How to be #1 in the IOI? A study on Rating Nations Participating in the International Informatics Olympiad

Mohamed O. I. Mahmoud[1]

Project Advisor: Dr. Timothy P. Chartier [2]

Abstract

This paper investigates the reliability of using Elo, TrueSkill, and Top Coder rating methods in analyzing the performance of nations participating in the International Informatics Olympiad from 2011-2022. This investigation aims to utilize the ratings to assist nations in improving and achieving more medals in future IOI contests. Based on ratings for whole contests and each problem category, including but not limited to graph theory, ad hoc, and data structures, we prove and compare the reliability of the rating methods by measuring their predictive accuracies. By taking Egypt as a case study, we show how to extract useful information from rating changes over time to assist in improvement. In addition, we use standardization and percentiles in locating Egypt, or any nation, in each category among other nations to find which categories weaken the whole contests ratings of Egypt. Thus, Egypt can focus on these categories for improvements. Moreover, we relate each specific range of whole contests percentiles to medal achievements, showing that nations in each range have nearly the same number and types of medals, which means that a country needs to get to a higher specific range of percentiles to get more and better in type medals. Ultimately, we set recommendations for future work, encompassing a sensitive analysis of which category is easier to improve and the usage of a modified Elo version.

1 Introduction

Over 300 high school computer science enthusiasts from almost 90 countries gather annually in summer to participate in the International Informatics Olympiad (IOI) [1]. The IOI is one of the five international science Olympiads. It is a competitive programming contest where participants are assigned six challenging computing tasks to solve. The tasks are distributed over two days. Each day, contestants have five hours to solve three tasks. These tasks measure students' abilities in various computer science aspects, including but not limited to graph theory, dynamic programming, data structures, and mathematics [2].

According to DMOJ, a Canadian competitive programming website, the problem categories that frequently appear in the IOI include ad hoc, graph theory, interactive, data structures, and a few other categories that we will refer to as "others", encompassing mathematical topics and algorithmic approaches such as dynamic programming, greedy, and divide & conquer [3].

[1] High School Senior '24, 6th October STEM High School for Boys (mohamed.1021106@stemegypt.edu.eg)
[2] Joseph R. Morton Professor of Mathematics and Computer Science, Davidson College (tichartier@davidson.edu)

Since the IOI is highly competitive, nations' coaches seek to prepare their trainees to get the highest ranks. Each year, medals are given to the top 50 percent of contestants. The medal distribution follows a ratio of 1:2:3:6 for gold, silver, bronze, and no medal respectively. The coaches strive to find techniques to help their nation's team achieve the highest number of medals.

With the advancement of data analytics, researchers have adapted techniques for analyzing and predicting the performance of individuals or teams in sports or competitions. Competitive programming contests are multiplayer contests where each contestant competes independently. Rating methods like Elo's rating method and Microsoft TrueSkill have been adapted to multiplayer competitions such as esports and Formula-1 to quantify contestants' skills [4, 5]. Nevertheless, there is a gap in the research on analyzing and predicting the performance of individuals or teams in competitive programming.

This study aims to provide valuable insights for researchers, data analysts, and national coaches analyzing nations' performance in the International Informatics Olympiad (IOI). By applying and comparing various Bayesian-based rating algorithms to the IOI datasets, the study aims to determine the most effective method for rating nations. These ratings will serve as a tool for coaches to assess their nation's performance in each category, helping them identify areas that need improvement.

Furthermore, the ratings will enable coaches to track their country's performance over the years, allowing them to identify improvement patterns. For instance, if Egypt's overall rating showed improvement from 2016 to 2019, the coaches of the 2023 team can consult with the coaches from 2016-2019 to understand the actions taken to enhance the Egyptian team's performance. Likewise, declining rating patterns can be studied to uncover the reasons behind the decline.

Finally, this research contributes to answering the question of "How to achieve medals in the IOI?" by providing insights based on effective rating algorithms and identifying focus areas for national teams to enhance their performance.

The study will be organized into four main sections. Section 2 will review literature related to the thesis and multiplayer rating methods. Section 3 will describe the datasets and the methodologies used to perform the ratings and analyze their effectiveness. Section 4 will comprise an analysis of the findings in addition to recommendations for future work in section 5.

2 Related Work

Chapter 9 in *Contest Theory* [6] gives a thorough explanation of the fundamental principles of designing rating systems proposed to rate players' skills based on contests' outcomes. It starts with noting that the purpose of rating is to determine the probability of a team winning its *m+1st* match, knowing that the team has won *w* matches out of its first *m*. There are two fundamentally independent approaches for getting the probability, of which Bayesian inference is mostly used in rating multiplayer systems. Bayesian inference assumes that the probability is a random variable on a prior distribution, usually normal distribution (aka Gaussian distribution). Afterward, it computes the posterior distribution conditional on the past contests' data. In addition, the chapter discusses the concept of online Bayesian inference. A Bayesian inference method is said to be online if it applies a Bayesian inference sequentially over successive rounds in which new input data is made available. For the specific problem of inferring contestants' strengths, the online Bayesian inference method updates the posterior distributions of the strengths of the alternatives as soon as a new round of rankings is made available. This update describes the rating of programming contests as there are always periodic contests on websites such as Top Coder or annual contests such as the IOI. Thus, the ratings need to get updated after each contest. Ultimately, the chapter discusses other topics, including but not limited to Factor Graphs and Gaussian Density Filtering. It concludes with a discussion of popular Bayesian rating methods: Elo's method, TrueSkill Method, and Top Coder method.

In the 1950s, Arpad Elo developed the Elo rating system to be used in chess rating for skill estimation and tournament sectioning [7]. The system calculates a numerical rating, usually between 0 and 3000, for every player based on competitive chess performance. When two players compete, the rating system expects that the player with the higher rating wins. Based on two equations that will be discussed in section 3.2.2, Elo calculates a player's expected score and uses this score to update the player's rating. Elo is mainly used for pairwise comparisons. Nevertheless, it can be used for multiplayer ratings by applying some modifications.

