Better Allocation to Reduce Voting Queue Length

Enas Mohamed, Hesham Naiem, and Mostafa Izz,
(em_cmp_eng , hesham.naiem)@yahoo.com),mostafa.3ez@gmail.com,
Faculty of Engineering, Cairo University

***Abstract*— Providing high turnout rate in elections is a desirable goal to reach democracy. In this paper we are extending some literature studies that were trying to provide equity between voters across all precincts which will result in reducing maximum waiting time between voters. In this paper we are extending the simulation model used in literature to study voting queues and in the same time we are comparing the results of the state of the art technique in distributing voting machines across precincts (namely GIA which is simulation based greedy algorithm) with our simple algorithm (RA which is Random Algorithm based on some heuristics for improvements) on a data conducted from 2004 US presidential election in one county.**

***Index Terms*— Simulation, Voting Queues, Voter Turnout**

I. Introduction

As waiting times increase, more voters are forced to leave without having cast their ballots due to impatience and other time commitments. Thus, it is very important to design voting systems that result in voters waiting the least amount of time possible. Ideally, we would provide a sufficient quantity of voting machines such that voters would never have to wait to cast ballots. However, due to the expense of the voting machines and the voting administrators we cannot reach the ideal case.

Perhaps even more importantly, voting systems should provide equity to all group of voters regardless of (geography, voting preference, race of voter, etc.) by having shorter lines in all precincts. This is a big concern in recent elections. Furthermore, Walter Mebane, Political Science Professor at the University of Michigan, has produced several studies criticizing the Department of Justice's response to the issues in Franklin County. Professor Mebane concludes from his research, "The allocation of voting machines in Franklin County was clearly biased against voters in precincts with high proportions of African Americans" [12].

Determining an optimal voting-machine allocation is challenging and difficult to solve for several reasons as discussed in [4]. "(1) Voter arrives randomly and according to non-stationary processes to polling locations. (2) Voter queues may not reach steady state. (3) Actual voting scenarios involve considerable computational complexity."

In large elections there are thousands of precincts and the input variables are stochastic. Thus, both building model that simulate elections and developing solution methods for it are challenging.

We model the voting process using a simulation model that allows us to employ non-stationary arrivals and non-steady state queues. We allocate voting machines to precincts using guided random with improvement heuristic and comparing results with greedy improvement algorithm discussed in [4]. The objective in our machine allocation is to provide voter equity across precincts. The rest of the paper is arranged as follows. Sections II and III provides information on motivation and more detailed problem definition. Section IV discuss the suggested algorithm RA and the state of the art GIA. Section V discuss the performance metrics, experimental design and results analysis. Section VI provides a brief review on related work in literature. Section VII discusses conclusions. And section VIII discusses limitations and future work.

II. Motivation

A referendum was held in Egypt on March 19 following the 2011 Egyptian Revolution. For most Egyptians, this was the first genuinely free vote in their lives, so there was large turnout for the referendum which reached 41.2 % of the 45 million eligible voters. As a result, the queues outside polling stations have continued to grow as Egyptians go out to cast their votes and voters waited patiently for hours in lines. This historical step toward democracy needs to be studied for improvement in the future.

III. Problem Definition

In most elections we see large lines of voters waiting in queues outside precincts to cast their vote. The voter sometimes has to wait for hours and some voters are forced to leave without voting due to impatience and other time commitments. This affects negatively on the turnout rate of the voters due to long waiting times. And we need to motivate people to contribute in the elections in Egypt because we are developing a new democracy life. The administration process mainly determine the parameters of the Election Day from the distribution of different precincts, to the total number of Machines used for voting and the distribution of these machines between different precincts. We have taken the path of searching for best allocation of the voting machines between different precincts. We are extending a simulation model used to show that the algorithm described in [4] – namely GIA - outperforms UEM (Utilization Equalization Method) which have been used in US 2004 presidential election in many counties. In this paper we are extending the simulation model used in [4], and comparing the results of our algorithm RA with the algorithm described in [4], GIA.

IV. Suggested Solution

A. *Queuing Model*

1) *Model Logic*

On the Election Day, all precincts in Ohio open at 6:30 am and close at 7:30 pm. Once a voter arrives at a precinct, the voter enters a single queue waiting for a DRE machine to be free. If time to close reached and there are voters waiting in queue, then the precinct must be open until all voters finishes and not allowing any new voter to enter queue during that time.

