Computers & Industrial Engineering 140 (2020) 106243

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Determining resource requirements for elections using indifference-zone generalized binary search

Theodore T. Allen[a,*], Muer Yang[b], Shijie Huang[c], Olivia K. Hernandez[a]

[a] Integrated Systems Engineering, The Ohio State University, Columbus, OH 43210, United States
[b] Opus College of Business, University of St. Thomas, Saint Paul, MN 55105, United States
[c] Business Consulting Group, Carilion Clinic, Roanoke, VA 24014, United States

ARTICLE INFO

Keywords:
Multiple comparisons
Comparison with a standard
Ranking and selection
Indifference-zone
Simulation-optimization

ABSTRACT

Election lines are more than a nuisance. In recent elections, needing to wait in lines deterred hundreds of thousands from voting and likely changed the winner in multiple cases. Part of the challenge is that even after the voter reaches the front of the line in some locations, it can require more than ten or twenty minutes to cast a ballot. Moreover, the ballot in one precinct might be twice the length of another precinct's in the same county because of additional referenda and levies. We consider the decision problem faced by election officials and other leaders: How many resources (poll books, workers, booths, or voting machines) should be allocated to each location so that lines can reasonably be expected to be minimal? We formulate a simulation optimization problem to identify the combinations that minimize resource requirements while guaranteeing acceptable user-defined service levels. We propose an Indifference Zone Generalized Binary Search (IZGBS) method with rigorous assurances on solution quality, and demonstrate it using the 2016 presidential election. In that election, we describe how the methods helped to reduce waiting times by more than three hours in at least one location affecting thousands of voters and likely increasing turnout.

1. Introduction

In the United States, voting is arguably the most important right, the right from which all other rights flow. However, long waiting lines experienced by voters in recent elections (e.g., Norden, 2013; Stewart & Ansolabehere, 2015) have challenged this fundamental right. For example, some voters in Ohio waited over 10 h to cast their ballot in the 2004 presidential election, with the last recorded ballot cast at 4:00 am the next day (Cohen, 2008). Voters at some precincts in Virginia waited several hours during the 2012 presidential election (Walker, 2013). Voters' wait times in Florida could be nearly six hours long during the early voting period in 2012 (Pastrana, 2012). Voters in Maricopa County, Arizona, could wait for as long as five hours in the primary election in 2016 (CBS News, 2016). Some voters across the country, such as in Georgia, Texas, and Arizona, suffered hours-long lines during the 2018 presidential election (Cassidy, Long, & Balsamo, 2018).

Unfortunately, long voting lines may "ultimately undermine the confidence that citizens have in the electoral process" (Stewart III & Ansolabehere, 2013), because they can result in voter disenfranchisement. Ury v. Santee (303F. Supp. 119, 1969) found that the voters were "effectively deprived of" their voting rights by forcing them to have long wait times and therefore a re-election was ordered in the Village of Wilmette, Illinois. Alvarez et al. (2008) estimated that long lines in the 2008 presidential election prevented 2.6 million voters in the nation from voting. Spencer and Markovits (2010) estimated that 1.89% of voters were disenfranchised by long lines during the 2008 primary election. Long voting lines turned away more than 200,000 voters in Florida in the 2012 general election (Powers & Damron, 2013). Highton (2006) discovered that voter turnout rates would be lowered by 10% for every additional 100 registered voters per voting machine.

Insufficient voting resources (including voting machines, poll books, and poll workers) and unscientific allocation strategies have caused long voting lines (Allen, 2013; Allen, 2014; Allen & Bernshteyn, 2006; Erhardt, Huang, Huerta, & Allen, 2015; Stewart & Ansolabehere, 2015). In addition, twenty-five states in the United States have placed new restrictions on voting since 2010 to make it harder to vote, and fifteen states have more restrictive voter ID laws (Brennan Center for Justice, 2019). These new laws lengthen the time for voters to check in or register on Election Day (e.g., Stein et al., 2019), which exacerbates the problem of insufficient voting resources, worsening the long-line

* Corresponding author.
E-mail addresses: allen.515@osu.edu (T.T. Allen), yangmuer@stthomas.edu (M. Yang), shuang@carilionclinic.org (S. Huang), hernadez.127@buckeyemail.osu.edu (O.K. Hernandez).

https://doi.org/10.1016/j.cie.2019.106243
Received 3 July 2019; Received in revised form 12 November 2019; Accepted 20 December 2019
Available online 23 December 2019

problem. For example, wait times in New Hampshire during the 2012 election increased by 43% when the voter ID law was partially implemented (The Pew Charitable Trusts, 2014).

The common method used in practice is to allocate voting machines proportionally to the total number of registered voters in a precinct. For instance, Ohio stipulates that counties must have one DRE voting machine for every 175 registered voters. New York requires that a maximum of 800 registered voters should be allocated to one voting machine. Queueing theory has concluded that wait time is affected by both the arrival process and the service process. Voters' turnout rates are likely to be different among precincts. The number of issues on the ballot often varies across precincts, which affects how long it takes for a voter to cast a ballot. Voting machine breakdowns further complicate the problem. Clearly the proportional allocation method, which does not consider these sources of variation in voting queues, is questionable.

Counties typically have hundreds of precincts and the voting resource allocation is, of course, also an issue of fairness (Bertsimas, Farias, & Trichakis, 2011; Bertsimas & Gupta, 2011; Gini, 1912; Kozanidis, 1991; Marsh & Schilling, 1994; Yang, Fry, Kelton, & Allen, 2014; Young, 1995). Unfortunately, many studies have found that long lines disproportionately affect African American and Latino voters in urban areas (e.g., Famighetti, 2016). Therefore, there is a critical need for rigorous and equitable methods to allocate voting resources so that voters' wait times can be acceptable.