Researchers in [8] use the Elo rating system to rank scientific journals. They address the problem of evaluating journals with respect to a given year. This problem can negatively affect the prestige of a journal if its evaluation significantly drops in a specific year despite its performance being good in the years before. Elo

deals with this problem by considering the performance until a specific year to specify the rankings. The paper's approach to ranking journals simulates Elo to journals over the years based on each journal's SNIP (source normalized impact per publication). Like IOI, if we assume that journals are participating in a competition, the competition will be a multiplayer competition where players compete independently. To apply Elo to this competition, researchers applied pairwise comparisons. The dataset is composed of 8246 journals. The Elo update for each journal is the sum of each pairwise Elo update result with the other 8245 journals. The ratings are changed annually, where the Elo update of each journal is added to the previous Elo rating. In addition, the researchers restrict volatility by allowing a maximum rating difference to be 400 in the expected score equation, which will be discussed in section 3.2.2.

In [5], Microsoft researchers present TrueSkill**:** a Bayesian skill rating system proposed for determining players' skills in multiplayer online video games. TrueSkill addresses two main challenges that counter multiplayer games. Firstly, the game outcome often refers to a team of players while skill rating for individual players is required for future matchmaking. Secondly, two or more players or teams compete, so the outcome of a game is permutations of teams or players rather than a winner or loser. Our study aims to rate nations as a whole team, unlike video games, rather than individuals. This purpose means that rating IOI is facing the first challenge since contestants participate as individuals rather than as a team. Moreover, the second challenge applies to competitive programming as many contestants compete independently. Hence, there is no winner or loser. To solve the first challenge, TrueSkill assigns the sum of the performances of the team members to the performance of a team. According to TrueSkill, p_i (the performance of an individual *i*) relies on the Elo's method performance calculation, which specifies that the performance is a Gaussian distribution centered around S_i (skill of player *i)* with fixed variance $ß^2$. Getting the sum of the performances works for video games since the teams compete directly. Nevertheless, in competitive programming, teams compete independently. As a result, summing will not be fair as it will give an advantage to teams with more individuals. Factor graphs are employed to map each team of players to a distribution; according to this mapping and the update formulas, each player's skill is estimated by distributing the skill updates of the team. Finally, TrueSkill achieves online Bayesian inference using Gaussian Density Filtering that approximates the posterior distribution to be Gaussian and uses it as the prior distribution for the next game.

Ebtekar & Liu [9] present a novel Bayesian rating system for contests with many participants called Elo-MMR. Elo-MMR focuses on four main aspects: accuracy, computing efficiency, incentive compatibility, and human interpretability. The researchers defined accuracy as the ability to predict future contests' outcomes, computing efficiency as the ability to run with the least amount of time or memory, incentive compatibility as the prevention of a player's rating increase when performed worse or rating decrease when performed better, and human-interpretability as the capability of being understood effortlessly by players to understand and predict how their performances affect their rating. MMR in Elo-MMR stands for Massive (supports any number of players with linear runtime), Monotonic (synonym for incentive-compatible), and Robust (has bounded rating changes with more minor changes for consistent players than volatile players). By developing a Bayesian model and taking the limit as the number of participants goes to infinity, Elo-MMR achieves a generalization of the two-player Glicko (a Bayesian rating method successor to Elo), allowing any number of players. Elo-MMR has dealt with the flaw found in various rating methods, such as Glicko and Top Coder, known as "volatility farming." Sometimes, these attacks can inflate a user's rating several hundred points above its natural value, producing essentially impossible ratings to beat via honest play. Based on pair inversion comparison (the average of the pairwise comparison of the correct predicted pairs after each round based on the rating of the past rounds) and ranking deviation comparison (getting the difference between the predicted ranking and actual ranking divided by the number of players for each player, summing them, and getting their average), Elo-MMR proves that it outperforms in predictive accuracy when compared to other multiplayer rating algorithms, including but not limited to Codeforces, Top Coder, and TrueSkill.

3 Data and Methodology

3.1 Datasets

The official IOI statistics website [1] comprises the IOI contests data from the first contest in 1989 until the last contest in 2022. The complete and achieving final form of IOI contests data is available only during the period 2011-2022. Hence, the research focuses on the contests between 2011 and 2022 for unbiased results. For each contest, the data shows the country, the score of each problem, and the absolute total score for each individual. To scrape the data, we used Google Sheets to import the data from the IOI website in addition to Microsoft Excel and Python scripts for cleaning and manipulating the data.

As mentioned earlier in this paper, our study aims to rate nations, not individuals. Since the IOI is individual-based, we had to find a method for turning the data into nations-based data. Given that the IOI contestants compete independently, using the simple TrueSkill method of summing the individuals' scores to act as the nation's score will cause a flaw in ratings, giving an advantage to nations with more individuals participating. Hence, we considered the problem score of each nation to be the average of the nation's individuals' problem scores. For each contest, let us consider $\rho = 1, 2, \dots, 6$ as the index of each problem, $NS_{i,\rho}$ as the score of nation i in problem ρ, n_i as the number of individuals of nation i, and $S_{i,j,\rho}$ as the score of individual j from nation i in problem ρ. (1) shows how the nation's problem score is calculated.

$$NS_{i,\rho} = \frac{\sum_{j=1}^{n_i}(S_{i,j,\rho})}{n_i}. \quad (1)$$

As mentioned previously, the study aims to analyze nations' performances in each problem category, not just the whole contests. Hence, the category of each problem is derived from DMOJ [3]. For each contest, six datasets are made, representing rankings Per Whole Contest, Per Ad Hoc problems, Per Graph Theory problems, Per Data Structures problems, Per Interactive problems, and Per "Others" problems. The rankings of each dataset are merely based on the total absolute scores of nations in the specified problems for each category. For instance, problems 1, 2, 4, and 5 in the 2011 contest are graph theory problems. The rankings of the graph theory dataset of the 2011 contest are based on the sum of the scores of the graph theory problems for each nation.