There can be one or more DRE voting machines inside each precinct. We assume that all DRE voting machines are identical and shared by all voters within a precinct and assume that voting machines are working all the time without failure.

Our simulation model provides the expected waiting time in each precinct for a given number of allocated voting machines. The numbers of DRE voting machines allocated to each precinct are our decision variables.

2) *Input Distribution Assumptions*

We use a data set based on statistics from the 2004 election in Franklin County, Ohio (available at <http://copperas.com/fcelection/>).

For the number of voter, we fit a normal distribution with mean 1070 and standard deviation 319 to the number of registered voters in each precinct, and then generate the number of voters at each precinct independently from this fitted normal distribution.

For the voter turnout rate, we fit a Weibull distribution with Shape Parameter α=6.9514 and Scale Parameter β=60.884 to turnout percentage in each precinct.

For the voting service time in every precinct we use a gamma distribution with shape parameter of 5.71 and scale parameter of 1.05 and 0.58 according to data from the 2006 Ohio gubernatorial election from sample of Election Systems & Software machines. Actual voter service times will depend on the length of the ballot which requires the voter to read and take decision of his vote, but in our simulation we assume that voting times in every precinct follow this same gamma distribution.

The arrival of the voters to each precinct is a non-stationary arrival process since it is not predicted due to some variables such as time of day, traffic and working hours for voters. There is difference in arrivals during the time of the day.

The Voting Experience Survey [6][11], which is based on a sample of voters throughout Ohio, provides the percentages of turnout voter arrivals by the time of day (see Table 1).

We assume that in each time period the number of arriving voters follows a Poisson distribution. The timing and size of these times of day may not be the same across all precincts due to differences in voters' differences, but here we assume that all precincts experience similar arrival patterns.

Table 1: Voter Arrivals by Time of Day

Period of Time	Percentage of Turnout Voters
Before 8 a.m.	20.61
8 a.m. – 11 a.m.	27.34
11 a.m. – 3 p.m.	24.05
3 p.m. – 5 p.m.	13.26
After 5 p.m.	13.87

B. *The Greedy Improvement Algorithm*

The Greedy Improvement Algorithm (GIA) was proposed in [4]. We used it to compare our new proposed method with it.

It is a heuristic method to allocate voting machines to precincts. It contains two phases: Phase I is a simple greedy heuristic and Phase II is a local improvement search. For more information refer to [4].

C. *The Random Algorithm*

This is our proposed method for allocating machines across precincts. It works by creating 4 combinations of machine allocation in precincts by using the random method, then choosing the best distribution that result in minimizing the max waiting time in all precincts, which will result to minimize the equity After that we combine this random method with some local search to improve the results.

Phase I: Iterative random method to find the best allocations in the precincts that reduce the max waiting time.

Step 1. Assign random values to x_i for each i except for $i = N$ (minimum value = 1, maximum value = $^{N}/_{4}$)

Step 2. $Counter$ = $\sum_{i=1}^{N-1} x_i$

Step 3. If $= N$, then $x_i = N - Counter$

Step 4. Run the simulation and calculate the max waiting time $W_i(x_i)$

Step 5. Go to step 1 and repeat 4 times, save each combination of x_i

Step 6. Choose the combination with the min waiting time

Phase II: Iterative improvement that works as follow

Step 1. Add Machine to the precinct with the maximum waiting time and remove one from the precinct with the minimum waiting time

Step 2. Run the simulation and calculate the equity (new equity)

Step 3. Repeat while (new equity < old equity); otherwise stop.

D. *Implementation*

We had two options to implement the simulation model which are: 1) Ready simulation modeling software such as Arena. 2) Standard Programming Language such as C++ or Java.

The Ready simulation software doesn't require great effort to simulate queues and obtain waiting times, but needs greater work to build an algorithm in it because it is not designed as a programming language. On the other hand, programming languages can simulate algorithms very easily but needs greater effort to simulate queues and obtain waiting time.

We preferred to use a general purpose programming language to implement the simulation model (namely C# .Net) with the help of open source .Net simulation library called React.NET [10]. We implemented the GIA and RA method using this library.

V. Performance Measurement and Analysis

A. Performance Metric

Some performance metric used in simulating elections is minimizing the total expected waiting time across all precincts, but it leads to long voter waiting times in some precincts and short voter waiting time in other precincts. This is not preferred since we need to make all voters are experiencing same situations across all precincts.

