Bukovinian Math. Journal. 2025, **13**, 2, 137–151 Буковинський матем. журнал 2025, Т.13, №2, 137–151

DOI: https://doi.org/10.31861/bmj2025.02.14

Taras Andrytsuliak, Sergiy Martyniuk

GENERALIZATION OF THE ELO SYSTEM FOR INTERACTIONS BETWEEN MANY AGENTS

The paper proposes a generalization of the classical Elo rating system for multi-player environments and employs optimization techniques to identify the model's optimal configuration. The proposed approach represents multi-agent interactions as a set of pairwise comparisons and formulates the rating estimation problem as the minimization of a logistic loss function. This formulation enables the derivation of analytical expressions for rating updates based on gradients used to search for optimal hyperparameters, resulting in smoother and more stable learning dynamics compared to the classical Elo system.

An additional adaptive update coefficient is introduced, which depends on the number of interactions and the number of agents involved in each event. Such normalization helps prevent excessive rating drift in large datasets and mitigates overly dampened updates when the amount of data or the number of agents is small. The paper also examines principles for initializing the ratings of new agents and for dynamically adjusting the learning rate based on the accumulated information for each agent, allowing the system to converge more rapidly to accurate rating levels and improving overall stability and interpretability.

Gradient descent is employed to search for the optimal hyperparameter values by minimizing the loss function and automatically selecting appropriate parameter settings. This enables the system to produce more accurate rating estimates and, consequently, achieve better predictions of future interactions.

The proposed system easily adapts to environments with multi-party interactions while remaining fully compatible with binary-comparison scenarios. When necessary, it can be extended with additional hyperparameters to account for the specifics of a given domain.

The model illustrates the advantages of combining classical rating methodology with modern optimization techniques and is applicable to sports, gaming, and educational systems where accuracy, adaptability, and interpretability are essential.

Key words and phrases: gradient descent, hyperparameter, elo ratings, probabilistic forecasting, statistical modelling, time series, machine learning.

Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine
e-mail: *s.martyniuk@chnu.edu.ua (Sergiy Martyniuk), andrytsuliak.taras@chnu.edu.ua (Taras Andrytsuliak)*

УДК 519.8
2010 *Mathematics Subject Classification:* 68T05, 62F07.

INTRODUCTION

In modern gaming, sports, and educational environments, there is a growing need for fair, adaptive, and interpretable methods of ranking and evaluating interactions between agents (players) within a closed environment. One of the most popular models, known for its simplicity, transparency, and high effectiveness, is the Elo system, proposed by Arpad Elo in the mid-20th century[4]. Although its original application was related to chess, today the Elo system is successfully used in a wide variety of fields, from other sports (tennis[2], football[1, 7], board games) to online games, machine learning and even the evaluation of text or translation quality.

The essence of the classical Elo system approach lies in assigning each agent a numerical value as a rating that reflects their probability of defeating an opponent. After each interaction, the rating is adjusted based on the actual and expected outcomes of the interaction:

$$r_{i(k+1)} = r_{ik} + K\left(S_{ik} - P_{ik}\right)$$

where $r_{i(k+1)}$ is the agent rating i after the k-th interaction, r_{ik} is the agent rating i before the k-th interaction, S_{ik} is the actual outcome of the interaction (for example, 1 for a win, 0 for a loss, 0.5 for a draw), P_{ik} is the expected outcome of the interaction, and K is the adjustment coefficient that determines the speed of rating updates.

At the same time, the expected outcome of the k-th interaction between agents i and j, with corresponding pre-interaction ratings r_{ik} and r_{jk}, is calculated as:

$$P_{ik} = \frac{1}{1 + 10^{\left(\frac{r_{jk} - r_{ik}}{400}\right)}}$$

Consequently, if a higher-rated agent defeats a lower–rated one, their rating increases only slightly, while in the case of a loss, it decreases significantly. This approach allows the ratings to quickly adapt to the underlying skill level of the agents and is also intuitive for the users.

Among the key advantages of the classical Elo system are its computational simplicity, the absence of complex hyperparameters, and the lack of a need for historical data for each agent (the rating is formed solely based on the previous value and the most recent interaction outcome). However, the vast majority of specific applications of the Elo system require introducing additional parameters and modifying the classical version. These modifications include:

- Multifactor models that account for additional interaction characteristics such as home advantage in football or court surface type in tennis.
- Models with sub-environments that make it possible to capture relationships between different environments with identical or distinct hierarchies, such as different leagues.
- Elo-based models for simultaneous interaction among many agents. For example, in multiplayer online games, dozens of participants may be involved, and it is necessary to determine rating changes for each of them based on the collective outcome.
- Bayesian approaches, which treat the rating not as a point estimate but as a probability distribution, allowing uncertainty to be taken into account.

Related systems inspired by Elo are also widely used, such as Glicko[6], which additionally incorporates the variance (volatility) of a agent's skill, allowing faster adaptation to changes in performance. A Bradley-Terry model[3], which originated earlier, has a similar logic to estimate the "probability of victory", but formally expresses it through logistic or normal distributions and is typically used in the context of statistical analysis of pairwise comparisons, as Weng and Lin did[11].