3.2 Rating systems

Our study adapts the Bayesian-based Elo, TrueSkill, and Top Coder rating systems to the IOI datasets. Section 3.2.1 will present predictive accuracy: a method that would determine the effectiveness of each rating method. Sections 3.2.2 and 3.2.3 will discuss Elo theoretically, how to adapt it for multiplayer IOI contests, and how to modify it to get the highest predictive accuracies. Top Coder rating system—already being used in competitive programming contests' ratings—will be investigated theoretically in section 3.2.4. A glimpse of the theoretical part of TrueSkill was examined in section 2. The implementation of TrueSkill to IOI data was done using "trueskill" python module.

3.2.1 Predictive accuracy for effectiveness comparison

Predictive accuracy is the measurement of the ability of a rating system to predict the rankings of a contest from the accumulative ratings of the preceding contests. The predictive accuracy adapted in our study is the pair inversion method used by Elo-MMR [9]. Let *i+1* be the contest that follows the contest *i.* The method distributes the contestants of *i* and *i+1* into a combination of pairs. Then, it checks if each pair's rankings based on the ratings after *i* is predicted correctly according to the rankings of *i+1*. Finally, the number of correct predictions is divided by the number of combinations and multiplied by 100, calculating the percentage of predictivity. (2) clarifies the pair inversion predictive accuracy method for each contest:

$$acc_i(\%) = \frac{\#\ correctly\ predicted\ pairs}{C(N_i,\ 2)} * 100, \tag{2}$$

where acc_i is the predictivity of contest i, N_i is the number of nations participating in contest i, and $C(N_i, 2)$ is the number of combinations.

To compare the effectiveness of the rating systems for each category, the average of all the predictive accuracies of contests is taken and considered as the predictivity of a rating system in that specific category.

3.2.2 Elo Rating Method

As the initial rating for new contestants is independent of the final ratings, we chose the initial rating of each nation ($R_{a,0}$) to be 1500 (a sufficiently large number to prevent dealing with negative Elo ratings). The two main steps of rating comprise (i) calculating the expected score and (ii) updating the player's rating [8]. The match outcome of two players can be approximated to (3):

$$E_a = \frac{1}{1 + 10^{(R_b - R_a)/400}}, \quad (3)$$

where E_a is the expected score of player *A*, based on the unknown strength for both players (R_a and R_b).

(4) is the rating updating formula:

$$R_{a,i+1} = R_{a,i} + k\,(S_a - E_a). \quad (4)$$

The new rating of player *A* in contest *i+1* that follows contest *i*, is the player *A's* old rating plus the difference between S_a, the actual match score for player *A*, and the expected score of player *A* (E_a), weighted by the scaling factor *k* that controls how fast a rating can evolve.

Applying Elo to multiplayer IOI ratings is similar to applying it to journals in [8]. To apply Elo to an IOI dataset, pairwise comparisons method is adapted. Each dataset is composed of almost 90 nations. The Elo update for each nation is the sum of each pairwise Elo update result with the other 89 nations. For each category, the pairwise comparisons Elo is simulated over all the contests in which the category's problems exist.

3.2.3 Tuning the scaling factor k

The scaling factor *k* mainly used in chess is 32 for weaker players [7]. Nevertheless, the chess's scaling factor *k* is not optimal for all types of competitions. A too-small scaling factor *k* will lead to slow rating updates, so the ratings will not adapt to recent developments. Similarly, a large scaling factor *k* put too much weight on recent updates. Hence, the optimal scaling factor *k* can be discovered experimentally. By trying several values of scaling factor *k* from 1 through 50 with an increment of 0.1, the optimal scaling factor *k* for each category is determined based on the average predictive accuracy. *Table 1* specifies the optimal scaling factor *k* for each problem category based on the experiments shown in *Figures 1, 2, 3, 4, 5, 6*.

Per (Category)	Whole Contest	Graph Theory	Ad Hoc	Interactive	Data Structures	Others
Optimal *k*	26.6	12.2	17.2	9.4	18.0	4.4

Table 1: Optimal scaling factor k for each problem category.

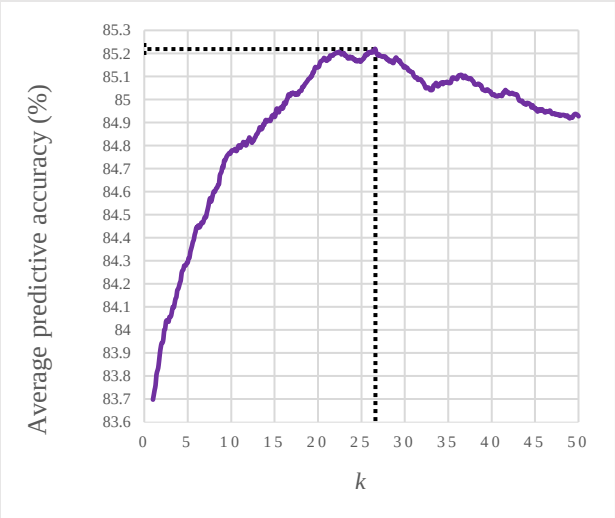


Figure 1: The effect of scaling factor k on the average predictive accuracy of ratings Per Whole Contest.