First, we need to select appropriate wait-time measures to evaluate voting queues. Although average wait time is the most common measure, it can mask the real problem. Extremely long waits could still occur even if the average wait time across all voters is short. It is estimated that 16.6% of voters nationwide waited more than 30 min during the 2008 presidential election (The Marist poll, 2008), and 12.5% of voters waited at least half an hour on Election Day in 2012 (Stewart III & Ansolabehere, 2013). We should therefore include measures of the order statistics, the percentile wait time and the expected maximum wait time at a polling station.

Second, we need to define how long would be considered as an "acceptable" wait time. South Dakota Codified Laws § 12-14-4 deems more than 30 min waiting time "unreasonable." New York Code Election Law stipulates that voter wait time at a poll site should not exceed 30 min. The Schaefer Center for Public Policy (2014) recommended that voters should wait less than 30 min in Maryland. The Presidential Commission on Election Administration (2014) reached a conclusion that no voter in America should wait more than 30 min to cast their ballots, a goal underscored by President Obama in his 2012 inauguration speech. Thus, we take 30 min as the standard of acceptable wait, and restate the problem as determining the minimum resource combinations such that the expected longest waiting time of voters (i.e., an estimator of "order statistic") to be less than or equal to 30 minutes, or expected p% of voters would wait no more than 30 min.

In this paper, we propose a computationally-efficient and rigorous simulation-optimization method to allocate resources to guarantee acceptable wait times in elections (i.e., to satisfy a given queue performance level) with proven probability bounds, which is particularly applicable in cases where the standard is written into law. Our proposed method, which addresses a class of problem where the system performance level is non-decreasing in the amount of resources and can only be evaluated via simulation, has wide applicability. For example, healthcare policy makers and administrators might want to know how many doctors are needed so that veterans wait less than thirty days on average for an appointment, or how many resources are needed to ensure that transplant recipients can wait fewer than a certain number of days (Akan, Oguzhan, Baris, Fatih, & Adnan, 2012). Homeland security officials might want to determine resource levels such that no one expects to wait more than thirty minutes at the border or the airport.

The remainder of this article is organized as follows. Section 2 reviews related literature. Section 3 introduces the notation and assumptions. In Section 4, methods are proposed to derive the minimum number of resources to achieve a standard service level for cases involving a single or multiple resource types. Guarantees relating to the solution quality are also provided. In Section 5, the real-world application of the methods in central Ohio in 2016 is described. Numbers of resources needed to guarantee expected waiting times of less than 30 min for general election problems are provided. Finally, Section 6 presents conclusions and opportunities for future work.

2. Literature review

Queueing formulas have been applied to voting queues (e.g., Allen & Bernshteyn, 2006; Yang et al., 2014). Despite its simplicity and transparency, queueing formula may not be a good approximation to real voting queues because it cannot address non-stationary arrivals and non-steady state. Moreover, queueing formula tends to underestimate the number of necessary voting machines for large polling locations in practice (Yang et al., 2014). As mentioned earlier, average wait time is not an appropriate measure and the expected maximum wait time, order statistics, or percentile wait time should be considered instead. Queueing formula, however, cannot provide such queue performance measures.

Discrete event simulation models, on the other hand, can capture many complex characteristics of queues, including voters arriving at different rates during the day, and can estimate the expected maximum wait time, order statistics, and percentile wait time. Many previous research studies on voting queues have used simulation models (e.g., Edelstein & Edelstein, 2010; Grant, 1980; Herron & Smith, 2016; Kumar, Yang, & Goldschmidt, 2018; Yang et al., 2014; Yang, Wang, & Xu, 2015). Yet, the random errors in discrete event simulation make it difficult for deterministic integer programming to generate optimal or even good solutions (Yang, Allen, Fry, & Kelton, 2013; Yang et al., 2014).

With hundreds or even thousands of precincts, running large numbers of discrete event simulations can be computationally prohibitive. For example, previous simulation optimization methods required 24 h of run time to achieve a recommended voting machine allocation (Yang et al., 2013, 2014) and still offered no rigorous guarantee about a prespecified performance objective (e.g., a certain percentage of voters expect to wait no more than 30 min). Besides, these methods can only allocate a single type of voting resources (e.g., voting machine or voter check-in station). In fact, election officials usually need to determine both the number of voter check-in stations and the number of voting machines (or booths) at a polling location. Therefore, more efficient and comprehensive methods are needed to enable officials to use them routinely.

In recent years, there has been considerable effort to determine the best alternative system in terms of maximum (or minimum) expected performance by applying ranking and selection (R&S) algorithms (Chen, Yucesean, & Chick, 2000; Chick & Inoue, 2001; Frazier, 2014; Hunter & Pasupathy, 2013; Nelson & Goldsman, 2001; Ni, Ciocan, Henderson, & Hunter, 2017), particularly when the number of alternatives is small. These R&S methods are often considered easy to implement and computationally efficient (Bechhofer, Goldsman, & Santaner, 1995). The "Fully Sequential Procedure" (FSP) from Kim and Nelson (2006) has been found to be efficient in terms of sample size in contrast to traditional "comparison with a standard" procedures (Chen, 2006). An input to this procedure specifies an "indifference-zone" as the smallest difference that is critical to be detected; alternative systems are eliminated once a certain confidence is achieved. Later, Andradóttir and Kim (2010) modified the FSP procedure and applied it to problems where stochastic constraints are present. However, the methods considered assume that every alternative system will be simulated (at least an initial number of observations) and terminate with a single "best" system. In relation to finding the minimum resource level, evaluating all possible resource levels is prohibitively expensive and unnecessary. For example, if six voting machines are found to be insufficient, then there is no need to evaluate any systems with five (or any number less than six) voting machines.

