Skill-Based Matchmaking for Competitive Two-Player Games

Supplemental Document

CEM YUKSEL, University of Utah and Roblox, USA

This supplemental document includes additional experiments with different parameters.

In the paper we present results with Elo [Elo 1978], Glicko [Glickman 1999], and TrueSkill [Herbrich et al. 2007] using some reasonable/default parameters. We also mention that the behavior of these rating estimation methods can be quite different with different parameters. Here, we present examples with alternative parameters, showing how the resulting rating estimation changes. We also include examples with different scaling factor parameters using our rating estimation method.

Alternative Parameters for Elo

In the paper, we use $K = 24$ for Elo, as recommended [Glickman 1995]. However, this is a rather small K-factor to use, since the maximum possible rating change $\Delta r_{\max} = 24$ (when the win probability was 0 or 1 and an upset happens, i.e. the winner is the player who was expected to lose) corresponds to at most 3.4% change in the win probability estimation, calculated as the change against an opponent with the player's previous rating, against whom the previous win probability would be 50%. Thus, the change in win probability estimation is evaluated using

$$f_\Delta(\Delta r) = \frac{1}{1 + 10^{-\Delta r/400}} - 50\% \ . \tag{1}$$

f_Δ is also used for converting rating estimation errors to win probability estimation errors.

Using $K = 24$, Elo in our tests in the paper can be too slow to keep up with the other methods, taking visibly longer to reach the vicinity of the player's rating.

Therefore, we use a higher K-factor of $K = 40$ in the examples here. This is still a reasonable parameter for Elo and the maximum rating change corresponds to at most 5.7% change in the win probability estimation. It makes Elo faster to respond, but noisier as a consequence.

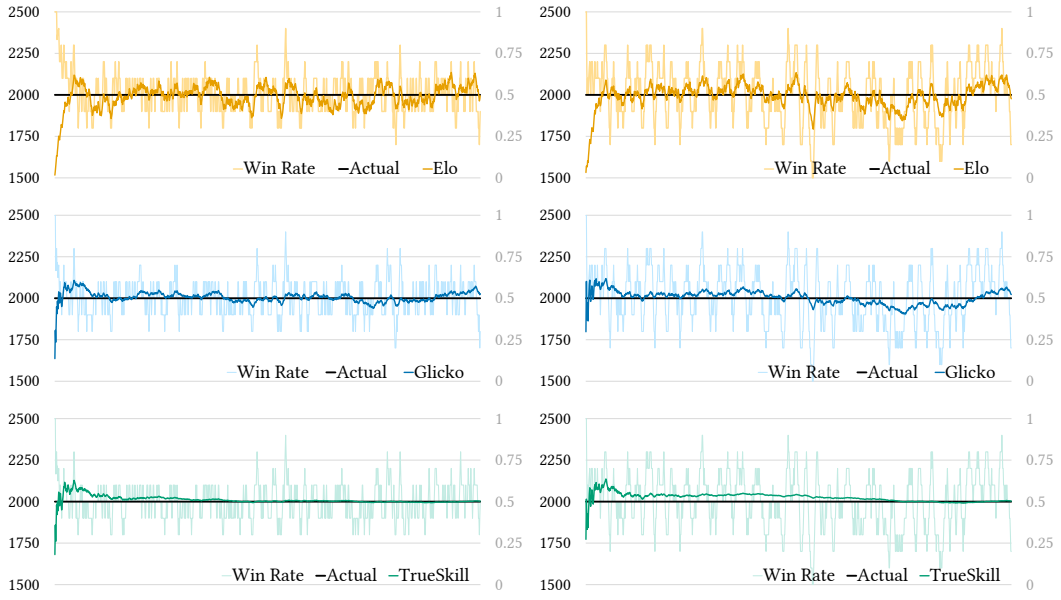
Alternative Parameters for Glicko

In the paper, we assume no time between games. This gives Glicko no chance to increase the rating deviations ϕ_a. As a result, the maximum possible rating change for Glicko keeps decreasing.

If we assume some time t between games, however, Glicko can increase the rating deviations before each game using $\phi'_a = \sqrt{\phi_a^2 + c^2 t}$, where c is a user-defined parameter. Glicko also bounds the maximum rating deviation to 350, its initial value. Note that this increase in the rating deviation is not tied to any change to the player's actual rating or game outcomes.

Author's address: Cem Yuksel, cem@cemyuksel.com, University of Utah, Roblox, USA.