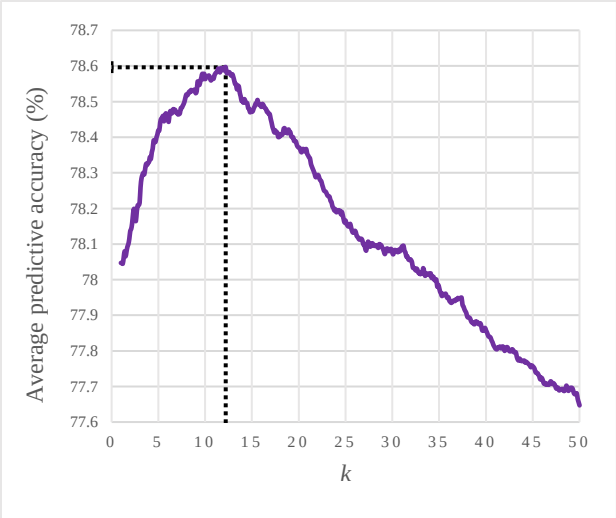


Figure 2: The effect of scaling factor k on the average predictive accuracy of ratings Per Graph Theory.

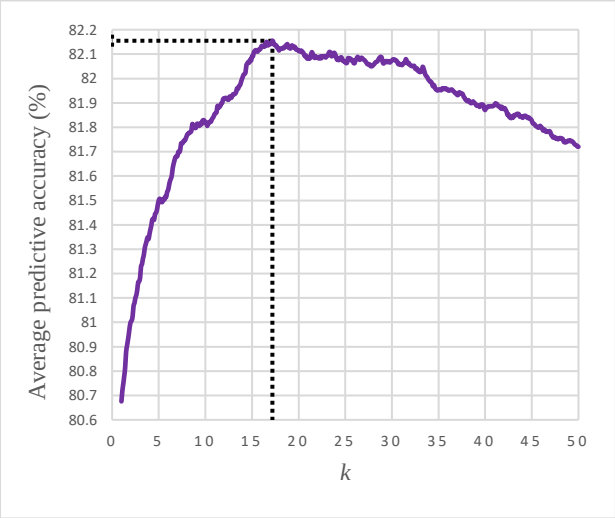


Figure 3: The effect of scaling factor k on the average predictive accuracy of ratings Per Ad Hoc.

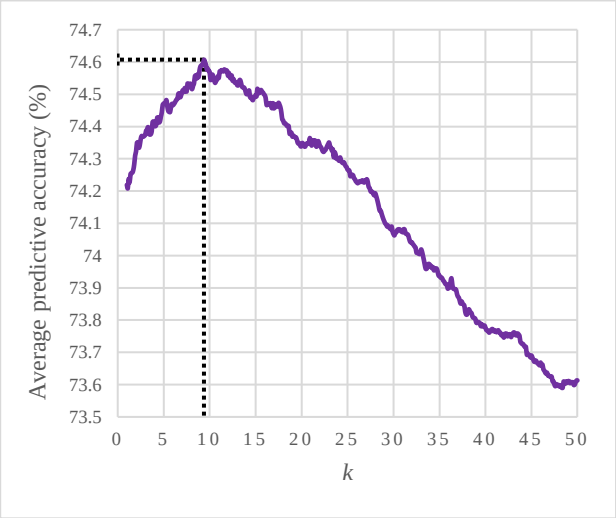


Figure 4: The effect of scaling factor k on the average predictive accuracy of ratings Per Interactive.

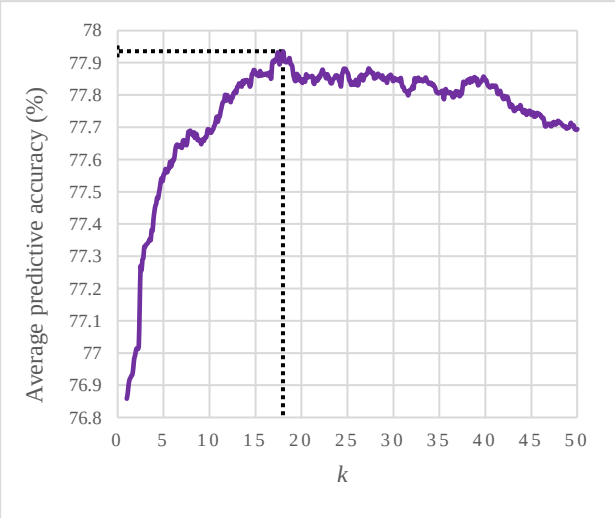


Figure 5: The effect of scaling factor k on the average predictive accuracy of ratings Per Data Structures.

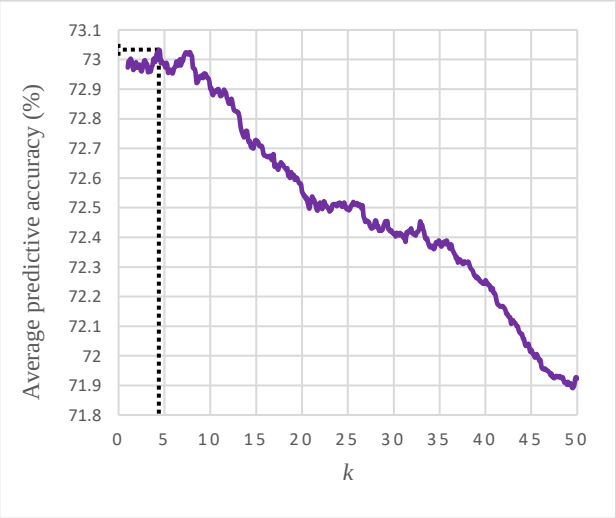


Figure 6: The effect of scaling factor k on the average predictive accuracy of ratings Per Others.