In this paper, we seek to identify efficiently all solutions that minimally satisfy a given performance objective. We formulate the allocation decision problem as a type of Pareto simulation optimization in which the numbers of resources of each type are minimized (Amaran, Sahinidis, Sharda, & Bury, 2016; Gao, Chen, & Shi, 2017). Lee, Chew, Tseng, and Goldsman (2010) proposed efficient and rigorous simulation optimization methods to enumerate Pareto sets. However, their methods provide asymptotic results and we seek finite sample-based guarantees. Also, they do not exploit the assumption that additional resources cannot, in our simulation models, harm performance ("non-decreasing assumption"). We leverage the non-decreasing assumption to seek more efficient methods that avoid the need to evaluate all possible systems.

Some researchers have considered effective enumerative methods exploiting the non-decreasing assumption with noisy system evaluations that use so-called "Generalized Binary Search" (GBS) (Dasgupta, 2005; Nowak, 2011). Unfortunately, the conditions of the theorems relating to GBS are not easily met when discrete event simulation is used. This follows because simulation cannot easily guarantee successful feasibility declarations with a bounded probability. Comparing a system mean to a fixed standard, the mean can be close or equal to the standard such that a large or even infinite number of evaluations could be needed to prove the mean is above or below the standard. As a result, for election problems, simulation-based finite sample guarantees are generally only possible with an indifference-zone parameter, δ. Another objective of this research is to extend GBS so that it is relevant to simulation optimization or, more generally, experimentation with continuous responses.

3. Notations and assumptions

In this section, we define our notation and assumptions.

3.1. Notation and model

Consider a system with m different resource types. For example, with $m = 2$ we might have poll books for voter check-in and voting machines. Then, vector $\mathbf{x} = \{x_1, \cdots, x_m\}$ denotes the set of resource levels as our decision variable, and $Y_j(\mathbf{x})$ represents the system with $\mathbf{x}$ resources from the jth replication. Here, we consider only systems for which empirical evaluation is needed to evaluate $Y_j(\mathbf{x})$ such as a simulation estimate of the mean waiting time of the voter who waits the longest. Then, the expected performance of the system is $\mu(\mathbf{x}) = E[Y_j(\mathbf{x})]$, and we seek to enumerate the smallest combinations of resources $\mathbf{x}$ to satisfy a given performance level of μ_0, i.e., $\mu(\boldsymbol{x}) \le \mu_0$. If $m \ge 2$, trade-offs are expected among different types of resources. The multiple criteria problem (Steuer & Na, 2003) can be formulated as below:

$$\min\{x_1, \cdots, x_m\}$$

$$\text{s. t. } \mu(\boldsymbol{x}) \le \mu_0 \tag{1}$$

$$\mathbf{L} \le \boldsymbol{x} \le \mathbf{U}$$

where $\mu(\mathbf{x})$ requires simulation for evaluations. $\mathbf{L}$ and $\mathbf{U}$ are the lower and upper limits of resources. The feasible ranges $\mathbf{L} = \{L_1, \cdots, L_m\}$ and $\mathbf{U} = \{U_1, \cdots, U_m\}$ are given, and $\mu(\mathbf{U}) \le \mu_0 \le \mu(\mathbf{L})$ holds for a non-increasing function. Constraint (1) ensures that the system can satisfy the given performance level, and a second constraint (2):

$$\mu(x_1, ..x_i - 1, \cdots, x_m) > \mu_0, \forall\ i \in [1, m] \tag{2}$$

implies that if a single resource of any type is removed, the system no longer satisfies the performance level. For example, in the election context consider two resources: poll books and voting platforms. By setting $\mathbf{U} = \{5, 20\}$ and $\mathbf{L} = \{2, 3\}$ as the feasible ranges, we assume using our judgement that with 5 poll books and 20 voting machines, the system's performance is superior to the standard of 30 min; but with 2 poll books and 3 voting machines, its performance is inferior. Suppose we find two feasible solutions $\mathbf{S}_P$={(4 poll books, 8 machines), (3 poll books, 10 machines)}, where $\mathbf{S}_P$ is the feasible region (i.e., a contour solution set). If any machine or poll book is removed from either of these solutions, then the performance level of 30 min is no longer satisfied.

This problem is different from traditional ranking and selection simulation-optimization problems, where the objective can only be evaluated via simulation rather than the constraints. Considering the variations and noises, we may be indifferent to the exact value as long as the true value $\mu(\mathbf{x})$ is at most an "indifference-zone" parameter, $\delta > 0$, above or below the performance requirement. In other words, only when $\mu(\mathbf{x}) \ge \mu_0 + \delta$ or $\mu(\mathbf{x}) \le \mu_0 - \delta$, we consider it practically significant to detect the differences. In practice, we make our declarations based on sample means and the indifference zone value is simply a parameter of our testing method and concept used in method evaluation.

Next, we follow the notation for Generalized Binary Search (GBS) from Nowak (2011). A hypothesis matrix, $\mathbf{h} = \{h(\mathbf{x})\}$, is a set of declarations $h(\mathbf{x})$ about the feasibility or infeasibility of all combinations of resources, for any $\mathbf{x}$ satisfying $\boldsymbol{L} \le \mathbf{x} \le \boldsymbol{U}$. The evaluation of $h(\mathbf{x})$ is called a query. The value of an $h(\mathbf{x})$ is defined by

$$h(\mathbf{x}) = \begin{cases} +1 & \textit{if declaration is feasible} \\ -1 & \textit{if declaration is infeasible} \end{cases}$$

where a ranking and selection procedure makes the declarations for each query. Note that Nowak (2011) only considered the case in which the indifference zone parameter is zero ($\delta = 0$). Here, we offer a generalization taking indifference into account. Technically, if you declare empirically that the system is feasible, you are declaring $\mu(\mathbf{x}) - \delta \le \mu_0$. If you declare the system infeasible, you are declaring $\mu(\mathbf{x}) + \delta > \mu_0$.

