Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

ARE ALL VOTING QUEUES CREATED EQUAL?

Muer Yang
Michael J. Fry
W. David Kelton

Department of Quantitative Analysis
and Operations Management
University of Cincinnati
Cincinnati, OH 45221, U.S.A.

ABSTRACT

Providing equitable voting experiences across voting precincts has been noted as an important goal in elections. We seek to provide equity to all voters so that no one particular group of voters is disadvantaged or disenfranchised. This paper uses the average absolute differences of waiting times across all precincts as a performance metric for equity. A simulation-based greedy improvement algorithm is proposed to generate machine allocations. We examine our allocation method using a factorial experimental design, and we conclude that our heuristic outperforms the utilization-equalization method which was used by at least one county in the 2008 presidential election.

1 INTRODUCTION

Election administration is a service provided to the voting public. As with other service systems, waiting lines are non-value-added activities and waiting time is viewed negatively. As discussed in Mebane (2005), waiting times to vote are directly correlated to voter turnout. 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 new voting machines such as direct-recording electronic (DRE) machines (also known as “touch-screen” machines), election boards are limited in their ability to procure additional equipment.

Perhaps even more importantly, voting systems should provide equity, which means that we should not design systems that favor some voting groups (defined by geography, voting preference, etc.) by having shorter lines in some precincts than others. Such inequities have been a concern in recent elections. The Department of Justice investigated claims that the Board of Elections in Franklin County, Ohio, “systematically assigned fewer voting machines in polling places serving predominantly black communities as compared to its assignment of machines in predominantly white communities” during the 2004 election (Tanner 2005). While the Department of Justice ultimately found no evidence of systematic decisions to create voting inequities, their report does point to many factors, including specific measures of voter wait times and differences in voting patterns across precincts, which are typically not taken into account under current voting-machine allocation policies. 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” (Mebane 2006).

Determining an optimal voting-machine allocation is challenging and difficult to solve for several reasons. (1) Voters arrive randomly and according to non-stationary processes to polling locations. There are typically surges in voter arrivals during the morning, noon and evening times due to work schedules (Edelstein 2006). Moreover it is difficult to estimate voter turnout rates prior to Election Day because it depends on many uncontrollable variables (e.g., weather, composition of the voting ballot, etc.). (2) Voter queues may not reach steady state. Ohio law requires that the polls be open 13 hours, plus however much time is needed to accommodate voters waiting to vote at 7:30 pm. Given the limited amount of time that voting precincts are open, the voting queues may still be in a transient state. Such non-stationary arrivals and non-steady-state

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

queues violate the fundamental assumptions of traditional queueing theory. (3) Actual voting scenarios involve considerable computational complexity. There are thousands of polling stations across Ohio and the input variables are stochastic. The result is a large-scale non-linear stochastic optimization problem. Thus, both building model and developing solution methods are challenging endeavors.

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 a greedy improvement heuristic. The objective in our machine allocation is to provide voter equity across precincts. The rest of the paper is arranged as follows. Section 2 provides a review of related literature. Section 3 introduces a performance metric for voter equity and discusses several analysis options for this problem. Section 4 describes the setup of the simulation model and the greedy improvement algorithm (GIA) implemented on the model. Section 5 systematically studies the performance of the simulation-based GIA through an experimental design. Section 6 presents conclusions and future work.

2 LITERATURE REVIEW

There are relatively few papers that apply operations-research models to the voting-machine-allocation problem. However, this problem is closely related to queueing-model applications and resource-allocation problems that are common in operations research. Here we will mention only those papers that are most closely related to our application of voting- machine allocation or that reference solution methods we use directly.

The only papers of which we are aware that apply operations research to the voting-machine-allocation problem are Edelstein (2006), Allen and Bernshteyn (2006a) and Allen and Bernshteyn (2006b). Edelstein (2006) and Allen and Bernshteyn (2006b) use simulation for their models. Using simulation allows these authors to consider some of the realistic complications in their models including voting-machine failures and non-stationary voter arrivals. However, neither paper explicitly considers voter equity in terms of maintaining equivalent waiting times across precincts. Allen and Bernshteyn (2006a) suggest using queueing models to measure voter waiting times for given machine-allocation policies and to improve allocation decisions. They use simple analytical queueing models to predict average waiting times for voters. Allen and Bernshteyn then suggest an optimization model that uses a minimax objective function to allocate voting machines. Specifically, they suggest allocating machines to minimize the maximum expected voter waiting time across all precincts. The minimax objective is designed to promote voter equity as we discuss above, but there are many other objectives that could be considered. Allen and Bernshteyn (2006a) also do not consider complicating issues such as non-stationary voter arrivals, machine failures, and specific differences in voting-time requirements due to differences in ballot lengths. Furthermore, the authors propose only simple greedy-heuristic solution methods for their models, which can produce significantly suboptimal policies.

