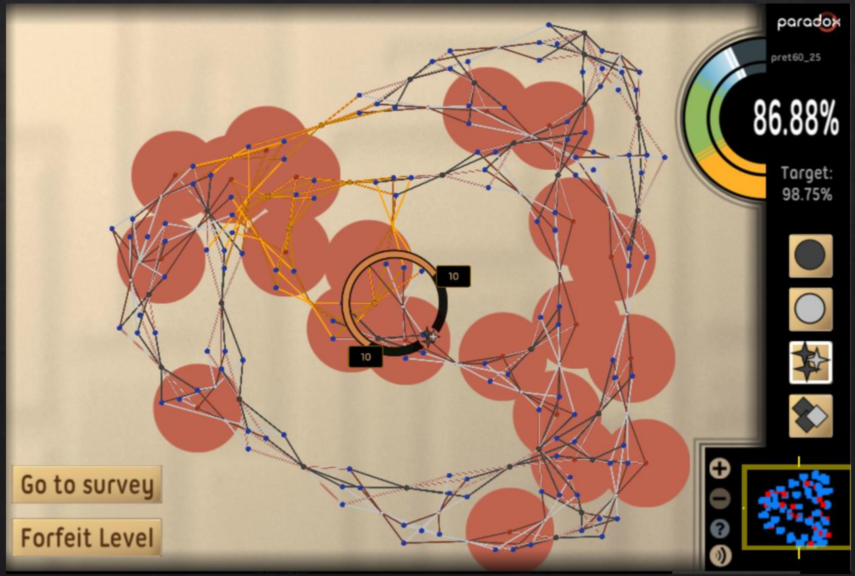


Participant Recruitment and Study

- Players recruited using Mechanical Turk

amazon mechanical turk™
Artificial Artificial Intelligence

- 400 players randomly assigned to one of the 8 difficulty curves
  - 8 tutorial levels (static order)
  - 50 challenge levels (dynamic difficulty order)
  - Post-game Intrinsic Motivation Inventory (IMI)



Measures of Engagement

- Behavioral Engagement
  - *Challenge Time*
  - *Levels Attempted*
  - *Levels Completed*
  - *Player Rating*

    (Player's Glicko-2 rating after finishing playing)
  - *Highest Level Rating*

    (Highest Glicko-2 rating of any level completed by the player)

- Intrinsic Motivation Inventory
  - *Interest/Enjoyment*
  - *Perceived Competence*
  - *Effort/Importance*

Results

	Play Time	Levels Attempted	Levels Completed	Perceived Competence	Highest Level Rating
INVERT	516	***4***	***0***	***23***	1880
INFLATE	433	***4***	***2***	***17***	1517
FIX@50	527	5	3	16	1587
FIX@START	413	6	5	26	1222
STEEPEN	618	7	4.5	25	1587
BASELINE	610	10	7	25	1260
SMOOTH	762	***10***	***8***	***28***	1416
DEFLATE	682	***15***	***14***	***28***	1328

Statistical Tests: Aligned Rank Transform, post-hoc Wilcoxon Rank-Sum Test

- No significant omnibus difference across curves for *Player Rating* and *Effort/Importance*
- No post-hoc differences for *Interest/Enjoyment*

Results

	Play Time	**Levels Attempted**	**Levels Completed**	**Perceived Competence**	**Highest Level Rating**
INVERT	516	***4***	***0***	***23***	1880
INFLATE	433	***4***	***2***	***17***	1517
FIX@50	527	5	3	16	1587
FIX@START	413	6	5	26	1222
STEEPEN	618	7	4.5	25	1587
BASELINE	610	10	7	25	1260
SMOOTH	762	***10***	***8***	***28***	1416
DEFLATE	682	***15***	***14***	***28***	1328

Statistical Tests: Aligned Rank Transform, post-hoc Wilcoxon Rank-Sum Test

- Some ramp-up in difficulty may be more engaging than a fixed, low level of difficulty

Results

	Play Time	Levels Attempted	Levels Completed	Perceived Competence	Highest Level Rating
INVERT	516	***4***	***0***	***23***	1880
INFLATE	433	***4***	***2***	***17***	1517
FIX@50	527	5	3	16	1587
FIX@START	413	6	5	26	1222
STEEPEN	618	7	4.5	25	1587
BASELINE	610	10	7	25	1260
SMOOTH	762	***10***	***8***	***28***	1416
DEFLATE	682	***15***	***14***	***28***	1328

Statistical Tests: Aligned Rank Transform, post-hoc Wilcoxon Rank-Sum Test

- *Levels Attempted*, *Levels Completed* and *Perceived Competence* increased by making curve ‘easier’