Table 1 gives a hypothetical example of a hypothesis matrix with correct declarations involving two resources, x_1 and x_2. Suppose that $\mu(\mathbf{x} = (6, 6)) < \mu_0$, i.e., $\mathbf{x} = (6, 6)$ is feasible. If a ranking and selection procedure finds $h(\mathbf{x} = (6, 6)) = +1$, then it is a correct declaration; otherwise, the procedure makes an incorrect declaration. Suppose $\mathbf{x} = (10, 3)$ in this example corresponds to a situation in which $\mu_0 - \delta < \mu(\mathbf{x}) < \mu_0 + \delta$, either declaration $h(\mathbf{x} = (10, 3)) = +1$ or $h(\mathbf{x} = (10, 3)) = -1$ is considered correct within the indifference parameter.

3.2. Assumptions

Assumption 1. In this article, batch means from multiple simulation replications are written, $\bar{Y}_1(\mathbf{x})$, $\bar{Y}_2(\mathbf{x})$,..., and are assumed to be independent identically distributed (IID) normally distributed with

Table 1
A hypothesis matrix with correct declarations of a hypothetical example.

		x_1							
		6	7	8	9	10	11	12	13
x_2	1	−1	−1	−1	−1	−1	−1	−1	−1
	2	−1	−1	−1	−1	−1	−1	−1	−1
	3	−1	−1	−1	−1	−1/+1	+1	+1	+1
	4	−1	−1	+1	+1	+1	+1	+1	+1
	5	−1	+1	+1	+1	+1	+1	+1	+1
	6	+1	+1	+1	+1	+1	+1	+1	+1
	7	+1	+1	+1	+1	+1	+1	+1	+1
	8	+1	+1	+1	+1	+1	+1	+1	+1

finite variance for all possible values of $\mathbf{x}$. As noted by Nelson and Goldsman (2001), the IID assumption always holds true for replications. Also, the IID normally distributed assumptions are approximately true when using batches within a single long replication if the process is stationary and the batches are sufficiently large.

Assumption 2. Another assumption used is that the fitness function is monotone (non-increasing or non-decreasing) in all resource types. In the context of waits and elections, this could be interpreted as assuming that the batched average of the longest waiting time is non-increasing with the numbers of poll workers and voting machines. For example, for a single resource type this implies:

$$\mu(1) \geq \mu(2) \geq \cdots \geq \mu(\infty) \tag{3}$$

Note that assumption 2 may not apply to all cases. Theoretically, in some extreme cases, increasing resources in one queue might increase wait time in other queues within the same queueing network, which therefore could result in longer total wait time. Practically, however, such extreme cases are rarely encountered. Previous research has found monotonicity in certain queueing systems. For example, Yang et al. (2013) proves that for any given sample path the wait time of any customer in a single queue system is non-increasing as the number of server increases. It can be derived that the expected total wait time in an open network of *M/M/m* queues is non-increasing as the number of servers at some nodes increases (Bose, 2002; Jackson, 1963). Niu (1981) proves that the expected total wait time in a tandem queue with single server or multiple constant servers decreases as the service time decreases. Tay (1992) shows that the response time in a tandem queue with single servers decreases as the service times at any single server decreases. Wu, Shen, and Zhao (2017) provides an approximation formula for the mean queue time of a two-stage tandem queue with multiple servers with finite buffer, from which it can be derived that the approximated mean queue time is a non-increasing function of the number of servers. These theoretical results suggest that Assumption 2 is likely to be satisfied in practice.

To investigate the practical relevance of the non-increasing assumption, ten articles published in or after 2011 containing practical case studies with sufficient details for evaluation are identified (Gurumurthy & Rambabu, 2011; Inoue & Katsunori, 2012; Lam, Ng, Lakshmanan, Ng, & Ong, 2016; Pool, Wijngaard, & Zee, 2011; Robinson, Radnor, Burgess, & Worthington, 2012; Rohleder, Lewkonia, Bischak, Duffy, & Hendijani, 2011; Seo, Song, Kwon, & Kim, 2011; Sun, Lee, Chew, & Tan, 2012; Tizzoni et al., 2012; Zhang, Puterman, Nelson, & Atkins, 2012). In all ten articles, there is no mechanism reported such that adding resources would harm the service performance in their real-world applications. Moreover, a voting queue is tested using VBA coded discrete event simulation with 100 replications to evaluate the average and maximum expected waiting times (Fig. 1). The scenario is built for a polling station with 2,976 expected voters, based on an assemblage of the information supplied by the Ohio Franklin County Board of Elections for the 2012 presidential election. In Franklin County, polls are open for 13 h on Election Day, and the arrival process is modeled as a constrained non-stationary Poisson process. We assume the voter arrival rate doubles for the first two hours (6:30 AM to 8:30 AM) and final two hours (5:30 PM to 7:30 PM) compared to the rest of the day to simulate the "pre-work morning rush" and "afternoon rush". In addition, we employ an early voter arrival with the base arrival rate starting at 5:30 AM to preload the system. The registration time follows a distribution of *triangular*(0.9, 1.1, 2.9), and the service time using the voting machines is calculated based on the ballot length. In this scenario it follows a distribution of *triangular*(4.43, 9.14, 6.7) with an average of 6.76 min. The simulation outcome complies with the monotone assumption. As the number of DRE machines increases, both lines are smooth with error bars representing relatively narrow 95% confidence intervals. The half-widths of the confidence intervals range from 0.29 to 3.74 min.