A better performance metric which provides equity to all voters was proposed in [4].

The proposed metric is the average absolute differences of expected waiting times among precincts which is defined in equation (1):

$$Z(X) = \sum_{i=1}^{N} \frac{\sum_{j=i+1}^{N} \left| W_i(x_i) - W_j(x_j) \right|}{\left(\frac{N(N-1)}{2} \right)}, \qquad (1)$$

Where N is defined as the total number of voting precincts, $X = (x_1, x_2, \ldots., x_N)'$, $x_i\ (i = 1, \ldots, N)$ is the number of voting machines allocated to precinct *i*, and $W_i(x_i)(i = 1, \ldots, N)$ is the expected waiting time for voters at precinct *i*.

To find the allocation that provides the best equity we need to solve the following optimization problem:

$$\min\{Z(X)\, X \in \, l\}$$

Where *l* is the set of feasible solutions, and | *l* | is finite.

B. Confidence Interval

A confidence interval (CI) is: “a particular kind of interval estimate of a population parameter and is used to indicate the reliability of an estimate.”[8]

In our case we need to calculate the equity performance parameter and we get a sample of its value from different replications. This sample has unknown mean µ and unknown standard deviation, so the confidence interval for the population mean, based on sample of size n is defined in equation (2):

$$\bar{X} \pm t \times \frac{s}{\sqrt{n}} \qquad (2)$$

Where $\bar{X}$ is the sample mean $\bar{X} = \frac{1}{n}\sum_1^n Xi$ and t is the critical value for the *t* distribution with *n-1* degrees of freedom and s is the estimated standard deviation known as standard error.

We implemented the calculation of confidence interval for the results of the equity metric in Microsoft Excel.

C. Experimental Design

In order to examine the performance of the simulation-based RA and compare it with the greedy improvement algorithm (GIA), we set up experimental design using four factors: Voting Time, Number of Precincts, Ratio of the number of machines to the number of precincts, and Allocation Strategy (see Table 2). The result of this experimental design is the “equity” metric given in (1).

Table 2: Factors and Values for Experimental Design

Factors	Possible Values
Number of Precincts	20 – 30 - 50 Precincts
Voting Time (Scale Parameter of Gamma Distribution)	0.58 - 1.05
#Machines/#Precincts	2 - 3.6
Allocation Strategy	RA - GIA

To obtain the allocations for each combination of design factors, we run the RA and GIA using the React.Net Library [10]. We use 50 replications for each scenario so that the 95% confidence-interval half width will be less than 10% of the average waiting time in a precinct.

There are 12 design points in total (see Table 3). We run 50 replications for each design point. Different random numbers are used for different design points to ensure that the design points are independent.

Table 3: Design Points

Design Point	Voting Time	No. of Precincts	No. of Machines
1	0.583	20	40
2	1.05	20	40
3	0.583	30	60
4	1.05	30	60
5	0.583	50	100
6	1.05	50	100
7	0.583	20	72
8	1.05	20	72
9	0.583	30	108
10	1.05	30	108
11	0.583	50	180
12	1.05	50	180

D. Results

The running of simulation for the RA and GIA methods shows that the GIA method takes very long time to run all iterations compared to RA, this is because Phase 1 of the GIA runs the simulation number of times equivalent to the number of DRE machines to be allocated in one replication, and the RA method allocates them random at Phase 1 of it. For example if we have 100 DRE Machines and 50 replications, the simulation will run for about 100*50 times in Phase 1 only which is a lot of time. So the RA outperforms the GIA in the speed of simulation.

The results of the experimental design are shown inside

Table 4 which contains the equity and confidence interval (CI) of the RA vs. GIA method through 50 replications of the simulation.

From the results we can say that RA method is significantly better than GIA at large numbers of DRE Machines which is 6 out of 12 combinations of design points, and in small numbers of DRE machines the GIA is slightly better than RA in 6 out of 12 combinations of design points and at the best result the equity is better with about 5 minutes less than RA equity result (First Design Point).