3.2.4 Top Coder Rating System

According to [10, 11], Top Coder deals with three crucial values: rating, volatility, and number of times a player participated in contests. To perform a contest's ratings, firstly, newcomers are all treated with rating 1200, volatility 535, and number of previous participations 1: the default initial values declared by the system creators. Secondly, the average rating of all players is calculated. Thirdly, contest's challenge factor—measurement of hardness—is calculated based on (5):

$$CF = \sqrt{\frac{\sum_{i=1}^{N} voltality_i^2}{N} + \frac{\sum_{i=1}^{N}(Rating_i - AvgRating)^2}{N-1}}, \tag{5}$$

where *i* is player in a set of players participating in a contest numbered from 1 to *N* (number of players). Afterwards, win probability between each two players is calculated based on (6):

$$WP_{i,j} = 0.5 \left(\text{erf}\left(\frac{Rating_i - Rating_j}{\sqrt{2(voltality_i^2 + voltality_j^2)}} \right) + 1 \right), \tag{6}$$

where erf is the error function: $erf(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2}\, dt$.

Then, the expected rank of each player is calculated using (7):

$$ER_i = 0.5 + \sum_{j=1}^{N-1} WP_{i,j}, \tag{7}$$

where $WP_{i,j}$ is the win probability of player *i* against other players.

Next, the expected performance of each player is calculated as shown in (8):

$$EP_i = -\Phi\left(\frac{ER_i - 0.5}{N}\right), \tag{8}$$

where Φ is the inverse of the Cumulative Distribution Function (CDF) of the standard normal distribution.

Similarly, the actual performance of each player is calculated based on (9):

$$AP_i = -\Phi\left(\frac{ActualRank_i - 0.5}{N}\right), \tag{9}$$

where $ActualRank_i$ is the actual rank of player *i* in the contest to be rated based on the absolute score.

After, the actual performance may differ from the expected one. This means that the rating and volatility of a contestant did not predict the result correctly, and they shall be updated. Using (8) and (9), we may compute the rating—the *PerfAs*—that would correspond to this performance as shown in (10):

$$PerfAs_i = OldRating_i + CF * (AP_i - EP_i). \tag{10}$$

Weight of challenge for each player is calculated based on (11):

$$W_i = factor * \frac{1}{\left(1 - \left(\frac{0.42}{TimesPreviouslyPlayed_i} + 0.18\right)\right)} - 1, \tag{11}$$

where *factor* is 1.0 for contestants with ratings below 2000, 0.9 for contestants within ratings 2000-2500, and 0.8 for contestants with ratings above 2500.

Then, a cap—the maximum rating change for each player—is calculated using (12):

$$cap_i = 150 + \frac{1500}{TimesPreviouslyPlayed_i + 2}. \tag{12}$$

The new rating of each player is based on (13):

$$NewRating_i = \frac{OldRating_i + W_i * PerfAs_i}{1 + W_i}.^{3} \tag{13}$$

If the $cap_i < |NewRating_i - OldRating_i|$, then:

$$NewRating_i = OldRating_i \pm cap_i. \tag{14}$$

Finally, the new volatility of each player is calculated using (15):

$$NewVoltality_i = \sqrt{\frac{(NewRating_i - OldRating_i)^2}{W_i} + \frac{OldVoltality_i^2}{W_i + 1}}. \tag{15}$$

However, (15) is only used after the second game. The first new volatility is determined to be 385.

[3] $NewRating_i$ is the $PerfAs_i$ with the influence of weight of challenge W_i on the rating change.

4 Analysis

After applying the rating methods to the IOI datasets, section 4 evaluates the findings. Firstly, section 4.1 assesses the rating methods' performances, considering each's capability in predicting future contests' rankings in addition to proving the reliability of using the output as feedback for nations' skills. Secondly, section 4.2 shows how the results can be used to help nations enhance their skills and gather extra medals by studying Egypt as a case study.

4.1 Assessing the rating systems' performances

Table 2 unveils a sample showing the ability of each rating system to predict the rankings of the latest contest (2022) from the preceding overall ratings. Elo rankings and TrueSkill rankings are almost similar, and they are the closest to the actual rankings. Top Coder shows mostly correct predictions for the top 10 nations. Nevertheless, the system needs to be more precise to determine the exact rankings, particularly in the case of the Republic of Korea.

	Actual rankings of "Per Whole Contest" in 2022 contest	**Predicted rankings for 2022 contest by rating systems in "Per Whole Contest"**		
Rank	**Actual**	**Elo**	**TrueSkill**	**Top Coder**
1	China	China	China	China
2	USA	USA	USA	Russia
3	Japan	Russia	Russia	USA
4	Russia	Republic of Korea	Japan	Japan
5	Republic of Korea	Japan	Republic of Korea	Taiwan
6	Canada	Iran	Iran	Poland
7	Taiwan	Canada	Taiwan	Iran
8	Ukraine	Taiwan	Poland	Bulgaria
9	Iran	Singapore	Romania	Romania
10	Romania	Vietnam	Singapore	Republic of Korea

Table 2: A sample of the difference between the actual rankings of 2022 contest's "Per Whole Contest" category and the predicted rankings based on the previous contests' overall ratings by the three rating methods.

Table 3 shows the average predictive accuracy of each rating method for each category. From the table, we can declare which rating method is better for which category. The predictivities of Elo and TrueSkill are too close. Although Top Coder is mainly used for competitive programming, it was found to be the worst performing in the IOI case among the other rating systems. Compared to the results of Elo-MMR [9], our study shows impressive results according to pair inversion

predictivity in all the three rating systems. This proves the reliability of using the ratings as feedback for skill by nations' coaches to utilize them in improvement.

	Average Predictive Accuracy Per					
Rating system	Whole Contest	Graph Theory	Ad Hoc	Interactive	Data Structures	Others
Elo	**85.2189%**	78.5965%	**82.1554%**	74.6074%	**77.9353%**	**73.0332%**
TrueSkill	84.8023%	**78.6945%**	81.7188%	**74.8729%**	76.2757%	72.8895%
Top Coder	82.2446%	77.0277%	78.7000%	73.7930%	75.1334%	72.1569%

Table 3: The average predictive accuracy for each rating method in each category.