Define $\mathbf{H}$ as a finite hypothesis matrix set consisting of all hypothesis matrices, $\mathbf{H} = \{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_M\}$. Let U_k and L_k denote the upper and the lower limits of each resource type k for $k \in \{1, \cdots, m\}$. Then, each hypothesis matrix $\mathbf{h}$ has $\prod_{k=1}^{m}(U_k - L_k + 1)$ declarations, where each declaration $h(\mathbf{x})$ is mapped to either "+1" or "-1". For instance, Example 1 shown in Table 1 has $(8 - 1 + 1)(13 - 6 + 1) = 64$ declarations $h(\boldsymbol{x})$. Thus, without assumption 2, there are a total of $2^{\prod_{k=1}^{m}(U_k - L_k + 1)}$ hypothesis matrices in $\mathbf{H}$. Because of assumption 2, we only need to consider the effective hypothesis matrices with the following structure:

- For any query generating a declaration of + 1, only $h(\mathbf{x})$ with + 1 values for higher resource levels are correct. For example, suppose (3, 5) generate a declaration of + 1, i.e., $\mu(3, 5) < \mu_0$, then $\mu(x_1, x_2) < \mu_0, \forall x_1 \geq 3 \cap x_2 \geq 5$.
- For any query generating a declaration of –1, only $h(\mathbf{x})$ with –1 value for lower resource levels are correct. For example, suppose (2, 4) generate a declaration of –1, i.e., $\mu(2, 4) > \mu_0$, then $\mu(x_1, x_2) > \mu_0, \forall x_1 \leq 2 \cap x_2 \leq 4$.

Therefore, the size of the hypothesis space, $|\mathbf{H}|$, can be reduced significantly. Table 2 gives a simple example of an initial hypothesis space $\mathbf{H}$ with two types of resources (x_1, x_2), where each type of resource has two possible values, and each hypothesis matrix $\mathbf{h}$ has $2 \times 2 = 4$ declarations. Thus, $|\mathbf{H}| = 2^4 = 16$ without assumption 2. However, as shown in Table 2, because of assumption 2, this hypothesis space has six effective hypothesis matrices $\mathbf{h}$ in total, i.e., $|\mathbf{H}| = 6$.

4. Indifference-Zone Generalized Binary Search algorithm

In this section, the Indifference-Zone Generalized Binary Search (IZGBS) method is proposed to find the contour solution set of the problem. This problem has been previously proven as NP-complete (Hyafil, 1976). Because the true value of each declaration is unknown, solving the problem involves finding the correct declarations and the effective hypotheses to enter the next iteration. IZGBS makes use of a procedure from Andradóttir and Kim (2010), which is reviewed in the appendix. We refer to this incorporating procedure as Andradóttir and Kim Phase I (AKPI) because it involves only the first phase of their method. As noted previously, the proposed method further extends the Generalized Binary Search because it involves an indifference parameter, δ, with $\delta \geq 0$ (Dasgupta, 2005; Nowak, 2011). Also, we included more than a single decision variable. Indifference parameters are relevant for empirical or simulation-based feasibility evaluations. Denote $\mathbf{H}_i$ as the current hypothesis space in the i^{th} iteration and $\mathbf{h}_j = \{h_j(\mathbf{x})\}$.

The Indifference-Zone Generalized Binary Search (IZGBS) Method

Initialize $i = 0$; significant level α, indifference parameter δ, and objective μ_0.
Enumerate all the hypothesis matrices $\mathbf{h}_j$ in the initial hypothesis space:
$\mathbf{H}_0 = \{\mathbf{h}_1, \cdots, \mathbf{h}_M\}$.
Do While $|\mathbf{H}_i| > 1$
Select point $\mathbf{x} = argmin_{\mathbf{x}} \left|\sum_{\mathbf{h_j} \in \mathbf{H_i}} h_j(\mathbf{x})\right|$
Use AKPI with $r = 1$, $\alpha / log\ |\mathbf{H}_0|$, δ, and μ_0 to make a declaration on whether $\mu(\mathbf{x}) \leq \mu_0$.
If it is declared that $\mu(\mathbf{x}) > \mu_0$ then,
Set$\mathbf{H}_{i+1} = \{\mathbf{h_j} \in \mathbf{H}_i : h_j(\mathbf{x}) = -1\}$
Else,
Set$\mathbf{H}_{i+1} = \{\mathbf{h_j} \in \mathbf{H}_i : h_j(\mathbf{x}) = +1\}$
End if
$i = i + 1$
End Do
$\mathbf{h}^* = \mathbf{H}_i$;
Select the contour solutions from $\mathbf{h}^*$ to the set
$\hat{\mathbf{S}}_P = \{(\hat{x}_1, \cdots, \hat{x}_m): h^*(\hat{x}_1, \cdots, \hat{x}_m) = +1 \cap (\hat{x}_1, \cdots \hat{x}_i - 1, \cdots, \hat{x}_m) = -1, \forall\ i \in [1, m]\}$.
Terminate

The algorithm starts with M hypothesis matrices in its initial search

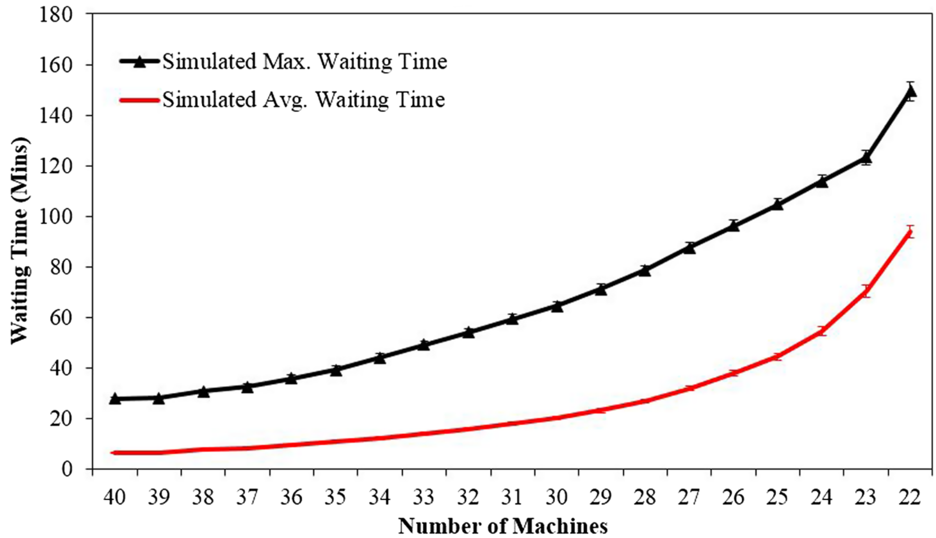

Fig. 1. Election day waiting time performance evaluation using DES and the average of 100 replications.