Table 4: Results of Experimental Design

DP	RA-Equity	RA - CI	GIA - Equity	GIA - CI
1	34.949	27.88 to 42.02	30.027	22.25 to 37.80
2	67.874	56.08 to 79.66	65.738	53.43 to 78.05
3	27.857	19.06 to 36.66	26.675	19.07 to 34.27
4	55.567	46.26 to 64.88	65.880	52.18 to 79.58
5	31.651	19.86 to 43.44	29.149	22.17 to 36.13
6	28.626	16.49 to 40.77	69.653	57.90 to 81.41
7	13.354	7.29 to 19.41	12.0472	8.56 to 15.54
8	32.961	21.26 to 44.66	36.377	27.52 to 45.24
9	14.867	7.19 to 22.55	16.031	9.74 to 22.32
10	21.002	10.98 to 31.02	45.641	35.45 to 55.83
11	6.689	0.00 to 13.47	23.236	16.31 to 30.17
12	9.106	2.57 to 15.65	41.936	32.85 to 51.03

Figure 1 and Figure 2 displays 95% confidence intervals and the range of the equity values against the allocations set by the RA and the GIA.

Figure 1 show the scenario for design point 10, where RA is better than GIA and Figure 2 show the scenario for design point 1, where GIA is better than RA, but the difference between the confidence interval is very small.

Figure 1: GIA vs. RA Allocation Strategies Confidence Interval of Design Point 10

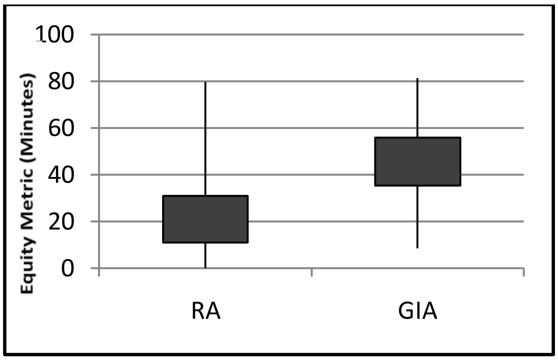
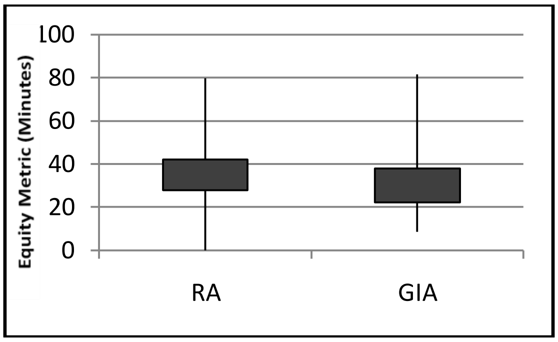


Figure 2: GIA vs. RA Allocation Strategies Confidence Interval of Design Point 1



VI. Related Work

There are a few papers that deal with the voting machine allocation problem in order to avoid long lines for voters and provide equity.

The first one that we developed our research using it is [4], where they proposed a simulation-based Greedy Improvement Algorithm (GIA) to generate machine allocations to provide equitable voting experiences to all voters so that no one particular group of voters is disadvantaged or disenfranchised, they used the average absolute differences of waiting times across all precincts as a performance metric for equity, so did we.

Then there are other papers that research the allocation of voting machines problem which are [5], [1], and [2]. The two papers [5] and [2] use simulation for their models and this allow them to consider some realistic complications in the model like voting-machine failures and uncoordinated voter arrivals, but [5] used a simple method by allocating machines in proportion to the expected number of voters at each precinct.

VII. Conclusions

The occurrence of long lines in elections depends on many unpredictable factors and is difficult to control. One way to explore this problem is through simulation which can be useful in the election process and can result in good recommendations to the allocation of DRE machines across precincts.

In our simulation model we used non-stationary voter arrivals, transient queues, and different turnout rates across all precincts.

The machine allocation RA method is proposed and illustrated using an example from the 2004 election in Franklin County, Ohio. Then it is compared to the GIA method [4].

The RA method is shown to offer potential advantages, both through the reduction of equity metric (the average absolute differences of waiting times across all precincts) and it takes smaller time in simulation compared to the GIA.

The GIA is better slightly in the case of small number of DRE machines, but the RA is significantly better than GIA in large numbers of DRE machines, which indicate that the number of machines affects the performance of machine allocation policies.