At the same time, there exist variations of the Elo system designed for comparisons among many agents. In particular, Powell[9] considers the interaction of N agents as a set of $N-1$ separate sub-interactions, where after each sub-interaction the agent with the worst result is eliminated (endurance model). An alternative variant is one in which the strongest agent is eliminated after sub-interaction, and the next sub-interaction is then considered without them (speed model). The probability of obtaining the worst result in the first sub-interaction is therefore:

$$P_{ik1 \text{ eliminated}} = \frac{e^{-r_{ik}}}{\sum_{x \in N} e^{-r_{xk}}}$$

Let agent j take the last place and be eliminated. Then the probability of being the worst in the next interaction among the remaining agents is:

$$P_{ik2 \text{ eliminated}} = \frac{e^{-r_{ik}}}{\sum_{x \in N \setminus \{j\}} e^{-r_{xk}}}$$

And so on for each subsequent sub-interaction until only the winner remains. The rating after the interaction is updated as follows:

$$r_{i(k+1)} = r_{ik} + \sum_{t=1}^{N-1} KB_{ikt} \left(S_{ikt} - P_{ikt}\right)$$

where B_{ikt} is a boolean indicator showing whether agent i participated in the sub-interaction t of the interaction k.

This approach works well for modeling interaction outcomes, as shown by Powell, but it may slightly overestimate ratings for stronger agents, since agents who rank lower will participate in fewer sub-interactions than those who rank higher.

The Elo system was originally constructed to model and forecast outcomes of strictly pairwise comparisons, but subsequent research has explored ways to generalize it to multi-agent interactions such as group competitions and races. For example, an approach proposed by Moore et al.[8], treats an interaction with n agents as an equivalent collection of $\frac{n(n-1)}{2}$ independent binary duels, where each duel is awarded to the competitor who finishes ahead of the other. Powell [9] argues that this formulation introduces a systematic upward bias in the resulting rating estimates, particularly in scenarios involving a large number of interactions or a high density of agents per interaction. Although Powell's observation is well-founded, the underlying issue can be mitigated through appropriate methodological adjustments, and addressing this limitation constitutes one of the objectives of this study.

Almost all of the methods mentioned rely on certain fixed hyperparameters (for example, the learning coefficient in the classical Elo system), which are often chosen manually or based on empirical considerations. This creates a problem as an incorrect choice of hyperparameters may lead to an overly inert or an overly unstable rating system, distorting the underlying skill levels of the agents. This problem becomes especially acute in complex scenarios with many agents, where interactions are significantly more complex than in ordinary pairwise matches.

To address this issue in machine learning problems, scientific and practical studies propose various approaches to the automatic tuning of hyperparameters. Among these methods are optimization-based techniques, including gradient descent[5]. As shown for football games [1], gradient-based optimization can effectively tune the hyperparameters of an Elo-style rating system, allowing one to find the most suitable hyperparameter configuration by optimizing the logistic loss function[10].

In the context of the Elo system, gradient descent allows not only the automatic selection of an optimal update coefficient K, but also the calibration of other hyperparameters such as the shape of the win-probability function, or even the initial agent ratings if sufficient historical data are available. Applying gradient descent to optimize Elo parameters on historical data makes it possible to create a more accurate and stable rating system adapted to a specific game or league.

Thus, combining the Elo system with modern optimization techniques, such as gradient descent, creates an opportunity to develop a generalized rating framework that preserves the conceptual clarity and efficiency of classical pairwise Elo while extending it to settings involving multi-agent interactions. The goal of this study is to construct such a system: one that not only supports pairwise comparisons but also correctly models outcomes of interactions among multiple agents, addresses structural biases that arise when pairwise simplifications are applied to multi-agent scenarios, and leverages gradient-based optimization to identify an appropriate set of hyperparameters from historical interactions data.

This work aims to provide a principled, scalable, and data-driven generalization of Elo suitable for high-dimensional competitive environments.

1 Methodology

1.1 General principles

In our approach for evaluating agents in multi-agent interactions, each interaction is decomposed into a set of pairwise sub-interactions. Each pair represents a separate evaluation unit in which one agent ranks higher or lower relative to the other. To train the rating system based on these sub-interactions, we formulate the problem as an optimization problem.

Specifically, we aim to find agent ratings that maximize the agreement between the expected and actual outcomes of the sub-interactions. For this purpose, the logistic loss function (log-loss) was used as the loss function, which is a standard choice for binary classification problems, where the outcome variable takes values 0, 0.5, or 1 (win, draw, or loss in the pair).

For each sub-interaction between agents i and j in interaction k defined:

$S_{ijk} \in \{0, 0.5, 1\}$ - actual outcome of the sub-interaction: 1 if agent i ranks higher than agent j, 0 if lower, and 0.5 if equal;

$$P_{ijk} = \frac{1}{1 + 10^{\frac{r_{jk} - r_{ik}}{400}}}$$ - expected probability of agent i winning over agent j, computed from the current ratings.