There are several simpler methods used to allocate voting machines to precincts that have been used in previous elections. An intuitive and simple method of allocating voting machines used by many election boards is to allocate machines in proportion to the expected number of voters at each precinct (Edelstein 2006). This method ignores any direct models of queueing effects and differences between precincts. At least one county in Ohio used a utilization equalization allocation policy in the 2008 presidential election to allocate voting machines. This method enforces voter equity by equalizing the utilization of voting machines rather than equalizing waiting times of voters. Moreover, the utilization rate is obtained by traditional queueing theory, which assumes stationary arrivals and steady-state operating conditions.

3 ANALYSIS OPTIONS

3.1 Performance Metric

Simply minimizing total expected waiting time across all precincts is insufficient as this may allow long voter waiting times in some precincts in order to decrease voter waiting time in other precincts. This is undesirable in an election process as we seek to provide equity to all voters so that no one particular group of voters is disadvantaged or disenfranchised.

However, there is no universal way to interpret "equity." The ideal case is that the expected waiting time in queue at every precinct is the same. But it is generally not feasible to achieve this ideal situation. Therefore, the following metric (the average absolute differences of expected waiting times among precincts) can be used as a proxy for "equity:"

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

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

Thus, the allocations that provide the best "equity" are the global optimal solutions to the following optimization problem:

$$\min\{Z(X) | X \in \mathscr{S}\},$$

where $\mathscr{S}$ is the set of feasible solutions, and $|\mathscr{S}|$ is finite.

3.2 Simulation Model vs. Analytical Model

One of the most difficult problems in dealing with voting queues is that voters do not arrive according to a stationary arrival process. Moreover, steady-state may not be achieved in an actual election where voting occurs over a single day and queues begin empty.

Analytical queuing models require strong simplifying assumptions (such as stationary arrivals, steady-state queues, etc.) about the voting system. These models enable us to obtain insights and generate metrics such as expected waiting times very quickly without dedicated simulation software. Moreover, closed-form queueing-model formulas can be used in conjunction with optimization models to determine optimal policy decisions. As an example of the insights offered by such models for this application, an integer-programming-based solution method for this problem using M/M/s closed-form queueing equations shows that voter equity may be compromised if all available voting machines are allocated. The optimal solution to maximize voter equity (as described by (1)) in some scenarios is to not allocate all available voting machines. While this is an interesting, and potentially useful, insight, solving realistically sized problems through an integer program is not generally feasible. Thus, our solution methods described in this paper rely on simulation and heuristic search techniques.

Because of the short time frame of an actual election day, analytical results for the voting-machine-allocation problem require transient queueing analysis with non-stationary arrivals. Obtaining transient information is generally considered much more complicated in comparison to a steady state analysis (e.g., Houdt and Blondia 2005). Roughly speaking, two main approaches have been developed to obtain transient distributions. The first method relies on numerically inverting the Laplace transform or generating function involved (Choudhury, Lucantoni, and Whitt 1994; Hofkens, Spaey, and Blondia 2004; Lucantoni, Choudhury, and Whitt 1994). The second method is based on recursive computations. Others (Ny and Sericola 2003; Lee and Li 1990) combine uniformization techniques to reduce the problem to discrete time and afterward apply a recursive algorithm. Although these methods are effective in obtaining transient distributions related to the initial system behavior, their computational costs grow rapidly when considering later events. Moreover, most of the literature considers only single-server queues. Such limitations of the current analytical results on transient queues weakens the advantages of analytical models, which become more difficult to implement and needs more computational time to obtain results.

Thus, it is natural to turn to stochastic simulation, with its lesser reliance on simplifying assumptions that might render the model questionable in terms of validity. However, we then need to apply proper statistical design and analysis methods in order to deal with uncertainty in the output, and to enable valid and precise conclusions.