4.2 Case study: How can the ratings help Egypt improve?

The first significant question is which system's ratings to consider. The simple answer is all. We can rely on all of them as they all have high predictivities. Nonetheless, for each category, we prefer to consider the ratings of the system with the highest predictivity.

Figures 7, 8, and 9 illustrate the ratings of Egypt over the years in each category. From these figures, Elo maintains consistent volatility that is not too high or too low based on the scaling factor *k*. This advantages Elo because tuning the scaling factor *k* ensures that Elo runs at maximum efficiency. TrueSkill has high volatility during the first three rounds until it reaches consistent ratings. TrueSkill considers the first three rounds as the number required to reach stable ratings from non-stable ones. Once the stable condition is reached, the volatility gets too low. Top Coder shows high volatility during the whole period of ratings. This disadvantages Top Coder as the system sometimes gives too much unnecessary weight to some contests.

As mentioned in the introduction, the ratings will enable coaches to track their country's performance over the years, allowing them to identify improvement patterns. If we consider the Per Whole Contest Elo ratings because it has the highest predictivity, we will find that Egypt's rating significantly increased from 2019 to 2020. Hence, the coaches of the 2023 team can consult with the coaches from 2019-2020 to understand the actions taken to enhance the Egyptian team's performance. To sum up, helpful information can be inferred from the ratings by analyzing the patterns.

O date denotes the moment before the performance of ratings when all nations had initial ratings, Black (C) denotes Per Whole Contest, Orange (GT) denotes Per Graph Theory, Grey (AH) denotes Per Ad Hoc, Yellow (IN) denotes Per Interactive, Purple (DS) denotes Per Data Structures, and Green (OT) denotes Per Others

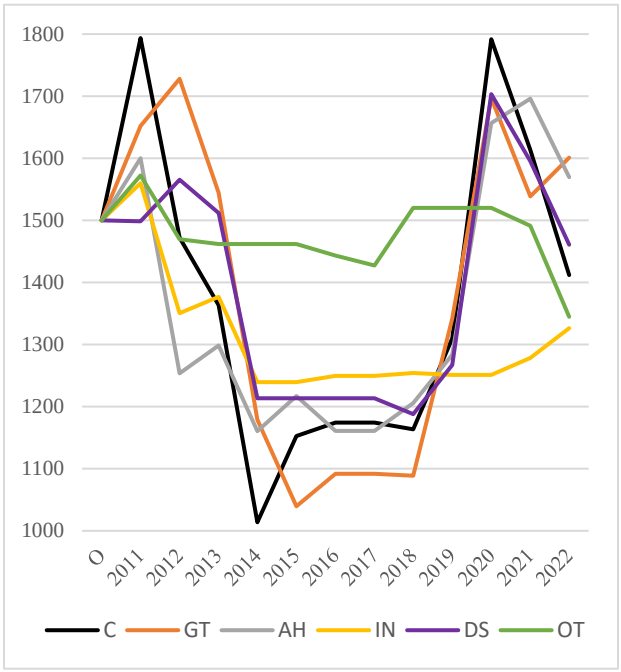


Figure 7: Egypt's Elo Ratings over years.

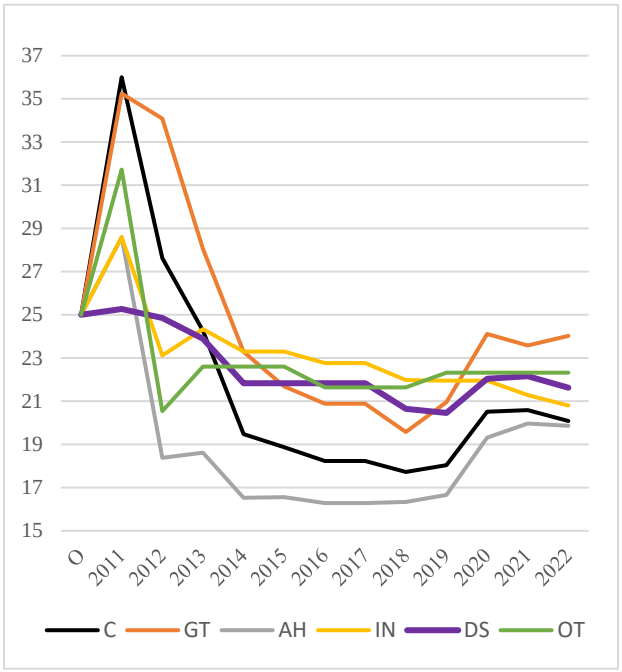


Figure 8: Egypt's TrueSkill Ratings over years.