Results

	Play Time	Levels Attempted	Levels Completed	Perceived Competence	Highest Level Rating
INVERT	516	***4***	***0***	***23***	1880
INFLATE	433	***4***	***2***	***17***	1517
FIX@50	527	5	3	16	1587
FIX@START	413	6	5	26	1222
STEEPEN	618	7	4.5	25	1587
BASELINE	610	10	7	25	1260
SMOOTH	762	***10***	***8***	***28***	1416
DEFLATE	682	***15***	***14***	***28***	1328

Statistical Tests: Aligned Rank Transform, post-hoc Wilcoxon Rank-Sum Test

- *Levels Attempted*, *Levels Completed* and *Perceived Competence* increased by making curve ‘easier’
- *Highest Level Rating* increased by making curve ‘harder’

Results

	Play Time	Levels Attempted	Levels Completed	Perceived Competence	Highest Level Rating
INVERT	516	***4***	***0***	***23***	1880
INFLATE	[illegible]	***4***	[illegible]	***17***	1517
FIX	[illegible]	[illegible]	[illegible]	[illegible]	1587
FIX@S	[illegible]	[illegible]	[illegible]	[illegible]	1222
STEP	[illegible]	[illegible]	[illegible]	[illegible]	1587
BASEL	[illegible]	[illegible]	[illegible]	[illegible]	1260
SMOOTH	762	***10***	***8***	***28***	1416
DEFLATE	682	***15***	***14***	***28***	1328

Transforming difficulty curves did impact player engagement thus supporting our hypothesis that different curve transformations would affect player behavior and experience

Statistical Tests: Aligned Rank Transform, post-hoc Wilcoxon Rank-Sum Test

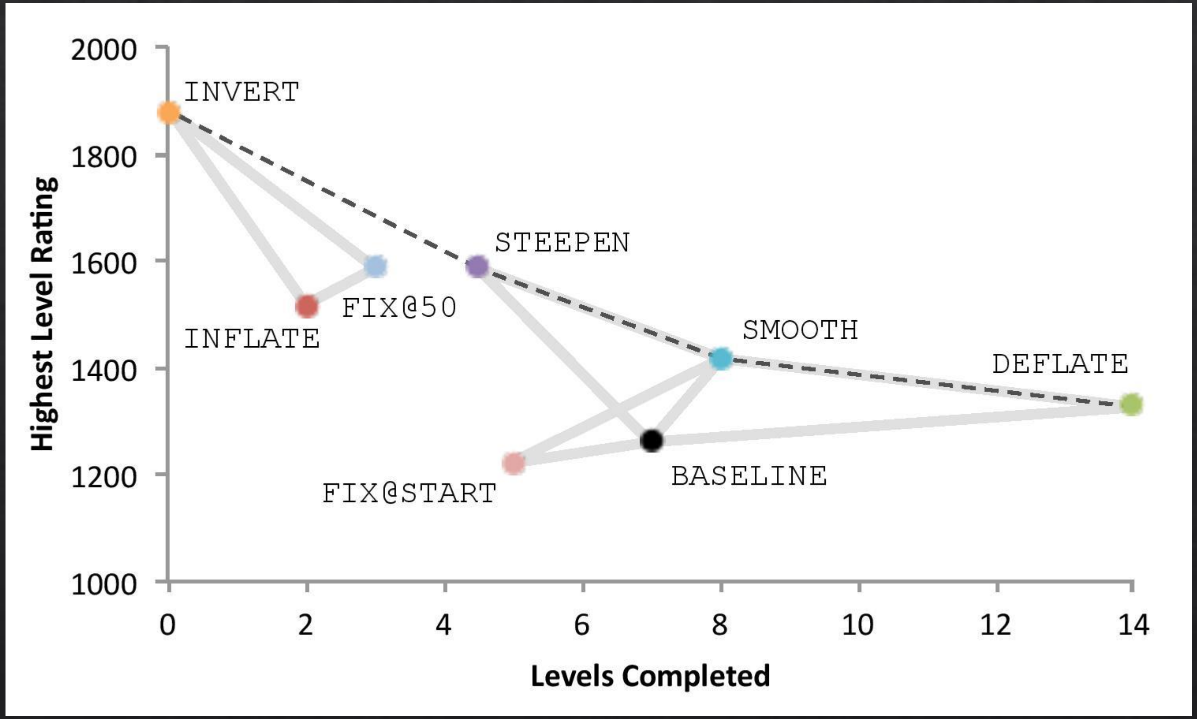
Pareto Efficiency

- HCGs try to maximize the number and/or quality of in-game tasks that players complete
  - Trade off between amount of work done *(Levels Completed)* and its quality (*Highest Level Rating)*