Then, the logistic loss function for a single pair is given by:

$$l_{ijk} = -\big(S_{ijk} \cdot \ln P_{ijk} + (1 - S_{ijk}) \cdot \ln(1 - P_{ijk})\big)$$

This function penalizes large deviations between the prediction P_{ijk} and the actual outcome S_{ijk}: it is minimal when the prediction is close to the true value and grows rapidly for large errors. This property makes the log-loss function convenient for training a rating model on historical data.

To minimize the loss function, gradient descent was applied. It is an iterative optimization procedure that updates the model parameters (in our case, agent ratings) in the direction opposite to the gradient of the loss function. Intuitively, the gradient indicates how the loss would change if the rating is adjusted, and thus shows how to correct the rating to reduce the error.

Since l_{ijk} is a complex function through P_{ijk}, the chain rule was applied:

$$\frac{\partial l_{ijk}}{\partial r_{ik}} = \frac{\partial l_{ijk}}{\partial P_{ijk}} \cdot \frac{\partial P_{ijk}}{\partial r_{ik}} \tag{1}$$

First, the derivative with respect to P_{ijk} is computed:

$$\begin{aligned} \frac{\partial l_{ijk}}{\partial P_{ijk}} &= -S_{ijk} \cdot (\ln P_{ijk})' - (1 - S_{ijk}) \cdot (\ln(1 - P_{ijk}))' \\ &= -\frac{S_{ijk}}{P_{ijk}} + \frac{1 - S_{ijk}}{1 - P_{ijk}} = \frac{P_{ijk} - S_{ijk}}{P_{ijk} \cdot (1 - P_{ijk})} \end{aligned} \tag{2}$$

Next, compute the derivative of P_{ijk} with respect to r_{ik} is computed:

$$\begin{aligned} \frac{\partial P_{ijk}}{\partial r_{ik}} &= \frac{\partial}{\partial r_{ik}} \left(\frac{1}{1 + 10^{\beta(r_{jk} - r_{ik})}} \right) \\ &= -\frac{1}{(1 + 10^{\beta(r_{jk} - r_{ik})})^2} \cdot \frac{\partial}{\partial r_{ik}} \left(1 + 10^{\beta(r_{jk} - r_{ik})} \right) \end{aligned}$$

where $\beta = \frac{\ln(10)}{400}$. Noting that

$$\frac{\partial}{\partial r_{ik}} \left(1 + 10^{\beta(r_{jk} - r_{ik})} \right) = -\beta\, 10^{\beta(r_{jk} - r_{ik})},$$

it follows that

$$\begin{aligned} \frac{\partial P_{ijk}}{\partial r_{ik}} &= \frac{\beta\, 10^{\beta(r_{jk} - r_{ik})}}{(1 + 10^{\beta(r_{jk} - r_{ik})})^2} \\ &= \beta \cdot \frac{1}{1 + 10^{\beta(r_{jk} - r_{ik})}} \cdot \frac{(1 + 10^{\beta(r_{jk} - r_{ik})}) - 1}{1 + 10^{\beta(r_{jk} - r_{ik})}} \end{aligned}$$

Substituting P_{ijk} in place of $1/(1 + 10^{\beta(r_{jk} - r_{ik})})$ yields:

$$\frac{\partial P_{ijk}}{\partial r_{ik}} = \beta\, P_{ijk}\, (1 - P_{ijk}) \tag{3}$$

Finally, substituting (2) and (3) into (1) yields:

$$\frac{\partial l_{ijk}}{\partial r_{ik}} = \frac{P_{ijk} - S_{ijk}}{P_{ijk} \cdot (1 - P_{ijk})} \cdot \beta\, P_{ijk}\, (1 - P_{ijk}) = \beta\, (P_{ijk} - S_{ijk}) \tag{4}$$

According to the gradient descent method, the rating update for a single sub-interaction is:

$$r_{ijk} = r_{ik} - \eta\, \frac{\partial l_{ijk}}{\partial r_{ik}}$$

Taking all sub-interactions into account yields:

$$r_{ik} = r_{ik} - \sum_{\substack{j=1 \\ j \neq i}}^{N} \eta \frac{\partial l_{ijk}}{\partial r_{ik}} \tag{5}$$

Substituting the derivative in (4) by (5) the multi-agent update formula becomes:

$$r_{i(k+1)} = r_{ik} - \eta \beta \sum_{\substack{j=1 \\ j \neq i}}^{N} (P_{ijk} - S_{ijk}) = r_{ik} - K \sum_{\substack{j=1 \\ j \neq i}}^{N} (P_{ijk} - S_{ijk}) \tag{6}$$

This provides a smooth and consistent rating update that combines signals from all pairwise comparisons. The formula generalizes the classical Elo update to the multi-agent case, based on a rigorous mathematical derivation via the loss gradient.

