Binary Voting System

Dany Poly[1], Benjamin Aji Joseph[2], Pranay Gupta[3]

[1-3]UG Students, Department of Computer Engineering, FCRIT, Mumbai University, Navi Mumbai, India

---

Abstract – *General voting system works in a way where an individual is given a set of options and has to select one option from the given options. In elections, the first-past-the-post system is used, where the highest-polling candidate wins the election. In this paper, we bring a change to the general voting system. Instead of a person choosing from various different options, we ask the person to choose from a 1v1 random option multiple times. Through this, we could achieve a true winner among all as there would be fewer options available for a person to vote from, thereby reducing the variable required for a person to choose from. We'll be running the results provided by an individual voter through Elo rating algorithm in queue data structure. Through this system, we could eliminate any sort of partiality and irrational decision that the voter would make in a conventional voting system.*

Key Words: Elo Rating, Voting System, Queue

1.INTRODUCTION

A voting system generally consist of selecting one option from various options available. This particular process is repeated equal to the number of voters present. In this way, the option with the highest number of votes is considered as the winner. We provide a better and efficient way of providing the rating from a set of products. We intend to ask the user providing them with only 2 options at a time rather that asking them to select one among many. In this way an individual voter would be voting "n" times instead of only one time, but the individual would have to vote from only two options, and would have to select only one winner.

This system of voting is made with taking inspiration from a 1v1 rating system among a list on n objects, known as Elo Rating System. The Elo rating system is a traditional method used to calculate the relative skill levels of players in zero-sum games like chess. It is named after its creator Arpad Elo, a Hungarian-American physics professor. To program this kind of voting system we need to have a flexible and a dynamic approach with the help of data structures. We intend to use the Queue data structures to create memory for the objects and enqueue all the new objects they want to add simultaneously.

We use queue for the reason that Queues are flexible, requiring no communications programming. The programmer need not have any knowledge of inter-process communication. Data queues also allow for the computers to handle multiple tasks as this would help in handling various requests at a particular time. The queue can remain active when there are no entries, ready to process data entries when necessary.

1.1 Queue Data Structures

Queue Data Structure is discussed in this following paper because Data queues are a fast method of inter-process communication. Instead of using arrays to store up the votes by a particular individual data queues are used here. Data queues free up jobs from performing some work, which can lead to a better response time and an overall improvement in system performance. Data queues serve as the fastest form of asynchronous communication between two different tasks, since there is less overhead than with database files and data areas. Queues are flexible, requiring no communications programming.

1.2 Other Data Structures

Data Structures which could be helpful for the application would be linked list & doubly linked list

1. Linked List: - Will help to link the object and the rating of that particular object rather than creating a new queue for that.
2. Doubly Linked List: - Will help to create the link between object, objects previous rating and objects current rating. Also, as dynamic initialization is there, there will be no wastage of memory.

2. Literature Review

Use of Elo-rating system to find dominant entities in a group has always been extensively used. A lot of methods of dominance rank ordination has been reviewed by de Viers, our paper focuses on these rank ordinations to find one true winner in a voting system rather than a rating system. Problems with matrix-based methods arrive at the time of calculation with linearity (i.e., if x is more dominant by a marginal factor than y, and y is dominant than z, then x is more dominant than z). This kind of system excels when the number of individuals in the system is less than 5. Thus, scientists face a huge problem when the sample size is very small.

Using two queues in graph growth algorithms is considered as the fastest shortest path algorithm on real road networks. This system if considered for a system in voting, could lead to quicker calculation and counts of votes across various networks.

3. Methodology

Considering the use of data structures for fast and vast computations, queues are used. Queues are the best applicable data structures that can be used, we need to have functions which would help in inserting and deleting values into the queue. So, two functions that can be used are INSQ and DELQ. We also need to have function which would find the probability and the Elo value.

Use of a random function is also needed to ensure that there is proper randomization of inputs provided to the user. The architecture of how we can proceed with the voting system has been mentioned below.