4 SIMULATION ANALYSIS

4.1 Description of Basic Polls Queueing Model

4.1.1 Model Logic

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

On election day, all polling stations in Ohio open at 6:30 am and close at 7:30 pm. Once a voter arrives to his or her designated precinct, the voter joins a single queue until there is a DRE voting machine available. There are multiple DRE voting machines per precinct. We assume that all DRE voting machines are identical and shared by all voters within a polling station.

4.1.2 Input-Distribution Assumptions and Data Sources

We use a data set based on statistics from the 2004 election in Franklin County, Ohio (available at <http://copperas.com/fcelection/>). Specifically, we fit a normal distribution with mean 1070 and standard deviation 319 to the number of registered voters in each precinct in the 2004 election. For a given number N of precincts (N will be set as a factor in our experimental design below), we generate the number of eligible voters for each precinct independently from this fitted normal distribution. The turnout rate is applied to this, and then the non-stationary arrival pattern is used to distribute these voters' appearance at the polling station, as described below. We assume that each precinct has the same voter turnout rate.

Ohio Revised Code (O.R.C.) §3501.32 states that on election day the polls shall be opened at 6:30 am and shall be closed at 7:30 pm "unless there are voters waiting in line to cast their ballots, in which case the poll shall be kept open until such waiting voters have voted." Ohio law thus requires that the polls be open 13 hours, plus however much time is needed to accommodate voters waiting to vote at 7:30 pm. Therefore, we allow all queues to clear, but we do not allow any additional voter arrivals after 7:30 pm in our simulations.

On election day, there are "peaks and valleys" of usage by voters depending upon the time of day, the weather, traffic and other variables outside of the control of election staff. Voters do not arrive according to a stationary arrival stochastic process. There are typically surges in voter arrivals during the morning, noon, and evening due to work schedules (Edelstein 2006). Precinct poll workers at the Hamilton County, Ohio, Board of Elections reported an early morning voter rush and lines shorter by day's end on election day 2008. The Voting Experience Survey (Feldman and Belcher 2005; Mebane 2005), 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 surges may not be the same across all precincts due to differences in voters' socioeconomic status, but here we assume that all precincts experience similar arrival patterns.

Table 1: Voter Arrivals by Time of Day

Periods 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

Ohio law states that voters are allowed up to five minutes to place their vote (Anthony et al. 2004). However, anecdotal evidence suggests that this law is rarely, if ever, enforced. Actual voter service times will depend on the length of the ballot - in particular, the number of issues on the ballot, which generally require the most time for voters to read and on which to make a choice. To determine approximate voter service times we use data from the 2006 Ohio gubernatorial election with six issues on the ballot. We fit a distribution to the data of voting times read from a sample of Election Systems & Software machines in this election and found that a gamma distribution with the scale parameter of 1.05 and the shape parameter of 5.71 fits the data acceptably. We assume that voting times in every precinct follow this same gamma distribution.

For now, we assume that voting machines are perfectly reliable (i.e., there are no voting-machine failures). We also assume that all available voting machines must be allocated to precincts as this is often the policy used in practice, and will reduce the system-wide voter wait time.

4.2 The Greedy Improvement Algorithm

4.2.1 Algorithm Description

We use a heuristic solution method to allocate voting machines to precincts. Our method combines a simple greedy heuristic to reduce expected voter wait times with a local improvement search where we use (1) as our objective function. We refer to this solution method as the greedy improvement algorithm (GIA), which consists of two phases: Phase I is a simple greedy heuristic and Phase II is a local improvement search.

Phase I: The simple greedy heuristic iteratively allocates a voting machine to the precinct with the largest estimated expected waiting time. The steps are as follows,

Step 1. Assign initial values to x_i for each precinct i (usually we set $x_i = 1$ for all i). Set Counter = $\sum_{i=1}^{N} x_i$.

Step 2. Let $x_i = x_i + 1$ for the precinct i with the largest estimated expected waiting time in queue, $W_i(x_i)$.

Step 3. Counter = Counter + 1.

Step 4. If Counter = N, stop. Otherwise, go to step 2.

Phase II: To improve the performance of the simple greedy heuristic, we introduce a local search component, which is shown as follows.

Step 1: Set the allocation obtained in Phase I as the initial allocation, $X_0 = (x_{01}, x_{02}, ..., x_{0N})'$. Let $X^* = X_0$, and $Z(X^*) = Z(X_0)$, where $Z(X)$ is given by (1).