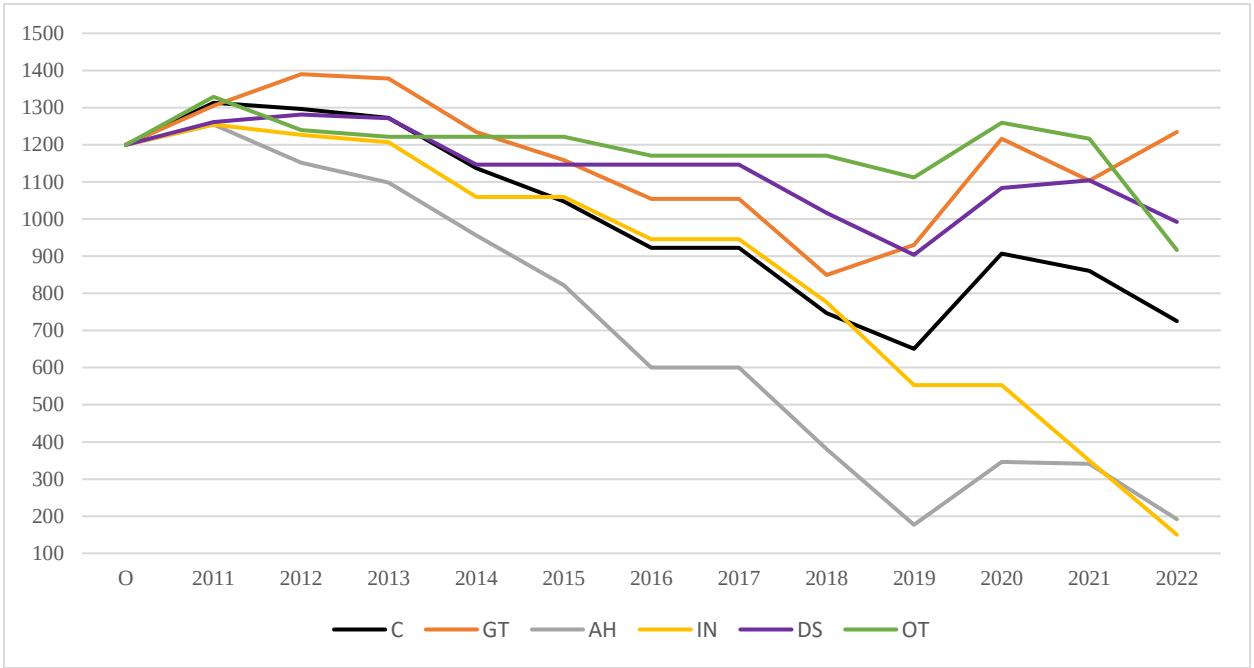


Figure 9: Egypt's Top Coder Ratings over years.

To assess the performance of Egypt or any other nation, we consider the rating system with the highest predictive accuracy for each category. Thus, comparing ratings as numbers will not reflect any helpful information as the scale of each rating system differs. For instance, without comparison charts, we cannot compare a score of 1500 on the SAT test and a score of 33 on the ACT test. In addition, the rating does not declare the performance of a nation, among others. For example, a nation with a rating of 2000, according to Elo in Per Graph Theory, might be in the top 10 percent among others, while a nation with a rating of 2500, according to Elo in Per Whole Contest, might be in the top 20 percent among others. Hence, standardization is crucial.

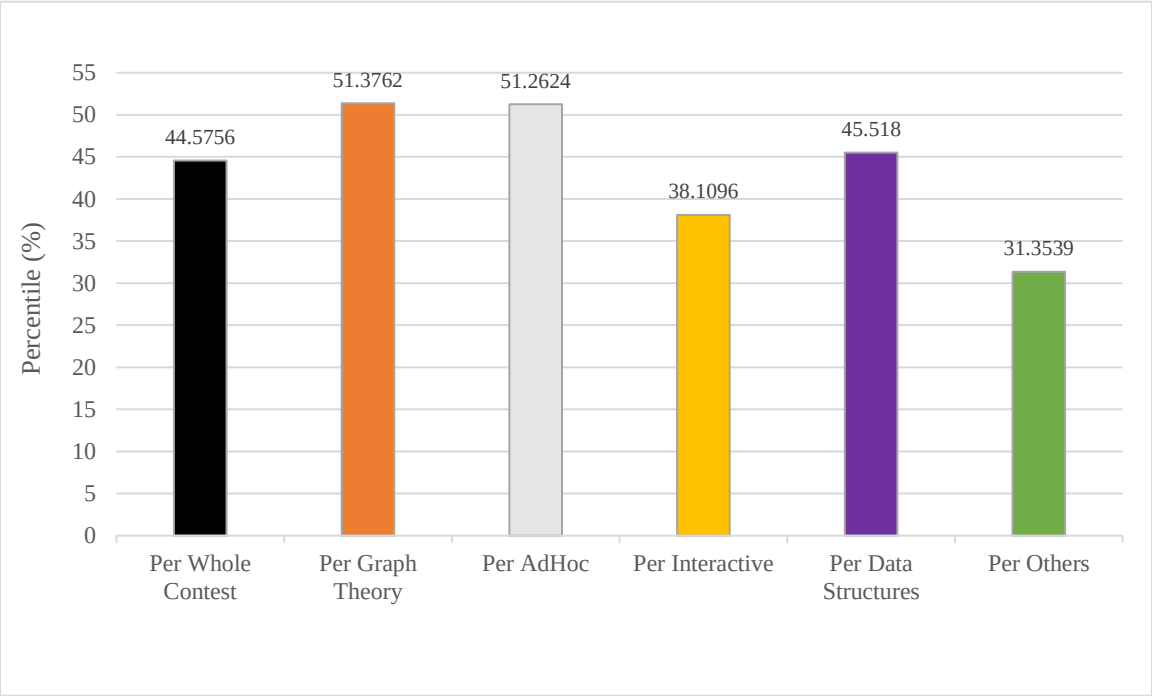
The standard score is the number of standard deviations in which a raw score is above or below the mean value of a sample of values. The standard score has many applications. One is standardizing scores of college tests such as the ACT and SAT. Since both have different scales, the *Z*-score helps in comparison by standardizing the scores. (16) represents the *Z*-score:

$$Z = \frac{x - \mu}{\sigma}, \tag{16}$$

Where Z is the standard score of a value of the data, x is the value of the data, μ is the mean, and σ is the standard deviation of the data. Standard scores help find the percentile, the value of which the data falls, of a sample, among others, based on the standard normal distribution. For example, if a team is at the 80% percentile, it is better than 80 percent of the teams. *Z*-scores can be turned into percentile using *Z*-score tables or calculation methods that rely on integral calculus. Using the Python statistics module to calculate the mean, standard deviation, and cumulative density function, we compare the output of the various rating methods by turning the *Z*-score into a percentile.