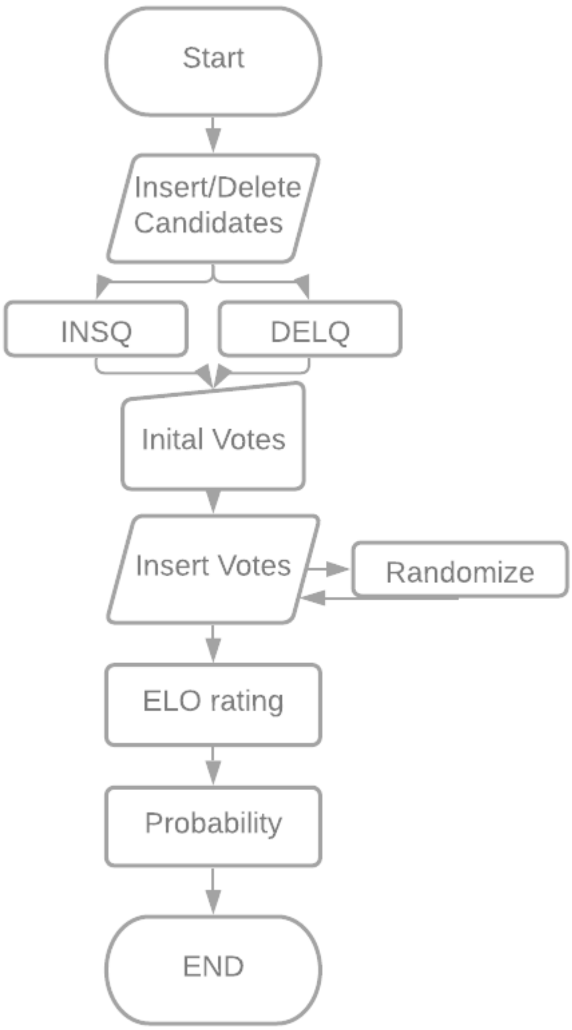

FIGURE 1:- FLOWCHART OF THE VOTING SYSTEM

The mentioned functions which require explaining have been discussed further. Additionally, all the algorithm of the functions has been mentioned.

2.1 Functions

Probability: - Finds the probability of one option being selected by the user.

P1: Probability of winning of player with rating2

P2: Probability of winning of player with rating1.

P1 = (1.0 / (1.0 + pow (10, ((rating1 – rating2) / 400))));

P2 = (1.0 / (1.0 + pow (10, ((rating2 – rating1) / 400))));

P1 + P2 = 1

ELO Rating: - The rating of an option selected by the user is updated using the formula given below.

rating1 = rating1 + K*(Actual Score – Expected score)

INSQ: - INSQ function is used to insert the objects provided by the user.

DELQ: - DELQ function is used to Delete the objects provided by the user.

2.2 Algorithm

INSQ
Step 1: If REAR >= SIZE – 1 then
Write "Queue is Overflow"
Step 2: REAR = REAR + 1
Step 3: QUEUE [REAR] = X
Step 4: If FRONT = -1 then
FRONT = 0.
Step 5: End

DELQ
Step 1: If FRONT = -1 then
Write "Queue is Underflow"
Step 2: Return QUEUE [FRONT]
Step 3: If FRONT = REAR then
FRONT = 0
REAR = 0
Else
FRONT += 1
Step 4: End

Probability
Step 1: Initialize P1, P2
Step 2: P1 = (1.0/(1.0+pow(10,((rating1 – rating2) / 400))));
P2 = (1.0/(1.0+pow(10,((rating2 – rating1) / 400))));
Step 3: End

ELO RATING
Step 1: Initialize Ra, Rb, K, Pa, Pb
Step 2: Check if d==1, if no, go to step 5
Step 3: Set Ra = Ra + K * (1 - Pa)
Set Rb = Rb + K * (0 - Pb)
Step 4: Set Ra = Ra + K * (0 - Pa)
Set Rb = Rb + K * (1 - Pb)
Step 5: End

3. FUTURE SCOPES

The possibility of adding a feature such as margin of victory within the Elo rating system could also be added. Currently, there are four models which give margin of victory in the Elo system. Those models are linear, joint additive, multiplicative and logistic.

Polling systems which give a forecast of the results could be replaced by Weight adaptation in the Elo rating system. This would help in much accurate results of the polls rather than counting of polls.

4. CONCLUSIONS

In this paper we have attempted to find a unique and better solution for voting systems removing the conventional parameters with the help of programming data structures. This paper has focused on creating a new system to reduce the irrational decisions made by the voter. Methodology can likewise be utilized in reality situation. Since randomizing is used, manipulation of votes or favorability of entities won't be there. A true result would be found with this paper eliminating any chance of partial favorable entities.

REFERENCES

[1] Andrei Chiriță, Camelia Delcea. (2020) A Laboratory Experiment for Analyzing Electors' Strategic Behavior in a First-Past-the-Post System

[2] Gregor Hohpe, Bobby Woolf, Kyle Brown(2004) Enterprise Integration Patterns, 2004 edition

[3] Jean Vuillemin, A data structure for manipulating priority queues, Communications of the ACMVolume 21 Issue 4, https://doi.org/10.1145/359460.359478

[4] Stephanie Kovalchik, Extension of the Elo rating system to margin of victory, International Journal of Forecasting, Volume 36, Issue 4, 2020, ISSN 0169-2070

[5] Neumann, C, Duboscq, J, Dubuc, C, Ginting, A, Irwan, AM, Agil, M, Widdig, A and Engelhardt, A (2011) Assessing dominance hierarchies: validation and advantages of progressive evaluation with Elo-rating. Animal Behaviour, 82 (4). pp. 911-921. ISSN 0003-3472

[6] StephanieKovalchik, Extension of the Elo rating system to margin of victory, ISSN 0169-2070, Volume 36, 2020 https://doi.org/10.1016/j.ijforecast.2020.01.006

[7] F. Benjamin Zhan, Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures, Journal of Geographic Information and Decision Analysis

[8] https://www.chess.com/blog/nivan100/elo-rating

BIOGRAPHIES

Dany Poly
Computer Engineering Student

Benjamin Aji Joseph
Computer Engineering Student

Pranay Gupta
Computer Engineering Student