Table 2
Hypothesis space **H** with two types of resources.

$\mathbf{h_1}$		x_1 = 1	x_1 = 2
x_2	1	+1	+1
	2	+1	+1

$\mathbf{h_2}$		x_1 = 1	x_1 = 2
x_2	1	-1	+1
	2	+1	+1

$\mathbf{h_3}$		x_1 = 1	x_1 = 2
x_2	1	-1	+1
	2	-1	+1

$\mathbf{h_4}$		x_1 = 1	x_1 = 2
x_2	1	-1	-1
	2	+1	+1

$\mathbf{h_5}$		x_1 = 1	x_1 = 2
x_2	1	-1	-1
	2	-1	+1

$\mathbf{h_6}$		x_1 = 1	x_1 = 2
x_2	1	-1	-1
	2	-1	-1

space $\mathbf{H}_0$. At iteration i, we compute $\sum_{\mathbf{h_j}\in\mathbf{H}_i} h_j(\mathbf{x})$ for each point $\mathbf{x}$, and select an $\mathbf{x}$ that minimizes $\left|\sum_{\mathbf{h}\in\mathbf{H}_i} h_j(\mathbf{x})\right|$. We use the AKPI Method to make a declaration for this point on whether $\mu(\mathbf{x}) \le \mu_0$. According to the declaration, approximately half of the hypothesis matrices in $\boldsymbol{H_i}$ will be eliminated and the other half will be selected for the new hypothesis space $\boldsymbol{H_{i+1}}$ at the next iteration $i+1$. This query step is performed recursively until there is only one hypothesis ($\mathbf{h}^*$) left in the hypothesis space. Most of the time, $\mathbf{h}^*$and can be found in *Olog* $|\mathbf{H}_0|$ steps (Nowak, 2011). We then return all contour solutions to set $\hat{\mathbf{S}}_{\boldsymbol{P}}$ from the hypothesis matrix $\mathbf{h}^*$. A solution $(\hat{x}_1,\cdots,\hat{x}_m)$ will be considered a contour solution if $h^*(\hat{x}_1,\cdots,\hat{x}_m) = +1$ and $h^*(\hat{x}_1,\cdots\hat{x}_i - 1,\cdots,\hat{x}_m) = -1, \forall\, i \in [1,\, m]$, in $\mathbf{h}^*$. These are the "non-dominated" solutions.

For example, we use the hypothesis space in Table 2 to illustrate the algorithm. The initial search space $\mathbf{H}_0 = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \boldsymbol{h}_4, \boldsymbol{h}_5, \boldsymbol{h}_6\}$, with four possible solutions $\mathbf{x} \in \{(1, 1), (1, 2), (2, 1), (2, 2)\}$. At iteration 0, $\left|\sum_{j=1}^{6} h_j(1,\,1)\right| = 4$, $\left|\sum_{j=1}^{6} h_j(1,\,2)\right| = \left|\sum_{j=1}^{6} h_j(2,\,1)\right| = 0$, $\left|\sum_{j=1}^{6} h_j(2,\,2)\right|$ =4. Because either $(1, 2)$ or $(2, 1)$ minimizes $\left|\sum_{\mathbf{h}\in\mathbf{H}_i} h_j(\mathbf{x})\right|$, either one can be selected. Suppose point $(1, 2)$ is selected and AKPI method declares that $\mu(1,\,2) > \mu_0$. According to the declaration, $h_j(1,\,2) = -1$. Thus, $\mathbf{H}_1 = \{\boldsymbol{h}_3, \boldsymbol{h}_5, \boldsymbol{h}_6\}$. Because $|\mathbf{H}_1| = 3 > 1$, the algorithm continues to perform another iteration. Suppose $\boldsymbol{h}_3$ is the last matrix remaining and then we set $\boldsymbol{h}^* = \boldsymbol{h}_3$. In $\boldsymbol{h}^*$**,** only $(2, 1)$ can be considered as a contour solution, i.e., $\hat{\mathbf{S}}_{\boldsymbol{P}} = \{(2,\,1)\}$.

The following result provides a probabilistic guarantee of the quality for the contour solutions in the IZGBS derived set $\hat{\mathbf{S}}_{\boldsymbol{P}}$.

Theorem 1. *Assume Assumptions 1 and 2, and* $\boldsymbol{L} = \{L_1,\cdots,L_m\}$ *and* $\boldsymbol{U} = \{U_1,\cdots,U_m\}$ *are given. The solution set* $\hat{\boldsymbol{S}}_P = \{\hat{\boldsymbol{x}}_1,\cdots,\hat{\boldsymbol{x}}_n\}$ *derived from applying IZGBS with specified* α, δ *and* μ_0, *where* $\delta > 0$ *and* $0 < \alpha < 1$, *satisfies*

$$\Pr\{\mu(\hat{\mathbf{x}}_i) - \delta < \mu_0 < \mu(\hat{x}_{i,1},\cdots,\hat{x}_{i,k} - 1,\cdots,\hat{x}_{i,m}) + \delta\} \ge 1 - \alpha\ \forall\ i \in [1,\, n]\ \forall\ k \in [1,\, m] \tag{4}$$

Our proof of Theorem 1 is based on Bonferroni inequality and multiple applications of AKPI procedures used for the selection of $\hat{\mathrm{x}}_{\mathrm{i}}$.