Let G_k denote the set of agents participating in the interaction k, and let there be K interactions in total. The total loss function is the sum of all logistic losses for pairs (i, j) arising from the pairwise decomposition of each interaction:

$$L = \sum_{k=1}^{K} \sum_{\substack{i,j \in G_k \\ i<j}} l_{ijk} \tag{7}$$

1.2 Additional factors in the Elo rating model

In general, the outcome of interactions between agents may depend not only on their ratings but also on additional factors. These factors should be taken into account when calculating the expected outcome of the sub-interaction of agent i with agent j within the interaction k. Therefore, this value is given by:

$$P_{ijk} = \frac{1}{1 + 10^{f'\left(r_{jk}, r_{ik}, g_1, g_2, \ldots, g_L\right)/400}} = f(r_{jk}, r_{ik}, g_1, g_2, \ldots, g_L) \tag{8}$$

where $f(r_{jk}, r_{ik}, g_1, g_2, \ldots, g_L)$ is a function representing the mathematical expectation of the outcome of the sub-interaction between agent i and agent j. The importance of a particular interaction or sub-interaction may differ from others; therefore, the rating update function in the general case is given by:

$$r_{i(k+1)} = r_{ik} + f(h_1, h_2, \ldots, h_L) \sum_{\substack{j=1 \\ j \neq i}}^{N} (S_{ijk} - P_{ijk}) \tag{9}$$

where $f(h_1, h_2, \ldots, h_L)$ is a function that characterizes the influence of additional factors on the importance of the event.

Thus, substituting (8) into (9) and generalizing yields:

$$r_{i(k+1)} = f\left(r_{1k}, \ldots, r_{ik}, \ldots, r_{Nk},\ S_{i1k}, \ldots, S_{iNk},\ g_1, g_2, \ldots, g_L,\ h_1, h_2, \ldots, h_L\right)$$

Taking into account that $R_k = (r_{1k}, \ldots, r_{ik}, \ldots, r_{jk}, \ldots, r_{Nk})$ represents the ratings of all agents before interaction k, and $(S_{i1k}, \ldots, S_{iNk}) = S_k \in S$ denotes the outcome of interaction k, the rating of agent i after interaction k is a function of the following parameters:

$$r_{i(k+1)} = f(R_k, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L)$$

Accordingly:

$$r_{j(k+1)} = f(R_k, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L)$$

Thus, the state vector of all agents' ratings after interaction k depends on the same parameters as in [1], which yields:

$$R_{k+1} = f(R_k, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L) \tag{10}$$

The state vector of agent ratings after interaction $k+1$ is given by:

$$R_{k+2} = f(R_{k+1}, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L) \tag{11}$$

Substituting (10) into ((11) yields:

$$R_{k+2} = f(R_k, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L)$$

Thus, the ratings of the agents at any moment in time can be expressed through the initial agent ratings, the interaction vector, and the parameters of the Elo system:

$$R_k = f(R_1, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L)$$

The rating matrix of the agents $R = \{R_1, \ldots, R_k, \ldots, R_{m+1}\}$, which contains the ratings of all agents at all time moments, can be expressed as follows:

$$R = f(R_1, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L) \tag{12}$$

If the initial ratings of all agents are the same and equal to a fixed value r_0, then the expression simplifies to:

$$R = f(r_0, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L)$$

Let P be the set of all win probabilities in all pairwise interactions. Then, from (7), it follows that:

$$L = f(P, S) \tag{13}$$

From (8), it follows that:

$$P = f(R, g_1, g_2, \ldots, g_L) \tag{14}$$

Substituting (12) into (14), and then substituting the resulting expression into (13), yields:

$$L = f(R_1, S, g_1, g_2, \ldots, g_L, h_1, h_2, \ldots, h_L) \tag{15}$$

Thus, as in the basic case for pairwise interactions[1], the value of the logistic loss function is determined by the initial ratings of the agents, the outcomes of their interactions, and the factors that influence both the probability of the expected outcome and the weight of each event.

1.3 Adjustment of correction coefficient K

As shown in (6), the rating update in a single sub-interaction is given by:

$$\Delta r \approx K \cdot (S - P)$$

In a multi-agent interaction, agent i participates in $N-1$ pairwise sub-interactions (with each opponent), and over the course of a year, there are M such interactions. Therefore, the total annual drift scales as:

$$\text{drift}_i \approx K \cdot \sum_{k=1}^{M} \sum_{\substack{j=1 \\ j\neq i}}^{N} (S_{ijk} - P_{ijk}) \approx K \cdot M \cdot (N-1) \cdot \overline{\varepsilon(S-P)}$$

where $\overline{\varepsilon(S-P)}$ is the average prediction error defined as the difference between the actual and expected outcome.

If a constant $K = K_0$ is used, then drift_i grows linearly with M and $N-1$. For large values of M and N, the rating may change excessively leading to overfitting, while for small values it barely changes (underfitting).