2024. 2577-6193/2024/5-ART
https://doi.org/10.1145/3651303



(a) *Fixed rating with our matchmaking* **(b)** *Fixed rating with random matchmaking*

Fig. 1. *The estimated rating of a new player with a fixed actual rating throughout their first 1000 games (a) with our matchmaking and (b) without skill-based matchmaking. The lighter colors show the win rates of the past 10 games.*

Here, we allow Glicko to increase its rating deviations by using a constant factor of $c^2t = 6^2$. This alternative parameter shows how much more responsive Glicko can become to changes in the actual rating when we do not allow the rating deviations to fall below 6.

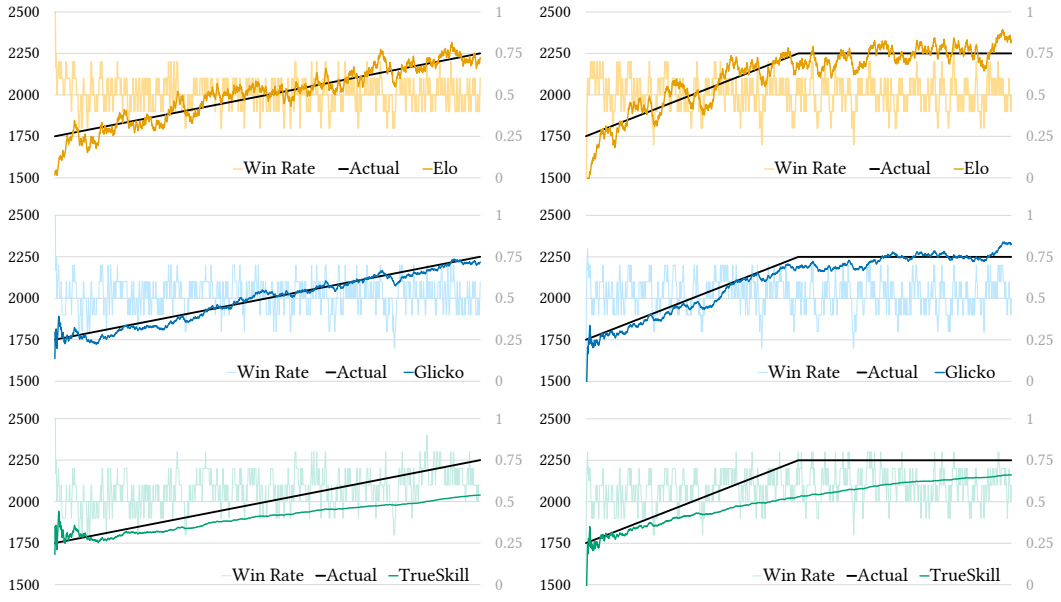
Alternative Parameters for TrueSkill

TrueSkill has a similar parameter γ to artificially increase the rating deviation before each game, using $\phi'_a = \sqrt{\phi_a^2 + \gamma^2}$. Unlike Glicko's formulation γ is not tied to time, but just like Glicko γ is not tied to any change in the player's actual rating either.

In the paper, we use $\gamma = 5$, which is close to the default value [Herbrich et al. 2007], after converting the performance variance of Elo using logistic distribution to the Gaussian distribution used by TrueSkill. Here, we use $\gamma = 0$ to show that, without the effects of this parameter, TrueSkill behaves similar to the results of Glicko in the paper.