Proof. IZGBS applies a total of up to *log* $|\mathbf{H}_0|$ GBS iterations and *log* $|\mathbf{H}_0|$ AKPI comparisons (Nowak, 2011). For each AKPI comparison, the Probability of Correct Selection (PCS) is greater than or equal to $1 - \alpha/log\ |\mathbf{H}_0|$ (Andradóttir & Kim, 2010). Denote the true value of selected $\hat{\mathbf{x}}_i$ as $\mu(\hat{\mathbf{x}}_i)$. Then, we have at least $1 - \alpha/log\ |\boldsymbol{H}_0|$ confidence of deriving the correct declaration when $\mu(\hat{\mathbf{x}}_i) \le \mu_0 - \delta$ or $\mu(\hat{\mathbf{x}}_i) \ge \mu_0 + \delta$. When $\mu_0 - \delta < \mu(\hat{\mathbf{x}}_i) < \mu_0 + \delta$, it does not matter whether the system is declared as feasible or not according to the problem statement. Thus, according to the Bonferroni inequalities, after at most *log* $|\mathbf{H}_0|$ times of AKPI comparisons (Nowak, 2011), we have at least $1 - \alpha$ confidence of picking the correct hypothesis simultaneously after all steps. By construction, the IZGBS method terminates when there is only one remaining hypothesis (Nowak, 2011). Also, by construction, all the neighbors of $\hat{\mathbf{x}}_i$ must have been directly or indirectly evaluated based on the monotone assumption. If the final hypothesis is correctly selected, all the $\hat{\mathbf{x}}_i$ remaining in the solution set $\hat{\mathbf{S}}_P$ must be declared as feasible solutions by AKPI, satisfying $\mu(\hat{\mathbf{x}}_i) \le \mu_0 + \delta$. Also, due to the contour solutions' property in the set $\hat{\mathbf{S}}_P$, their neighbor solutions $\{\hat{x}_{i,1},\cdots,\hat{x}_{i,k} - 1,\cdots,\hat{x}_{i,m}\}\ \forall\ i \in [1,\, n]\ \forall\ k \in [1,\, m]$ have to be declared infeasible, i.e., $\mu(\hat{x}_{i,1},\cdots,\hat{x}_{i,k} - 1,\cdots,\hat{x}_{i,m}) \ge \mu_0 - \delta\ \forall\ i \in [1,\, n]\ \forall\ k \in [1,\, m]$. Otherwise, the neighbor solutions would be better, and would replace $\hat{\mathbf{x}}_i$ in the contour solution set. As previously stated, Theorem 1 is guaranteed. ■

Note that if the user sets $\mu_0 + \delta = 30$ min and the performance metric is the waiting time, then IZGBS can be used to guarantee that all voters are expected to wait less than 30 min for each solution generated with bounded probability. The following corollary establishes conditions for which IZGBS terminates with the true contour set.

Corrolary 1. *Assume Assumptions 1 and 2. Also, assume (level separation) that the elements of the true contour solution set,* $\boldsymbol{S_P} = \{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}\ \forall\ \boldsymbol{x}_i \in \boldsymbol{S_P}$ *satisfy*$\mu(\boldsymbol{x}_i) \le \mu_0 - \delta$, $\mu(x_{i,1},\cdots,x_{i,k} - 1,\cdots,x_{i,m}) > \mu_0 + \delta\ \forall\ k \in [1,\, m]$, *and*$\mu(x_{i,1},\cdots,x_{i,k} + 1,\cdots,x_{i,m}) < \mu_0 - \delta\ \forall\ k \in [1,\, m]$, *then, the set* $\hat{\boldsymbol{S}}_{\boldsymbol{P}}$ *that IZGBS derives satisfies:*

$$\Pr\{\mathbf{S_P} = \hat{\mathbf{S}}_{\boldsymbol{P}}\} \ge 1 - \alpha \tag{5}$$

Proof. Theorem 1 gives $\mu(\hat{\mathbf{x}}_i) - \delta < \mu_0 < \mu(\hat{x}_{i,1},\cdots,\hat{x}_{i,k} - 1,\cdots,\hat{x}_{i,m}) + \delta$ $\forall\, i \in [1, n]\ \forall\, k \in [1, m]$ simultaneously with probability greater than $1 - \alpha$. Also, the separation assumption gives us that no neighboring solution of a contour solution, $\mathbf{x}_i$, has its mean within δ of $\mu_0\ \forall\, \mathbf{x}_i \in \mathbf{S}$.

Suppose $\boldsymbol{x} \in \hat{\boldsymbol{S}}_P$ but $\boldsymbol{x} \notin \boldsymbol{S}_P$ after performing IZBGS. From the Theorem 1 conditions, we have $\mu(x) - \delta < \mu_0 < \mu(x_1,\cdots,x_k - 1,\cdots,x_m) + \delta$. By the separation assumption, there are no non-contour solutions in this interval so we must have $\boldsymbol{x} \in \boldsymbol{S}_P$. Therefore, whenever the Theorem 1 conditions hold we have a contradiction (which occurs with probability greater than $1 - \alpha$).

Suppose $\boldsymbol{x} \in \boldsymbol{S}_P$ but $\boldsymbol{x} \notin \hat{\boldsymbol{S}}_P$. The separation assumption then contradicts the conditions of Theorem 1. Both the separation assumption and the conditions of Theorem 1 hold with probability greater than $1 - \alpha$. Therefore, $\Pr\{\mathbf{S_P} = \hat{\mathbf{S}}_\mathbf{P}\} \geq 1 - \alpha$. ■

5. Empirical evaluation

We compare the IZGBS method with the AKPI procedure using a full factorial experiment based on systems with multiple bottleneck resources (e.g., $q = 2$). The responses are average sample number (ASN) and probability of corrected selection (PCS). ASN is proportional to the computation time. This experiment involves four factors with two levels per factor. The first factor is the initial sample size considered at $n_0 = 10$ and 30. The second factor represents the feasible range of the number of resources. This factor relates to resource capacity as well as initial knowledge of the user. In our case, range sizes of 5×5 ($\mathbf{U} = \{10, 5\}$, $\mathbf{L} = \{6, 1\}$) and 8×8 ($\mathbf{U} = \{13, 8\}$, $\mathbf{L} = \{6, 1\}$) are considered. The third factor is the set of example problems as listed below:

Example 1: $y(x) = 100 - 2x_1 - 3x_2 - 2x_1x_2 + \varepsilon$
Example 2: $y(x) = 25 - 0.2x_1 - 0.3x_2 - 0.4x_1x_2 + \varepsilon$

where ε follows $N(0, 1)$. The last factor is the method, with the purpose to compare the effectiveness of AKPI and IZGBS algorithm.