Step 2: Given $X^* = (x_1^*, x_2^*, \ldots, x_N^*)$, select $X' \in argmin_{X \in N(X^*)} Z(X)$, where $N(X^*)$ is the neighborhood of X^*, $N(X^*) = \{X \in \mathscr{S} : \exists i, j \in \{1, ..., N\}, i \neq j, s.t. x_i = x_i^* + 1, x_j = x_j^* - 1, \forall k \in \{1, ..., N\} \backslash \{i, j\}, x_k = x_k^*\}$.

Step 3: If $Z(X') < Z(X^*)$, let $X^* = X'$, and go to Step 2; otherwise, stop.

4.2.2 Implementation in Arena

It is not difficult to implement the GIA in a standard programming language (e.g., C, VBA, and C++) once the expected wait times in any precinct $W_i(x_i)$ are obtained. However, extensive effort is required to build simulation models to obtain the expected wait times using standard programming languages. On the other hand, it is easy to obtain $W_i(x_i)$ in simulation-modeling software (e.g., Arena), but difficult to implement the GIA, because simulation-modeling software is not designed as a programming language.

To implement the simple greedy algorithm in Arena, the model needs to run multiple replications under each scenario (where a particular scenario corresponds to a given allocation of machines to precincts) to calculate the average waiting times for every precinct. Once average waiting times are computed for a particular scenario, we identify the precinct with the largest waiting time and allocate one more machine to that precinct. Additional replications are run for the new scenario and average waiting times are computed. This process repeats until all available voting machines are allocated.

Because Arena is a process-oriented simulation software, it is not natural for Arena to deal with the kinds of logic decisions mentioned above. For example, the inputs of a simulation model are usually set before the start of a simulation run and will not change during the entire simulation run, such as the number of replications, the capacities of resources (i.e., the number of DRE machines), etc. The estimated expected waiting times in any precinct can usually be read from the output generated by Arena after the simulation run is completed. However, the simple greedy algorithm requires that we change the number of replications and the capacities of resources during a simulation run, and to read the output to find the longest average waiting time between each scenario during a simulation run. Such technical details make it difficult to fully implement the algorithm in Arena.

To overcome these problems, we employ logical entities (i.e., exist only to serve a logical function and do not correspond to any real entities, like voters) and non-clearing variables (variables that retain their values between model runs) in Arena to perform those logical and control tasks. Before the simulation begins, a logical entity arrives at time zero to assign a current allocation of voting machines from a non-clearing vector variable of number of machines, i.e., to assign the resource capacities accordingly for this scenario. (Non-clearing means that values are retained between replications, not Arena's default since one generally wants to clear statistics between replications, but we need Arena to retain some variable values across runs in order to implement the GIA.) The total number of machines allocated in this scenario is also calculated.

Actual simulation of the voting day for all precincts is built in a general and scalable way. To change the number of precincts in the model, we need only to modify the Data/Spreadsheet modules in the Arena Basic Process and Advanced Process panels; no other parts of the model needs to be modified, nor does its topological structure.

The Arena output can only be generated after the entire simulation is finished. We cannot use it to find the longest waiting time in each scenario during a simulation run. We set up specific procedures to calculate the average waiting time for every precinct in the model and assign those values to a vector variable. At the end of each replication, another logical entity arrives. At the conclusion of the final replication the logical entity identifies the precinct with the longest waiting time from the vector variable of average waiting times, and adds a machine to that precinct. Then the logical entity checks whether there are still more machines available to allocate. If so, the internal Arena variable for the total number of replications is increased by a set value to force Arena into running additional replications; if not, the simulation terminates.

5 EXPERIMENTAL DESIGN AND RESULTS

In order to examine the performance of the simulation-based GIA and compare it with the utilization equalization method (UEM), we set up a factorial experimental design using five factors: *Turnout Rate*, *Voting Time*, *Size of County*, *Ratio of the number of machines to the number of precincts*, and *Allocation Strategy* (see Table 2). The response of this experimental design is the "equity" metric given in (1).

Table 2: Factors and Levels for Experimental Design

Factors	-	+
Turnout Rate	0.56	0.72
Voting time (Scale parameter of gamma distribution)	0.58	1.05
Size of County	20 precincts	50 precincts
#Machines/#Precincts	3.6	9
Allocation Strategy	GIA	UEM

5.1 Factors and Levels

We assume that the voter turnout rate is the same for all precincts. The overall voter turnout rate observed in the 2004 election in Franklin County, Ohio, was 56%. The turnout rate in the 2008 presidential election was the highest in at least 40 years at approximately 72%. Therefore, the two levels used for *Turnout Rate* are 0.56 and 0.72.