If the drift is fully normalized by dividing by M and $N-1$, the 'data-scale' effect is completely compensated for, but another problem arises. For $N = 2$ and $M = 1$, it follows that $K = K_0$, and the one-time increase after a win is:

$$\Delta r_{\text{low interaction}} = K_0 \cdot (1-P)$$

For the case with many agents and interactions (wins in each sub-interaction), with

$$K = \frac{K_0}{M \cdot (N-1)},$$

the total annual change is:

$$\begin{aligned}
\Delta r_{\text{high interaction}} &= \frac{K_0}{M \cdot (N-1)} \sum_{k=1}^{M} \sum_{\substack{j=1 \\ j\neq i}}^{N} (1 - P_{ijk}) \\
&= \frac{K_0}{M} \sum_{k=1}^{M} \left[\frac{1}{N-1} \sum_{\substack{j=1 \\ j\neq i}}^{N} (1 - P_{ijk}) \right] \\
&= \frac{K_0}{M} \sum_{k=1}^{M} \overline{\varepsilon(1-P_k)} \\
&= \frac{K_0}{M} \cdot M \cdot \overline{\varepsilon(1-P)} \\
&= K_0 \cdot \overline{\varepsilon(1-P)}
\end{aligned}$$

Since under constant wins the probability P increases with each interaction, the average value $\overline{\varepsilon(1-P)}$ in many-interaction scenarios will be smaller than $(1-P)$ in a single interaction, assuming initially equal ratings. Therefore,

$$\Delta r_{\text{low interaction}} \geq \Delta r_{\text{high interaction}},$$

which indicates excessive smoothing.

Hence, complete lack of normalization (M^0 and $(N-1)^0$) leads to excessive drift for large M and N, while full normalization (M^1 and $(N-1)^1$) results in over-smoothing at large scales and volatility in "small" scenarios (because there $K \approx K_0$). Balance is achieved through partial normalization, where the exponents for M ($\text{degree}_{\text{interactions}}$) and for $N-1$ ($\text{degree}_{\text{agents}}$) lie in the interval $(0,1)$. This partially compensates for the scale of M and N but does not "stifle" learning as the data grows.

Thus, the adaptive update coefficient is:

$$K(M,N) = \frac{K_0}{M^{\text{degree}_{\text{interactions}}} \cdot (N-1)^{\text{degree}_{\text{agents}}}} \tag{16}$$

It makes the expected annual drift approximately proportional to:

$$\Delta r \approx K_0 \cdot M^{1-\text{degree}_{\text{interactions}}} \cdot (N-1)^{1-\text{degree}_{\text{agents}}} \cdot \overline{\varepsilon(1-P)},$$

and due to the exponents in $(0,1)$, this allows the model to retain adaptability for a large number of interactions while reducing volatility in cases with few data points.

1.4 New agents initial rating

In systems with a dynamic environment, new agents continuously enter the system and initially have no interaction history. The choice of its initial rating is crucial:

- a rating that is too high creates inflated expectations, and the agent's rating will drop sharply after the first losses;
- a rating that is too low leads to underestimated expectations and excessively fast growth in the case of a few victories;
- in the long run, both distortions skew the dynamics of the overall system and complicate interpretation.

The basic and simplest approach is to assign newcomers a fixed initial rating r_0. A more sophisticated option is to use an adaptive value r_{0k}, which is determined based on the current distribution of ratings at the moment the agent appears. This approach is meaningful when the environment consists of several isolated subsystems with different average rating levels. However, such scenarios are rather exceptional for multi-agent interaction. Therefore, it is more reasonable to use the simpler variant with a fixed r_0.

Moreover, as will be shown below, it is possible to periodically normalize agents' ratings to improve system stability, and thus in such a case the basic approach effectively yields the same result as the adaptive one.

1.5 Impact of interaction number on correction coefficient

New agents entering the system initially have no interactions, so their rating is estimated with high uncertainty. With each subsequent interaction, this uncertainty decreases. The amount of Fisher information after one sub-interaction is proportional to:

$$I(r) = \mathbb{E}\left[\left(\frac{\partial}{\partial r}\log L(r)\right)^2\right]$$

Given that, as shown in (4),

$$\frac{\partial}{\partial r}\log L(r) = \beta \cdot (P - S),$$

it follows that:

$$I(r) = \mathbb{E}\big[(\beta \cdot (P - S))^2\big] \approx \beta^2 \cdot \mathbb{E}\big[P(P-1)^2 + (1-P)P^2\big] = \beta^2 \cdot P(1-P)$$

After T interactions, the cumulative information grows proportionally to T, and the uncertainty decreases approximately as $\sigma_T^2 \propto 1/T$. Thus, it is natural to connect the optimal step-size multiplier of the update coefficient with this uncertainty so that the multiplier decreases with each new interaction. To prevent an excessively rapid slowdown, a saturation mechanism is introduced: after $T \geq T_{\text{sat}}$, the update coefficient becomes constant.

As shown in (16), the correction coefficient is:

$$K(M, N) = K_0 \cdot M^{-\text{degree}_{\text{interactions}}} \cdot (N-1)^{-\text{degree}_{\text{agents}}},$$

so after saturation it must take this value. Before saturation, the coefficient gradually decreases, approaching a plateau that reflects the diminishing uncertainty.