Pareto Efficiency

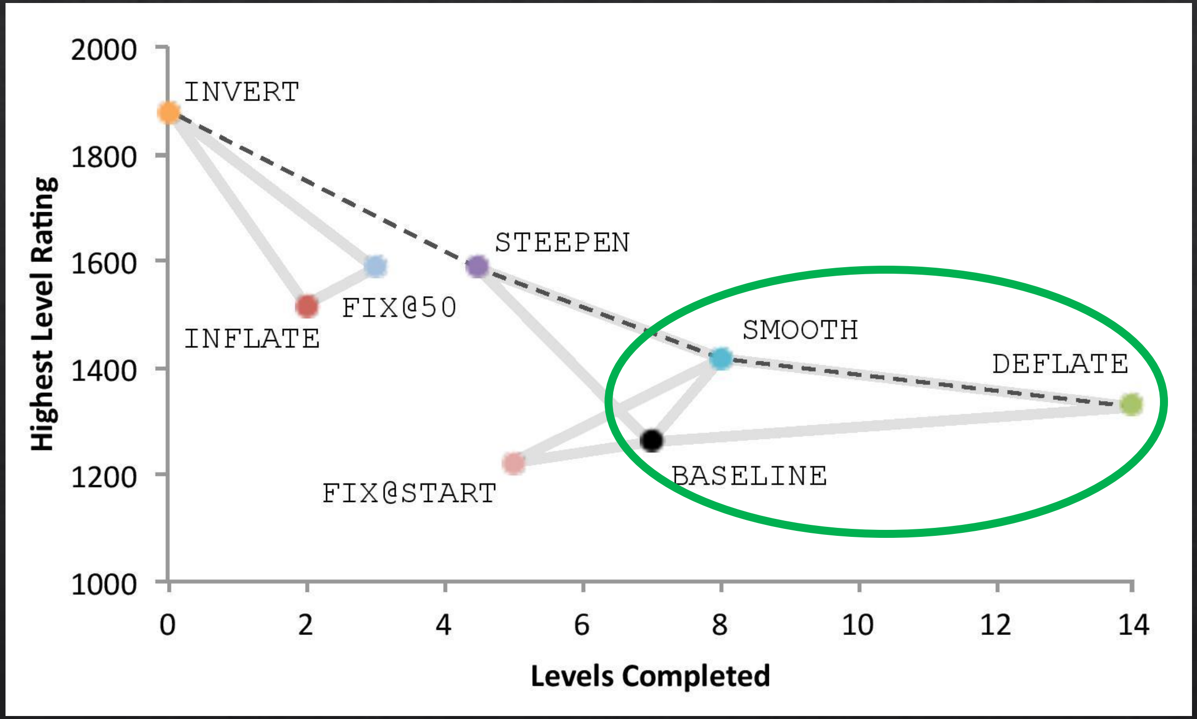
- HCGs try to maximize the number and/or quality of in-game tasks that players complete
  - Trade off between amount of work done *(Levels Completed)* and its quality (*Highest Level Rating)*

- Found the curves INVERT, STEEPEN, SMOOTH, DEFLATE to be Pareto efficient for *Levels Completed* and *Highest Level Rating*



Pareto Efficiency

- HCGs try to maximize the number and/or quality of in-game tasks that players complete
  - Trade off between amount of work done *(Levels Completed)* and its quality (*Highest Level Rating)*



- Found the curves INVERT, STEEPEN, SMOOTH, DEFLATE to be Pareto efficient for *Levels Completed* and *Highest Level Rating*

- Original BASELINE was outperformed by SMOOTH and DEFLATE suggesting that these might be better curves for *Paradox*

Conclusion

- A formal approach to transforming a game’s difficulty curve using function composition

Conclusion

- A formal approach to transforming a game's difficulty curve using function composition

- Modified curve of *Paradox* to generate new curves and precisely defined transformations

Conclusion

- A formal approach to transforming a game’s difficulty curve using function composition

- Modified curve of *Paradox* to generate new curves and precisely defined transformations

- Transformed curves impacted gameplay and some improved engagement

Conclusion

- A formal approach to transforming a game's difficulty curve using function composition
- Modified curve of *Paradox* to generate new curves and precisely defined transformations
- Transformed curves impacted gameplay and some improved engagement

Contact

Anurag Sarkar
Northeastern University
sarkar.an@husky.neu.edu

Acknowledgments

This material is based upon work supported by the **National Science Foundation** under grant no. 1652537. We would like to thank the **University of Washington**'s **Center for Game Science** for initial *Paradox* development.