We assume that the voting service time in every precinct follows a gamma distribution with shape parameter of 5.71. One level for this factor is set using data from the 2006 Ohio gubernatorial election by setting the gamma distribution scale parameter to 1.05. Allen and Bernshteyn (2006a) use a mean service time of voting of 3.33 minutes. We use this as the mean for the other level of our *Voting Time* factor. We do not have data for the voting times with mean of 3.33 minutes, but we assume such voting times also follow a gamma distribution with shape parameter 5.71 and set the distribution's scale parameter to 0.58 to match this mean voting time.

We set the two levels of *Size of County* to be 20 precincts and 50 precincts. This represents smaller than usual voting districts (counties), but should provide enough of a difference for analysis.

We set the two level of *Ratio of the number of machines to the number of precincts* to 3.6 and 9 to match observed average allocation values in the 2004 and 2008 elections, respectively, in one particular county in Ohio.

To obtain the allocations for each treatment combinations of *Turnout Rate*, *Service Time*, *Size of County* and *Ratio of the number of machines to the number of precincts*, we run the GIA in Arena. 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. UEM is based on M/M/s queues and implemented in Microsoft Excel, which can easily find the machine allocations for each treatment combination of the other four factors.

There are $2^5 = 32$ design points in total (see Table 3). We run 100 replications for each design point, which satisfied our requirement that the 95% confidence-interval half widths are less than 5% of the average values of the response. Different random number streams are used for different design points to ensure that the design points are independent.

5.2 Results

The results of the factorial experimental design (see Table 3) show that the GIA statistically significantly outperforms the UEM in 14 out of 16 different treatment combinations of the other four factors, ties in one scenario (*Turnout Rate* = 0.56, *Scale Parameter* = 1.05, *Size of County* = 20 precincts and *Ratio of the number of machines to the number of precincts* = 3.6), and underperforms UEM in a single scenario (*Turnout Rate* = 0.72, *Scale Parameter* = 1.05, *Size of County* = 20 precincts and *Ratio of the number of machines to the number of precincts* = 3.6).

Figure 1 shows the scenario with *Turnout Rate* = 0.72, *Scale Parameter* = 1.05, *Size of County* = 50 precincts and *Ratio of the number of machines to the number of precincts* = 3.6. It displays 95% confidence intervals and the range of the response values against the allocations provided by the GIA and the UEM. It clearly shows that the GIA is statistically significantly better in this scenario.

In the only scenario where the GIA underperforms, the average absolute differences of waiting times across all precincts is 78.73 minutes for the allocation provided by the Greedy Improvement Algorithm, only 3 minutes more than the average absolute differences of waiting times across all precincts for the allocation provided by the UEM. Thus, generally speaking, GIA generates better allocations than the UEM.

Figures 2 to 5 contain interaction plots of *Allocation Strategy* vs. the other four factors (on horizontal axes) respectively, where the "equity" metric (given in (1)) is on vertical axes. Figure 2 shows that the effects of *Turnout Rate* are additive. There is no interaction effect between *Turnout Rate* and *Allocation Strategy*. The allocations provided by the GIA are consistently better than the UEM, regardless of the turnout rate.

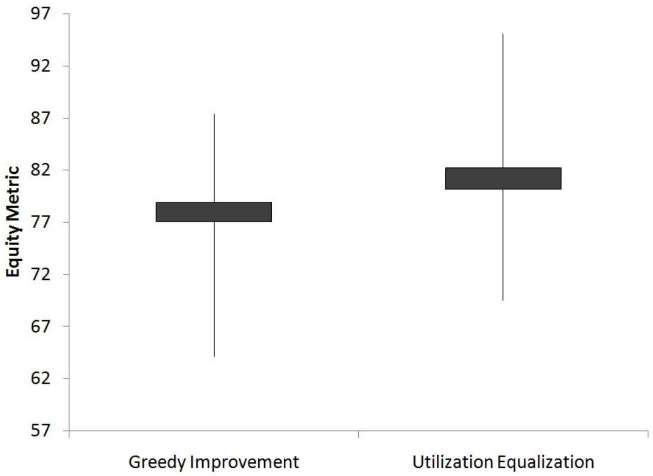

Figure 1: Greedy Improvement vs. Utilization Equalization Allocation Strategies

The interaction plot in Figure 3 shows that there is an interaction effect between *Voting Time* and *Allocation Strategy*. Although the allocation provided by the GIA performs better than the UEM regardless of level of *Voting Time*, the difference between the two allocations becomes smaller as the voting time increases.