Taking this into account, the following multiplier depending on the number of interactions is defined:

$$g(T) = 1 + \alpha \left(1 - \min\left\{1, \frac{T}{T_{\text{sat}}}\right\}\right)^2$$

where $\alpha > 0$ amplifies the initial update and the exponent 2 enforces a quadratic decay. Thus:

$$T = 0 \Rightarrow g(T) = 1 + \alpha,$$

$$0 < T < T_{\text{sat}} \Rightarrow 1 < g(T) < 1 + \alpha,$$

$$T \geq T_{\text{sat}} \Rightarrow g(T) = 1.$$

Therefore, the overall update coefficient is:

$$K(M, N, T) = K(M, N) \cdot g(T)$$

The resulting rating update formula becomes:

$$r_{ijk} = r_{ik} + K(M, N, T) \cdot \sum_{j=1, j \neq i}^{N} (S_{ijk} - P_{ijk})$$

$$K(M, N, T) = \frac{K_0 \cdot \left(1 + \alpha \left(1 - \min\left\{1, \frac{T}{T_{\text{sat}}}\right\}\right)^2\right)}{M^{\text{degree}_{\text{interactions}}} \cdot (N-1)^{\text{degree}_{\text{agents}}}}$$

Thus, introducing the dependence of the update coefficient on the interaction count ensures rapid adaptation for new agents during the initial stages and gradual stabilization as more observations accumulate.

1.6 Gradient descent hyperparameter optimization

The search for optimal Elo system parameters is performed by minimizing the logistic loss function, which reflects the discrepancy between the expected and actual outcomes of agent interactions. In this way, the model adapts its parameters to reduce the prediction error.

As shown in (15), the value of the logistic loss function depends on the agents' initial ratings, the outcomes of their interactions, the factors affecting the expected outcome probabilities, and the factors determining the importance of each event. In the general implementation of the Elo rating, such factors include the initial agent ratings R_1, the base correction coefficient K_0, the exponent accounting for the number of interactions in a period $\text{degree}_{\text{interactions}}$, the exponent accounting for the total number of agents in an interaction $\text{degree}_{\text{agents}}$, the regulator of maximum early-stage rating growth α, and the saturation threshold T_{sat}. Thus:

$$L = \sum_{k=1}^{K} \sum_{\substack{i,j \in G_k \\ i<j}} l_{ijk} = \sum_{k=1}^{K} \sum_{\substack{i,j \in G_k \\ i<j}} \left(S_{ijk} \ln P_{ijk} + (1 - S_{ijk}) \ln(1 - P_{ijk})\right)$$
$$= f\left(S, R_1, K_0, \text{degree}_{\text{interactions}}, \text{degree}_{\text{agents}}, \alpha, T_{\text{sat}}\right)$$

It follows that the optimal hyperparameter values, corresponding to the minimum of the loss function, can be found using gradient descent. In the classical formulation, the parameters are updated after each gradient descent iteration using a fixed step size:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla f(\theta^{(t)})$$

where η is the learning rate that controls the speed of parameter updates, and $\theta^{(t)}$ is the set of parameters. However, since different parameters have different orders of magnitude (for example, $\text{degree}_{\text{interactions}}$ and $\text{degree}_{\text{agents}}$ are expected to lie between 0 and 1, while R_1 and K_0 may be in the hundreds or thousands), the classical approach leads to slow updates of large-scale parameters. To improve convergence for parameters with small values, a modified approach is more appropriate:

$$\theta^{(t+1)} = \theta^{(t)} - \eta\, \theta^{(t)} \nabla f(\theta^{(t)})$$

This modification implements proportional parameter updates relative to their current values, helping to avoid convergence imbalance between parameters of different scales.

In the subsequent section of this work, this approach is applied to optimize the Elo system parameters on historical data, which improves model accuracy and the stability of algorithm convergence.

2 Application

2.1 Multi-agent interaction based on Formula One races

To illustrate the practical application of the proposed model, data from the Formula 1 World Championship were used. For modeling purposes, the dataset is divided into initialization, training, and validation subsets. For each individual interaction (Grand Prix), only drivers who completed the race are considered. This restriction is necessary because the Elo rating system is designed to quantify the inherent skill or strength of an agent, whereas the probability of finishing a race is influenced predominantly by exogenous factors, such as vehicle reliability, stochastic events, and interactions with competitors that may result in retirements. Consequently, modeling the probability of finishing requires a distinct algorithmic approach, which lies beyond the scope of the present study.

At the end of each season, the ratings are normalized to maintain a stable mean value. This normalization preserves the relative differences between agents while facilitating the initialization of new agents and stabilizing the overall scale of the rating system.

Since the Formula 1 interaction outcome is not binary but rather an ordered ranking of all race drivers, the interaction outcome were transformed. To adapt the task to an Elo-like format, each race is treated as a set of pairwise interactions: if driver i finishes ahead of driver j, this is interpreted as a victory of i over j. Thus, the race outcome can be represented as $\frac{N(N-1)}{2}$ pairwise interactions among all drivers.