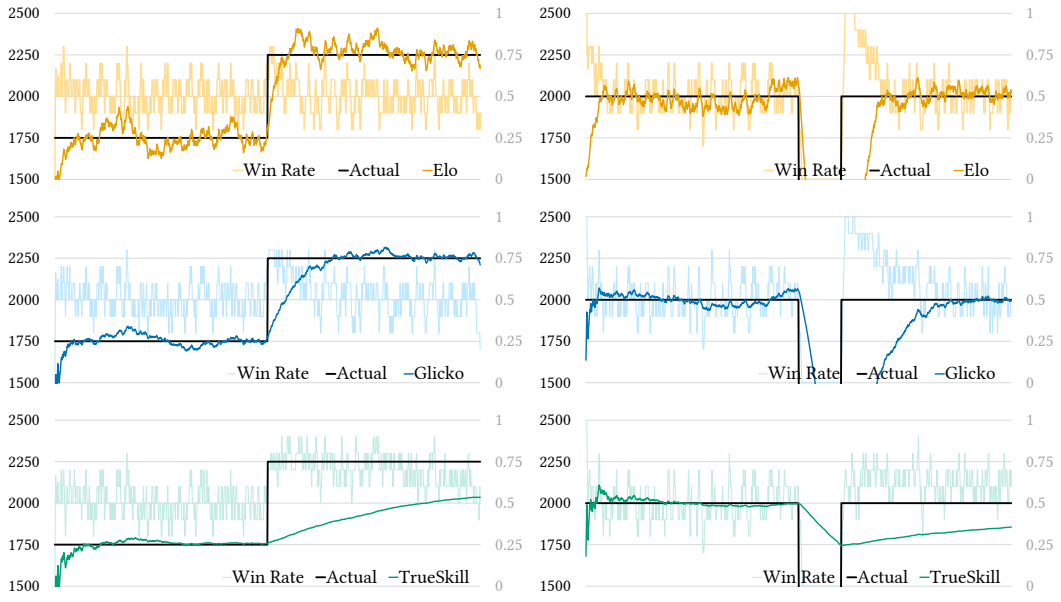
Estimating a Single Player's Rating

In Figures 1, 2 and 3 we repeat the experiments in corresponding figures of the paper for Elo, Glicko, and TrueSkill using the alternative parameters described above. Notice that in all of these experiments, Elo with $K = 40$ is noisier than the examples with $K = 24$ in the paper, but it responds faster to any change in the actual rating. Glicko no longer fails to respond properly to the changes in the actual rating, but behaves significantly noisier, as its rating deviation never reaches zero. TrueSkill with $\gamma = 0$ does an excellent job with a fixed actual rating, but fails to adjust the estimated



(a) *Increasing actual rating* **(b)** *First increasing, then fixed rating*

Fig. 2. *The estimated rating of a new player throughout their first 1000 games, comparing different rating estimation methods with our matchmaking for a window of 11 games: (a) steadily increasing actual rating and (b) an increasing actual rating that stops increasing half way through the games.*



(a) *A sudden jump in actual rating* **(b)** *A gap in actual rating*

Fig. 3. *The estimated rating of a new player throughout their first 1000 games: (a) a sudden increase in actual rating and (b) a gap in actual rating that is formed by the player intentionally throwing some games. The lighter colors show the win rates of the past 10 games.*

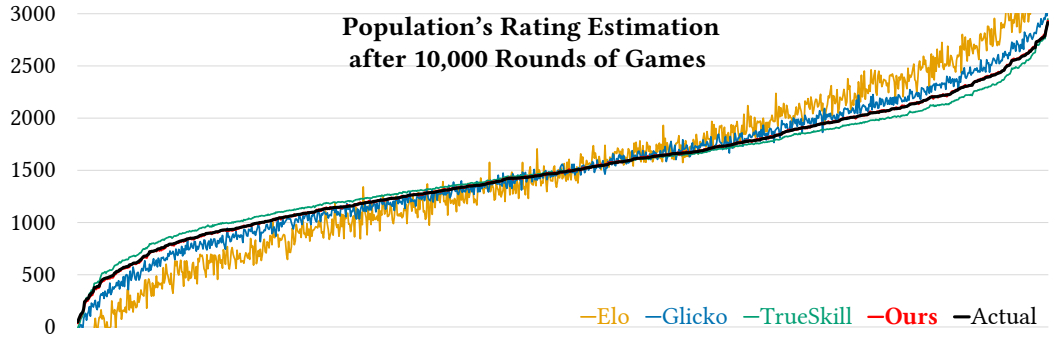


Fig. 4. *A population's rating estimation after ten thousand rounds of games with different rating estimation methods using the alternative parameters. The horizontal axis is the player population.*

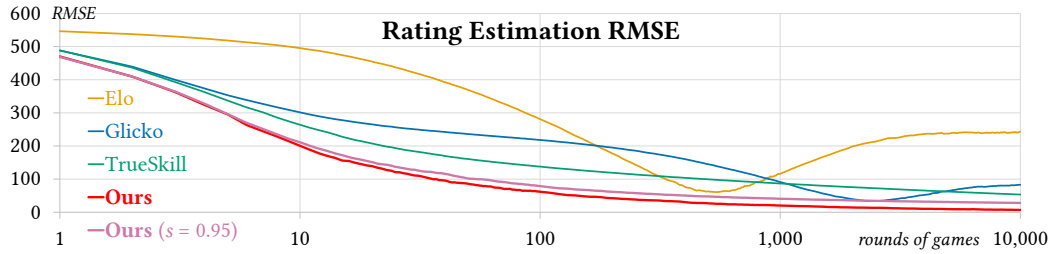


Fig. 5. *Root mean square error (RMSE) of different rating estimation methods for the same population in Figure 4 after each round of games on a log scale, using the alternative parameters.*

rating quickly enough when the actual rating changes, similar to the results in the paper for Glicko with no time between games. These results show that the behavior of these methods can be altered significantly by adjusting their parameters.