The interaction effects between *Size of County* and *Allocation Strategy* are clearly shown in Figure 4. The allocations provided by the GIA and the UEM are very close in terms of response values when the size of county is small (20-precinct county). However, as the size of county increases, the differences between the two allocations provided by the two different methods become larger. The GIA performs much better than the UEM in large counties. In practical applications, the GIA is expected to provide much better allocations than the UEM, because a county usually has hundreds of precincts.

Ratio of the number of machines to the number of precincts does have interaction effects with *Allocation Strategy* (see Figure 5). The allocation provided by the GIA provides more voter equity than does the UEM, no matter whether the ratio of the number of machines to the number of precincts is small or large. However, a large amount of available machines will compromise the advantages of the allocation generated by the GIA over the UEM.

6 CONCLUSIONS AND FUTURE WORK

Providing equitable voting experiences across voting precincts is noted as an important goal in elections. We seek to provide equity to all voters so that no one particular group of voters is disadvantaged or disenfranchised. The average absolute differences of waiting times across all precincts is proposed in this paper as a performance metric for “equity.” To deal with non-stationary voter arrivals and transient queues, we propose a simulation-based greedy improvement algorithm to generate machine allocations to provide increased voter equity. It is statistically shown that this heuristic outperforms the utilization equalization method for allocating voting machines.

Turnout rate and time required by voters to cast their ballot are two factors in the voting system that are essentially impossible to control. Our experimental results indicate that our greedy improvement algorithm generates better machine allocations in terms of voter equity than does the utilization equalization method, regardless of turnout rate and voting time values.

Our results indicate that the size of a county affects the performance of machine allocation policies. The more voting precincts in a county, the better the allocation generated by the greedy improvement algorithm performs in comparison to the utilization equalization method.

Not surprisingly, a large amount of available machines reduces the advantages of the greedy improvement algorithm. As the number of available machines becomes very large, the average voter waiting time approaches zero. In such a case, the allocation method used is inconsequential. Increasing the number of DRE machines is always recommended when feasible to reduce overall system voter wait times. However, analytical results suggest that voter equity can be maximized by not allocating all available machines in some scenarios.

Table 3: Factors and Levels for Experimental Design

Design Points	Turnout Rate	Scale Parameter	Size of County (# Precincts)	#Machines/ #Precincts	Allocation Strategy	Equity Metric
1	0.56	0.583	20	3.6	GIA	5.474
2	0.72	0.583	20	3.6	GIA	20.792
3	0.56	1.05	20	3.6	GIA	61.571
4	0.72	1.05	20	3.6	GIA	78.728
5	0.56	0.583	50	3.6	GIA	6.501
6	0.72	0.583	50	3.6	GIA	23.679
7	0.56	1.05	50	3.6	GIA	62.02
8	0.72	1.05	50	3.6	GIA	77.99
9	0.56	0.583	20	9	GIA	0.007
10	0.72	0.583	20	9	GIA	0.02
11	0.56	1.05	20	9	GIA	0.334
12	0.72	1.05	20	9	GIA	2.01
13	0.56	0.583	50	9	GIA	0.007
14	0.72	0.583	50	9	GIA	0.025
15	0.56	1.05	50	9	GIA	0.409
16	0.72	1.05	50	9	GIA	2.406
17	0.56	0.583	20	3.6	UEM	6.049
18	0.72	0.583	20	3.6	UEM	23.11
19	0.56	1.05	20	3.6	UEM	60.898
20	0.72	1.05	20	3.6	UEM	75.666
21	0.56	0.583	50	3.6	UEM	9.213
22	0.72	0.583	50	3.6	UEM	29.503
23	0.56	1.05	50	3.6	UEM	64.794
24	0.72	1.05	50	3.6	UEM	81.261
25	0.56	0.583	20	9	UEM	0.013
26	0.72	0.583	20	9	UEM	0.038
27	0.56	1.05	20	9	UEM	0.42
28	0.72	1.05	20	9	UEM	1.982
29	0.56	0.583	50	9	UEM	0.017
30	0.72	0.583	50	9	UEM	0.049
31	0.56	1.05	50	9	UEM	0.53
32	0.72	1.05	50	9	UEM	2.527

Although the greedy improvement algorithm is statistically significantly better than the utilization equalization method, the absolute values of the response metric are often small. In the experimental design, the maximum difference of the average values of response between the two allocation methods is less than six minutes. The difference may increase when the size of county increases. However, it still implies that although the assumptions of M/M/s queues are violated, including non-stationary arrivals, non-exponential service time, and non-steady-state queues, the allocation methods based on such assumptions (i.e., utilization equalization method) can still generate reasonable allocations. If time is a constraint and dedicated simulation software is not available, the utilization equalization method is not a bad choice - just not as good as the greedy improvement algorithm.