VIII. Limitations and Future Work

An incomplete list of additional topics for future study is as follows:

1) Include more realistic cases such as heterogeneous precincts to the simulation model, which have different voters-arrival patterns, and different distributions of voting times due to ballot length and nature of voter. Also considering the voting-machine failures in the model which occurs in real life.
2) Explore the elections in the developing countries such as Egypt, where the elections don't have a technological infrastructure for voting because it is not practical or present. It uses manual technique for voting, tallying, or verification of paper-ballots. These factors lead to the lack of necessary data which can be used to simulate elections in developing countries. Studying the elections there, find a way to simulate it, and applying the methods mentioned in the literature could help in strengthening the democratic process in developing countries.
3) Commercial software could be developed based on the RA or any other method on the literature. This software will help governments take decisions about election allocations that result in voters waiting the least amount of time possible and provide savings to the money spent on election process.
4) Compare proposed method to other methods in the literature such as Allocating machines to minimize the maximum expected voter waiting time across all precincts.

References

[1] Allen, T. T., and M. B. Bernshteyn. 2006a. Mitigating voter waiting times. Chance 19 (4): 25–36.

[2] Allen, T. T., and M. B. Bernshteyn. 2006b. optimal voting machine allocation analysis. In Analysis of May 2006 Primary Cuyahoga County, Ohio, ed. S. Hertzberg, 71–89. Election Science Institute. Available via http://moritzlaw.osu.edu/electionlaw/news/documents/DREAnalysisforMay2006Primary.pdf

[3] Anthony, A. A., M. F. Colley, K. E. Marinello, and C. C. Petree. 2004. Election 2004: A report to the community. Franklin County Board of Elections Report. Available via http://centralohiovoters.home.att.net/elect2004.htm

[4] Muer Yang, Michael J. Fry, W. David Kelton. ARE ALL VOTING QUEUES CREATED EQUAL? in Proceedings of the 2009 Winter Simulation Conference

[5] Edelstein, W. A. 2006. New voting systems for New York - long lines and high cost. Technical report, New Yorkers for Verified Voting Report. Available via http://www.nyvv.org/ doc/voterlines.pdf

[6] Feldman, D., and C. Belcher. 2005. Voting experience survey. In Democracy at Risk: The 2004 Election in Ohio. Democratic National Committee. Available via http://www.democrats.org/pdfs/ohvrireport/section03.pdf

[7] Houdt, B. V., and C. Blondia. 2005. Approximated transient queue length and waiting time distributions via steady state analysis. Stochastic Models 21:725–744.

[8] Confidence Interval, Wikipedia Entry: http://en.wikipedia.org/wiki/Confidence_interval

[9] Egyptian Constitutional Referendum, 2011, Wikipedia Entry: http://en.wikipedia.org/wiki/Egyptian_constitutional_referendum,_2011

[10] React.NET Discrete Event Simulation Framework, Avilable via: http://reactnet.sourceforge.net/

[11] Mebane, W. R. 2005. Timing and turnout in Ohio. Technical report, TomPaine.com. Available via http://www.tompaine.com/print/timing and turnout in ohio.php

[12] Mebane, W. R. 2006. Voting machine allocation in Franklin County, Ohio, 2004: Response to U.S. Department of Justice Letter of June 29, 2005

WORK DIVISION

Team Member	Activites
Enas Mohamed, em_cmp_eng@yahoo.com	1. Search for ideas and conferences 2. Draft of Project Proposal (State of the art, and References) 3. Search for Simulation Library 4. Understand the React.Net Library used. 5. Building the main components of the simulation model. 6. Implementation of the RA method. 7. Run different iterations of the simulation and collect output 8. Write the Final Paper (Related Work, Suggested Solution) 9. Write Poster Presentation
Hesham Naiem Mamoun, Hesham.naiem@yahoo.com	1. Search for ideas and conferences 2. Draft of Project Proposal (Target Conference, Motivation, and Potential Contribution) . 3. Search for Simulation Library 4. Understand the React.Net Library 5. Search for Data to be used in simulation. 6. Implementation of the GIA method. 7. Run different iterations of the simulation and collect output 8. Write the Final Paper (Suggested Solution, Performance Measurement and Analysis, Conclusions, Limitations and Future Work) 9. Write Poster Presentation
Mostafa Mohamed Izz, Mostafa.3ez@gmail.com	1. Search for ideas and conferences 2. Draft of Project Proposal (Problem Statement, and Proposed work and plan) 3. Study Arena (Simulation modeling software) 4. Understand the React.Net Library 5. Responsible for using svn through google code. 6. Responsible for creating data using distributions by the library. 7. Run different iterations of the simulation and collect output 8. Preparing the document template of the conference. 9. Write the Final Paper (Abstract, Introduction, Motivation, Problem Definition) 10. Present Poster Presentation