For pairwise interactions in Formula 1, no additional parameters were introduced for computing the win probability beyond the agents' ratings. Before qualifying, all drivers are in equal conditions (although their cars differ, such effects cannot be explicitly modeled). Therefore, the probability that agent i beats agent j is simplified to the classical Elo form:

$$P_{ik} = \frac{1}{1 + 10^{\left(r_{jk} - r_{ik}\right)/400}}.$$

Regarding the update coefficient, all parameters introduced in Methodology section are incorporated. Since this environment has no distinct sub-environments and ratings are normalized after each season, it is reasonable to apply a simplified approach for computing new-agent ratings by introducing a single parameter r_0 instead of the vector R_1. Considering that the update-enhancement parameter α and the saturation threshold T_{sat} have an inverse dependence (the fewer interactions required for saturation, the stronger the early-stage enhancement must be), there is no need to optimize both. Thus, saturation level was fixed $T_{\text{sat}} = 10$, and the optimal value of α for this saturation level is determined using gradient descent. The same applies to the parameters r_0, K_0, $\text{degree}_{\text{interactions}}$, and $\text{degree}_{\text{agents}}$. Interaction outcomes S are fixed values and, like T_{sat}, are excluded from optimization.

Therefore, the loss function for optimization depends on the following parameters:

$$L = f\left(r_0,\, K_0,\, \text{degree}_{\text{interactions}},\, \text{degree}_{\text{agents}},\, \alpha\right).$$

The initial parameter values are chosen randomly: r_0 in the range 1500–2000, K_0 in the range 10–500, $\text{degree}_{\text{interactions}}$ and $\text{degree}_{\text{agents}}$ between 0.1 and 1, and α between 0.1 and 10.

2.2 Data

The input data for the proposed model is derived from the comprehensive dataset of Formula 1 World Championship race results spanning the years 1950 to 2024, available on Kaggle: `https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020`. This dataset provides detailed information on individual race outcomes, which serves as the empirical basis for the estimation and validation of the generalized Elo rating system. To calculate ratings which are close to real ones the initial ratings for 1955 were computed using the results of all Grands Prix from 1950 to 1954 (inclusive). Data from 1955 to 1999 is reserved for model training, while the 2000 season is used as a validation set.

Also, drivers who completed only one race during a season were excluded from dataset. This serves two purposes. First, it prevents the formation of isolated subgroups of drivers who participated exclusively in a single Grand Prix (for example, the Indy 500) during the early years of Formula 1. Second, it avoids including test drivers who competed in the only one event, for whom there is insufficient data to reliably estimate a meaningful rating.

Finally, a total of 1,114 Grands Prix are considered across 75 seasons, with each season comprising between 6 and 24 events. For each Grand Prix, the number of drivers who completed the race and were included in the analysis ranges from 2 to 24.

2.3 Results and analysis

To prevent overfitting, an early stopping criterion was applied during gradient descent optimization. Specifically, training was terminated if the logistic loss on the validation set did not improve for 20 consecutive epochs. In our simulation, the procedure converged after 247 epochs, at which point early stopping was triggered. This indicates that the minimum validation loss was reached at epoch 227, after which further training produced no measurable improvement.

Optimal parameters are the ones which were calculated by Gradient Descent Optimizer after 227 epochs. Here the set of optimal parameters calculated:

Parameter	**Optimal value**
r_0	1834.6477
K_0	102.0843
$\text{degree}_{\text{interactions}}$	0.172968
$\text{degree}_{\text{agents}}$	0.624026
α	0.724035

Table 1: Optimal hyperparameter values.

The implementation of the model is available in the multi_elo repository on GitHub: `https://github.com/andritar/multi_elo`.

The developed approach (multi-agent Elo) was compared with endurance model, speed model, and pairwise comparison Elo model without number of agents and number of interactions normalization and initial interactions boost (multi-agent Elo default). The performance of all methods was assessed on the full set of evaluation-eligible interactions. The corresponding logistic loss values for each approach are reported in Table 2.

Approach	**Logistic loss**
Endurance Model	0.482522
Speed Model	0.502683
multi-agent Elo (default)	0.482469
multi-agent Elo (optimized)	0.481111

Table 2: Comparison of logistic loss values across different modeling approaches.

Among the considered approaches, the Speed Model exhibits the poorest predictive performance. The remaining three methods demonstrate substantially lower logistic loss values, indicating better predictive accuracy. As anticipated, incorporating normalization with respect to the number of agents and interactions significantly improves the model's performance, highlighting the benefit of these adjustments in the multi-agent Elo framework.

Conclusions

In this work, a generalization of the classical Elo system for multi-agent interactions is proposed, incorporating gradient descent for parameter optimization. The proposed approach provides:

- a formalization of multi-agent interactions as a collection of pairwise sub-interactions, followed by minimization of the discrepancy between expected and actual outcomes using a logistic loss function;
- analytical expressions for rating updates, derived from the computation of derivatives of the loss function, which ensure smoother and more stable learning dynamics;
- the introduction of an adaptive update coefficient dependent on the number of interactions and the number of agents, allowing the model to avoid excessive drift or excessive smoothing;
- justification of the principles for initializing the initial ratings of new agents and of the dependence of the correction coefficient on the number of interactions, increasing the flexibility and robustness of the model.