Figure 10 shows the percentile of Egypt in each category. Egypt is performing better at graph theory and ad hoc problems than other problems. As Egypt's percentile in whole contest rating is below 50 percent, Egypt is more unlikely to receive medals than other nations because, as mentioned earlier, only the top 50 percent receive medals. Nevertheless, by achieving consistent performance in the categories with the higher performances and concentrating on categories with lower performances, such as the interactive category, Egypt can reach a higher whole contest percentile, achieving more medals in the future.

Figure 10: The percentile of Egypt among other nations based on the latest ratings calculated after 2022 contest.



Nation	Percentile (%)	Rank	Gold	Silver	Bronze
China	98.25	1	4	0	0
USA	97.96	2	3	1	0
Russia	97.07	3	3	1	0
Japan	96.45	4	4	0	0
Republic of Korea	96.18	5	2	2	0
…					
India	77.26	21	0	2	2
Kazakhstan	72.95	22	0	2	2
Turkey	69.49	23	0	2	2
Brazil	67.66	24	0	3	1
Italy	65.84	25	0	1	3
…					
Switzerland	50.30	44	0	0	2
Macau	47.39	45	0	1	2
Latvia	46.24	46	0	0	2
Sweden	45.44	47	0	0	1
Egypt	**44.57**	**48**	**0**	**0**	**2**
Finland	44.32	49	0	0	2
Cuba	42.08	50	0	1	0
Mongolia	41.92	51	0	0	2
Cyprus	39.83	52	0	0	1
…					
Peru	29.49	61	0	0	1
Spain	29.41	62	0	0	1
Azerbaijan	27.84	63	0	0	1
Tajikistan	25.38	64	0	0	1
Norway	25.37	65	0	0	0

Table 4: A sample of medal achievements of few nations in 2022 contest. The ranks and percentiles are based on latest Per Whole Contest ratings after 2022 contest.

Finally, *Table 4* gives insights into the medal achievements of a few nations, showing their latest Per Whole Contest ranks and percentiles based on Elo in the last IOI 2022 contest. We can deduce from the table that nations within specific ranges of ranks nearly achieve the same number and types of medals. For instance, nations within ranks 44-52 achieve mostly two medals, and they are almost bronze. Hence, a nation needs to reach a lower range of ranks within which it can achieve more and better in type (gold > silver > bronze) medals. The ranges can be noticed from the full version of *Table 4* found on *"github.com/MoiMohamed/IOIRatings"*—the link comprises all the data, the codes of the rating methods, and the results.

5 Conclusion and Future Work

This paper proposed insights into improving and achieving more medals as a nation in the International Informatics Olympiad. By applying and comparing Elo, TrueSkill, and Top Coder rating methods to the IOI datasets between 2011 and 2022, we could analyze the performance of nations through ratings of whole contests and specify a rating for each problem category. The findings proved the reliability of rating methods in estimating nations' skills, as the rating methods showed high predictive accuracies. It was found that the performance of Elo and TrueSkill is nearly the same, with the highest predictive accuracies. Nevertheless, Top Coder, which is already being used in rating competitive programming contests, showed lower predictive accuracies.

By studying the patterns in rating changes over the years, the nations' coaches can utilize the ratings to know how to improve by assessing the periods of nations' improvement in performance. In addition, this study showed a method to compare the performances of each nation in each problem category based on the concepts of *Z*-scores, percentiles, and standardizing. A nation can assess its performance by observing its percentile in each category among other nations. Thus, a nation can work to improve in the categories that weaken their final Per Whole Contest ratings. From relating percentiles to medal achievements, we deduced that each specific range of percentiles achieved nearly the same number and types of medals. Hence, these ranges can be used to know how much improvement in percentiles is needed to achieve more and better medals.

Finally, we recommend a sensitive analysis of the results for future work to find which categories are easier to improve. Moreover, we recommend running the

same experiments using more advanced versions of Elo, such as Elo-MMR, as Elo's modifications can help achieve better predictive accuracies.

6 Bibliography

[1] IOI Statistics, https://stats.ioinformatics.org/.

[2] T. VERHOEFF, 20 years of IOI competition tasks - ioinformatics.org, https://ioinformatics.org/journal/INFOL047.pdf.

[3] "IOI '22 P1 - Catfish Farm - DMOJ: Modern online judge," DMOJ, https://dmoj.ca/problem/ioi22p1.

[4] R. D. and N. P. Justin Moore, "Who's the best formula One driver of all time?," FiveThirtyEight, https://fivethirtyeight.com/features/formula-one-racing/.

[5] R. Herbrich, T. Minka, and T. Graepel, "TrueSkillTM: A bayesian skill rating system," *Advances in Neural Information Processing Systems 19*, pp. 569–576, 2007. doi:10.7551/mitpress/7503.003.0076

[6] M. Vojnovic, *Contest Theory - Incentive Mechanisms and Ranking Methods*. Cambridge University Press, 2016.

[7] M. E. Glickman and A. C. Jones, "Rating the chess rating system," *CHANCE-BERLIN THEN NEW YORK*, vol. 12, pp. 21–28, 1999.

[8] R. Lehmann and K. Wohlrabe, "Who is the 'Journal Grand Master'? A new ranking based on the Elo rating system," *Journal of Informetrics*, vol. 11, no. 3, pp. 800–809, 2017. doi:10.1016/j.joi.2017.05.004

[9] A. Ebtekar and P. Liu, "Elo-MMR: A rating system for massive multiplayer competitions," *Proceedings of the Web Conference 2021*, 2021. doi:10.1145/3442381.3450091

[10] Top Coder Rating System, https://www.topcoder.com/thrive/articles/Ratings.

[11] M. Forišek, *Theoretical and Practical Aspects of Programming Contest Ratings*. [Online]. Available: https://people.ksp.sk/~misof/publications/2009thesis.pd