Estimating Population Ratings

The impact of the alternative parameters can be seen in our population test as well. Figure 4 shows the same experiment in the paper, but with the alternative parameters, after 10,000 rounds of games and Figure 5 shows the RMSE after each round. Notice that with $\gamma = 0$, the results of TrueSkill is similar to the results of Glicko in the paper (with no time between games), though closer to the actual ratings. On the other hand, this time Glicko begins to diverge after a while. Elo behaves similarly, but it moves faster, resulting in faster convergence at first and a more prominent divergence later.

To demonstrate how the parameters impact the behavior of these methods in this test, we provide the RMSE results with a range of parameter values for all methods in Figure 6. The default values used in the paper are highlighted as thicker curves. Notice that almost all parameters with Elo, Glicko, and TrueSkill lead to divergence after a while. The only exceptions are Glicko with $c^2 t = 0$ (the default parameter in the paper) and TrueSkill with $\gamma = 0$ (the alternative parameter). With those special parameters, however, they not only converge slower for this population, but also fail to respond fast enough when the player's actual rating changes, as demonstrated by our examples for the estimation of a single player's rating.

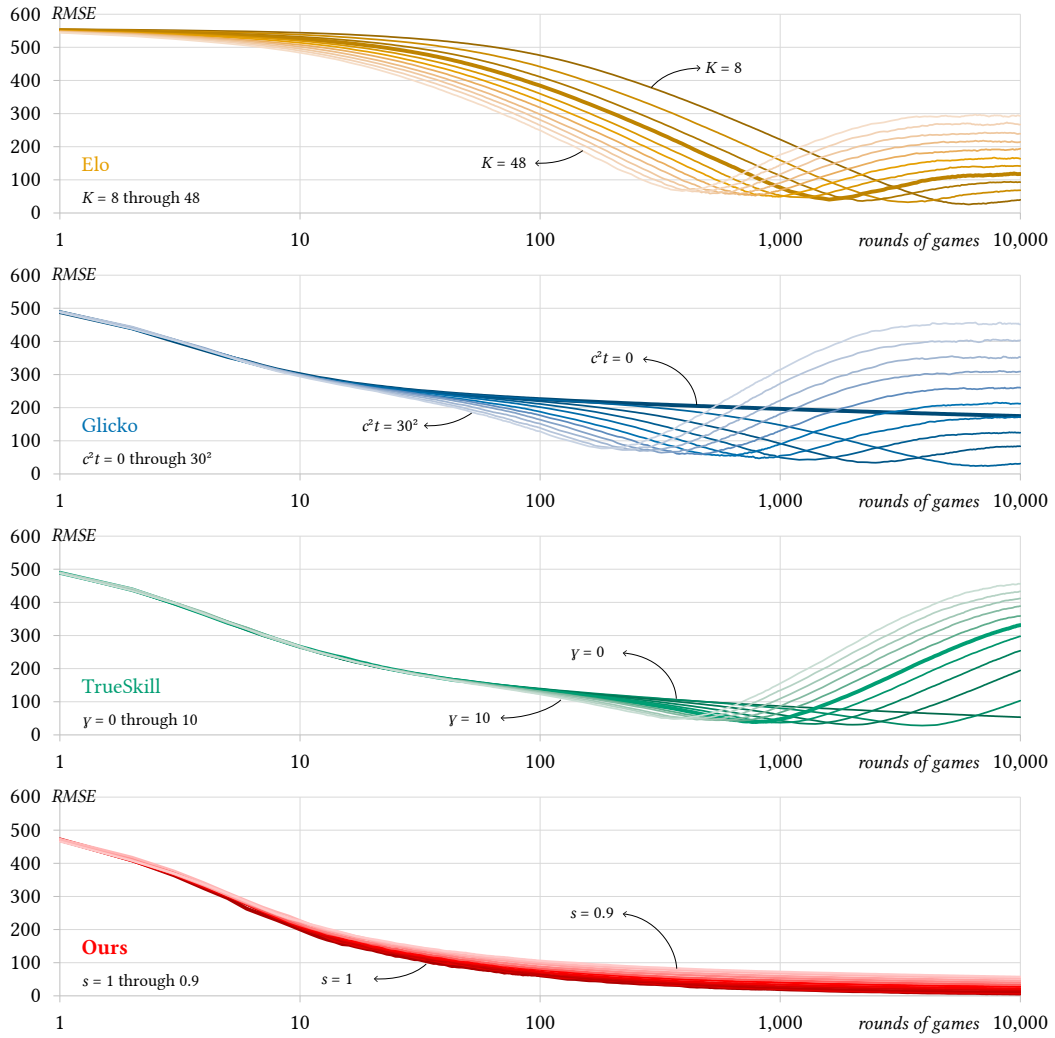


Fig. 6. *Root mean square error (RMSE) of rating estimation for a range of parameter values.*

Figure 6 also includes the RMSE results using a range of scaling factor parameters $s \in [0.9, 1]$ with our method. Notice that the s values in this range have a relatively small impact on the results. Even with $s = 0.9$, a highly safe parameter for boosting stability, our method still achieves a desirable convergence rate in this example.