The combination of the Elo system with gradient descent creates the prerequisites for developing next-generation rating systems that preserve the intuitiveness and simplicity of classical approaches, while acquiring self-adjusting properties. This is particularly important in high-dimensional scenarios with a large number of agents and interactions, where traditional methods lose accuracy or stability.

The obtained results confirm the promise of integrating classical rating models with modern optimization techniques. The proposed model may serve as a basis for further research and practical applications in sports analytics, multiplayer gaming environments, and educational assessment systems.

This model operates at the level of individual agents but does not support situations in which agents form a team and compete against an opposing team. Extending the model to such cases is the objective of future research.

References

[1] Andrytsuliak T. L., Martyniuk S. V. *Elo system parameters optimization using gradient descent.* World of scientific research 2025, **40**, 55–63. (in Ukrainian)

[2] Angelini G., Candila V., De Angelis L. *Weighted Elo rating for tennis match predictions.* European Journal of Operational Research 2022, **297** (1), 120–132. doi:10.1016/j.ejor.2021.04.011

[3] Bradley, R. A., Terry, M. E. *Rank analysis of incomplete block designs: I. the method of paired comparisons.* Biometrika 1952, **39** (3/4), 324–345. doi:10.1093/biomet/39.3-4.324

[4] Elo A. E. The Rating of Chess Players, Past and Present. Arco Publishing, New York, 1978.

[5] Franceschi L., Donini M., Frasconi P., Pontil M. Forward and reverse gradient-based hyperparameter optimization. In: International conference on machine learning. PMLR, 2017. p. 1165-1173.

[6] Glickman, M. E. *Parameter estimation in large dynamic paired comparison experiments.* Journal of the Royal Statistical Society: Series C (Applied Statistics) 1999, **48** (3), 377–394. doi:10.1111/1467-9876.00159

[7] Hvattum L. M., Arntzen H. *Using ELO ratings for match result prediction in association football.* International Journal of Forecasting 2010, **26** (3), 460–470. doi:10.1016/j.ijforecast.2009.10.002

[8] Moore, J., Dottle, R., and Paine, N. Who's the best formula one driver of all time? 2018, https://fivethirtyeight.com/features/formula-one-racing/.

[9] Powell B. *Generalizing the Elo rating system for multiplayer games and races: why endurance is better than speed.* Journal of Quantitative Analysis in Sports 2023, **19** (3), 223–243. doi:10.1515/jqas-2023-0004

[10] Vovk V. *The fundamental nature of the log loss function.* Fields of Logic and Computation II: Essays dedicated to Yuri Gurevich on the Occasion of His 75th Birthday, 2015, 307—318.

[11] Weng, R. C., Lin, C.-J. *A Bayesian approximation method for online ranking.* Journal of Machine Learning Research 2011, **15** (2), 109–144. doi:10.1093/biomet/39.3-4.324

Received 20.11.2025

Тарас Андрицуляк, Сергій Мартинюк *Узагальнення системи Ело для взаємодій між багатьма агентами* // Буковинський матем. журнал — 2025. — Т.13, №2. — С. 137–151.

Стаття пропонує узагальнення класичної рейтингової системи Ело для багатокористувацьких середовищ і застосовує методи оптимізації для визначення оптимальної конфігурації моделі. Запропонований підхід представляє взаємодії між багатьма агентами як набір попарних порівнянь і формулює задачу оцінювання рейтингів як мінімізацію логістичної функції втрат. Така формулювання дає змогу отримати аналітичні вирази для оновлення рейтингів на основі градієнтів, що використовуються для пошуку оптимальних гіперпараметрів, забезпечуючи плавнішу та стабільнішу динаміку навчання порівняно з класичною системою Ело.

Вводиться додатковий адаптивний коефіцієнт оновлення, який залежить від кількості взаємодій і кількості агентів, задіяних у кожній події. Така нормалізація допомагає запобігти надмірному зсуву рейтингів у великих наборах даних і пом'якшує надмірне згладжування, коли обсяг даних або кількість агентів є невеликими. У статті також розглядаються принципи ініціалізації рейтингів нових агентів і динамічного коригування швидкості навчання на основі накопиченої інформації для кожного агента, що дає змогу системі швидше зближуватися до точних рейтингових значень і покращує загальну стабільність та інтерпретованість.

Градієнтний спуск використовується для пошуку оптимальних значень гіперпараметрів шляхом мінімізації функції втрат і автоматичного вибору відповідних параметрів. Це дає змогу системі формувати точніші рейтингові оцінки й, відповідно, забезпечувати кращі прогнози майбутніх взаємодій.

Запропонована система легко адаптується до середовищ із багатосторонніми взаємодіями, зберігаючи повну сумісність зі сценаріями бінарних порівнянь. За потреби її можна розширити додатковими гіперпараметрами для врахування специфіки конкретної доменної області.

Модель демонструє переваги поєднання класичної рейтингової методології з сучасними методами оптимізації й може ефективно застосовуватися у спортивних, ігрових та освітніх системах, де важливими є точність, адаптивність до складних сценаріїв та інтерпретованість.