Notwithstanding that the non-stationary voter arrivals and transient queues are considered in this simulation model, we assume the turnout rate is the same for all precincts and voting times in every precinct follow the same distribution. In the future, we would like to study heterogeneous precincts, which have different voter-arrival patterns, different turnout rates, and different distributions of voting times.

In addition, we also would like to study a reallocation strategy. Voting machines can be initially assigned to precincts, but a reallocation of some voting machines could then be performed in response to observed machine failures and voter turnout. Such reallocation strategies are, admittedly, challenging logistically and would be subject to restrictions due to potential

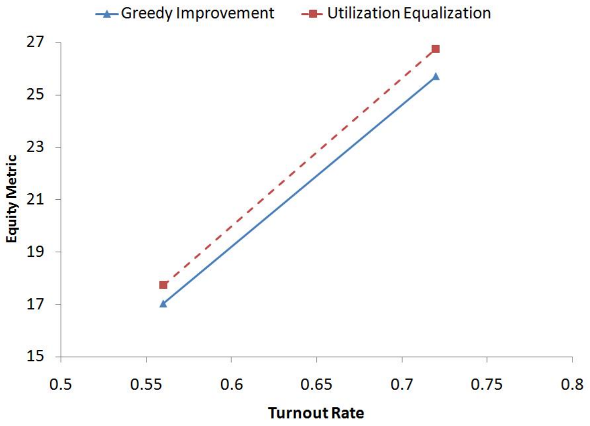

Figure 2: Voter Turnout Rates vs. Allocation Strategy

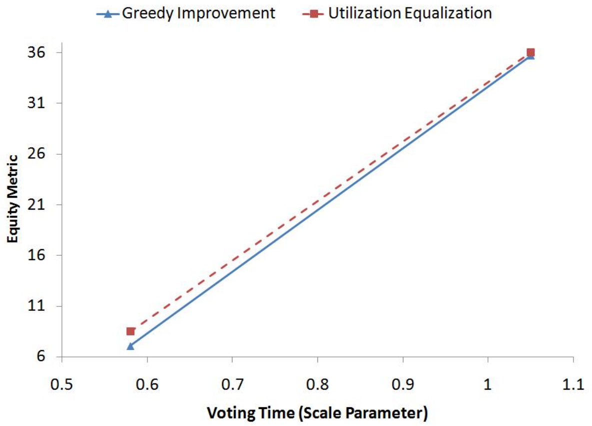

Figure 3: Voting Time Required vs. Allocation Strategy

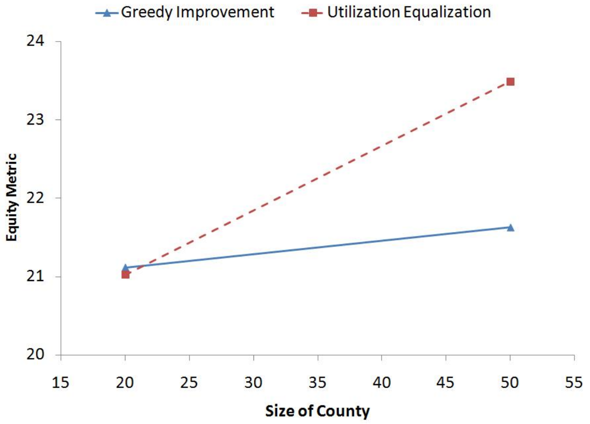

Figure 4: Size of County vs. Allocation Strategy

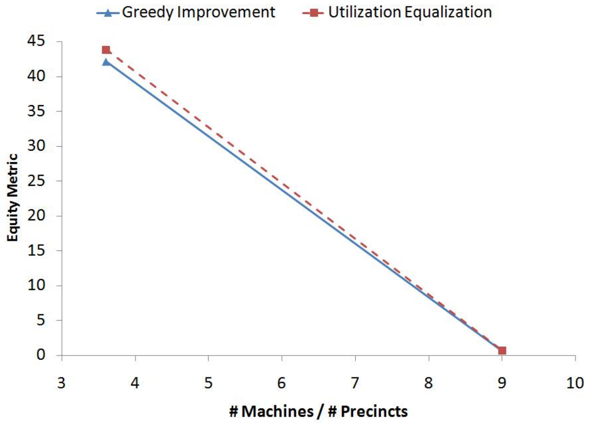

Figure 5: Machines per Precinct vs. Allocation Strategy

differences in ballot styles between precincts. However, reallocation represents a benchmark ”best-case” for recourse actions that allows us to compare the relative merits of one-time allocation policies as well as other recourse strategies.

ACKNOWLEDGMENTS

We would like to thank Professor Ted Allen for providing voter arrival data. We also thank Professor Jeff Ohlmann for his input on search heuristics and early feedback on our work on this problem.

REFERENCES

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

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> [accessed March 31, 2009].

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>[accessed March 31, 2009].

Choudhury, G. L., D. Lucantoni, and W. Whitt. 1994. Multidimensional transform inversion with applications to the transient M/G/1 queue. *Annals of Mathematical Statistics* 4:719–740.

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> [accessed March 31, 2009].

Feldman, D., and C. Belcher. 2005. Voting experience survey. In *Democracy at Risk: The 2004 Election in Ohio*. Democratic National Committee. Available via <http://a9.g.akamai.net/7/9/8082/v001 www.democrats.org/pdfs/ohvrireport/section03.pdf> [accessed March 31, 2009].

Hofkens, T., K. Spaey, and C. Blondia. 2004. Transient analysis of the D-BMAP/G/1 queue with an application to the dimensioning of a playout buffer for VBR traffic. In *Proceedings of Networking*, 184–191. Athens, Greece.

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