We also include the RMSE results with our method using the same parameter range for the version of our test that does not include skill-based matchmaking in Figure 7. As demonstrated in the paper, $s = 1$ leads to instabilities in this case. Using a smaller value for s quickly overcomes this instability.

Yet, having no skill-based matchmaking is an extreme case for a method designed for skill-based matchmaking. In Figure 8, we show that even with relatively poor skill-based matchmaking, where the sorted list is shuffled within a range of 30% of the population, our rating estimation method remains stable with $s = 1$, as well as $s < 1$.

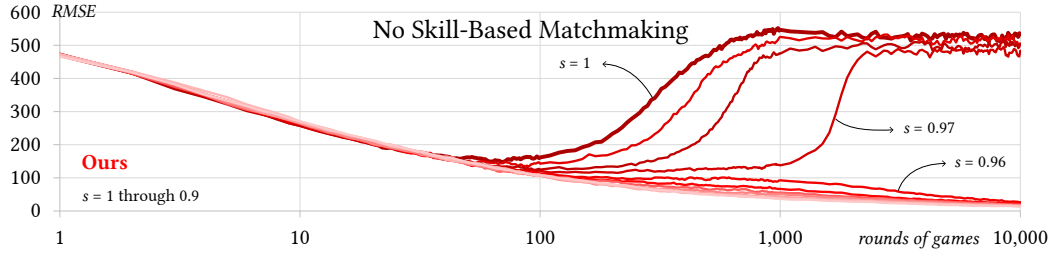


Fig. 7. *Root mean square error (RMSE) of rating estimation for our test with no skill-based matchmaking using different scaling factor parameters with our method.*

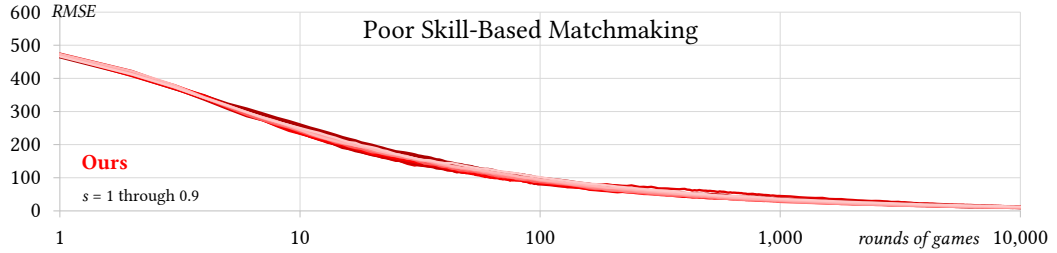


Fig. 8. *Root mean square error (RMSE) of rating estimation using different scaling factor parameters with our method for highly poor skill-based matchmaking, achieved by shuffling the sorted list of players within a window of 30% of the population, resulting in a high probability of poorly matched player pairs with significantly different ratings.*

REFERENCES

Arpad Elo. 1978. *The Rating of Chess Players Past and Present.* Arco.

Mark Glickman. 1995. A comprehensive guide to chess ratings. *Chess Journal* 3 (1995), 59–102.

Mark E. Glickman. 1999. Parameter estimation in large dynamic paired comparison experiments. *Journal of Applied Statistics* 48 (1999), 377–394.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. TrueSkill™: A Bayesian Skill Rating System. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference.* https://doi.org/10.7551/mitpress/7503.003.0076