Lee, D. S., and S. Q. Li. 1990, March. Transient analysis for two-state markov modulated poisson arrival queues with overload control. In *Proceedings of the 23rd Annual Conf. on Information Sciences and Systems*. New Jersey: Princeton University.

Lucantoni, D. M., G. L. Choudhury, and W. Whitt. 1994. The transient BMAP/PH/1 queue. *Stochastic Models* 10:461–478.

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>[accessed March 31, 2009].

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

Ny, L. L., and B. Sericola. 2003. Transient analysis of the BMAP/PH/1 queue. *International Journal of Simulation* 3 (3): 4–14.

Tanner, J. 2005. Letter from U.S. Department of Justice, Civil Rights Division to Nick Soulas, Jr., Assistant Prosecuting Attorney of Franklin County, Ohio.

AUTHOR BIOGRAPHIES

MUER YANG is a PhD candidate in the Department of Quantitative Analysis & Operations Management in the College of Business at the University of Cincinnati. He received his bachelor's degree in management information systems in 2004 and his master's degree in management science in 2006, both from Tsinghua University in Beijing, China. His email is <yangmr@email.uc.edu>.

MICHAEL J. FRY is an associate professor in the Department of Quantitative Analysis & Operations Management in the College of Business at the University of Cincinnati. Professor Fry received his Ph.D. in industrial and operations engineering from the University of Michigan in 2002. Professor Fry has worked with a variety of businesses and organizations in applying operations research techniques to public policy operations. He has worked with the Hamilton County Board of Elections in developing efficient delivery strategies for optical-scanning voting machines to reduce operational costs. Professor Fry publishes his work in the leading journals and he has been awarded both teaching and research awards at the University of Cincinnati. His email address is <mike.fry@uc.edu>.

W. DAVID KELTON is a professor in the Department of Quantitative Analysis & Operations Management at the University of Cincinnati. His research interests and publications are in the probabilistic and statistical aspects of simulation, applications of simulation, statistical quality control, and stochastic models. He is co-author of *Simulation with Arena*, and was also coauthor of the first three editions of *Simulation Modeling and Analysis*, both for McGraw-Hill. He was Editor-in-Chief of the *INFORMS Journal on Computing* from 2000-2007. He served as WSC Program Chair in 1987, General Chair in 1991, was on the WSC Board of Directors from 1991-1999, and is a Founding Trustee of the WSC Foundation. He is a Fellow of both INFORMS and IIE. His email is <david.kelton@uc.edu>.