The final hypothesis derived involves the Pareto Optimal solutions, feasible but not Pareto Optimal solutions, and the remaining solutions that are infeasible. The combination of those solutions could be represented by hypotheses with feasible/infeasible labels, and a correct selection is declared only when the correct hypothesis is picked. Theoretically, more than one correct hypothesis could exist because both AKPI and IZGBS are δ-optimality based methods.

In our numerical examples, the value of the indifference parameter, δ, is fixed at 0.5, and μ_0 is chosen to be 11, which means a feasible declaration is expected when $\mu \leq 10.5$, and an infeasible conclusion should be declared when $\mu \geq 11.5$. When $10.5 < \mu < 11.5$, either declaration can be accepted. The correct hypotheses of the example problems are listed in Table 1 and Table 3, respectively. In both tables, "-1" represents an infeasible solution, "+1" represents a feasible solution, and "-1/+1" indicates both declarations are acceptable.

Each design point is evaluated based on 1,000 replications. The resulting ASN and PCS as shown in Table 4. PCS denotes the probability of picking the correct hypotheses listed in Tables 1 and 3.

Table 3
Correct hypotheses of Example 2 in the full factorial design.

		x_1							
		6	7	8	9	10	11	12	13
x_2	1	−1	−1	−1	−1	−1	−1	−1	−1
	2	−1	−1	−1	−1	−1	−1	−1	−1/+1
	3	−1	−1	−1	−1	+1	+1	+1	+1
	4	−1	−1/+1	+1	+1	+1	+1	+1	+1
	5	+1	+1	+1	+1	+1	+1	+1	+1
	6	+1	+1	+1	+1	+1	+1	+1	+1
	7	+1	+1	+1	+1	+1	+1	+1	+1
	8	+1	+1	+1	+1	+1	+1	+1	+1

Table 4
Full factorial evaluation of AKPI and IZGBS.

Run	n_0	Range	Example	Method	ASN	PCS
1	30	5 × 5	Example 2	IZGBS	241.3	100.0%
2	10	8 × 8	Example 1	IZGBS	178.8	100.0%
3	10	5 × 5	Example 2	IZGBS	118.5	99.9%
4	10	8 × 8	Example 2	AKPI	767.3	99.7%
5	30	8 × 8	Example 2	AKPI	1929.7	99.9%
6	30	5 × 5	Example 2	AKPI	754.1	99.9%
7	30	5 × 5	Example 1	AKPI	756.7	100.0%
8	30	5 × 5	Example 1	IZGBS	242.2	100.0%
9	10	5 × 5	Example 1	AKPI	303.5	100.0%
10	30	8 × 8	Example 1	AKPI	1933.5	100.0%
11	30	8 × 8	Example 1	IZGBS	439.0	100.0%
12	10	8 × 8	Example 2	IZGBS	211.5	99.9%
13	10	5 × 5	Example 1	IZGBS	104.7	100.0%
14	10	8 × 8	Example 1	AKPI	716.1	100.0%
15	10	5 × 5	Example 2	AKPI	315.1	99.5%
16	30	8 × 8	Example 2	IZGBS	423.1	100.0%

The interaction plots for ASN and PCS are illustrated in Figs. 2 and 3. In both plots, the horizontal lines generally correspond to insignificant effects (with p-value ≥ 0.05). The lines with large slopes in absolute value generally correspond to strong effects (with p-value < 0.05). For example, the size of the range has a large and significant effect on ASN for the AKPI method but not for the IZGBS method.

Based on Fig. 2, we have the following findings:

1. Generally, IZGBS is uniformly superior to AKPI on the performance of ASN. This is expected because of the bisection nature of IZGBS as compared with the sequential approach of AKPI. In addition, the size of the range dramatically increases the total number of observations required by AKPI, but not for the IZGBS method due to the same reason. AKPI method performs evaluations for all systems, while the bisection nature of IZGBS approach largely reduces the effect of range on convergence (Nowak, 2011).
2. Different examples minimally affect the computational performance of both IZGBS and AKPI. These results indicate that the initial sample size of 10 or 30 is sufficient for the AKPI and IZGBS methods. For the same reason, ASN is highly associated with Initial Sample Size n_0 for both methods.

Fig. 3 shows the following findings:

1. In general, IZGBS allocates fewer samples but achieves a slightly higher PCS when compared with AKPI, but IZGBS and AKPI both perform well in terms of PCS owing to the application of the Bonferroni inequality. Bonferroni correction is free of any dependence assumption, and therefore leads to conservative results.
2. PCS is also significantly correlated with Initial Sample Size and Example. Intuitively, allocating more samples is more conservative, and a higher precision is expected. In addition, the difficulty of examples also negatively impacts the accuracy. It also seems that the PCS performance of IZGBS is slightly more robust to the Initial Sample Size, Range, as well as Example, when compared with AKPI.

Overall, both IZGBS and AKPI methods could achieve a high PCS in excess of the bounds as guaranteed in Section 3. The IZGBS method dominates the AKPI in terms of both ASN and PCS due to the more efficient bisection search design for choosing the next alternative system for comparison with the standard.

6. Guaranteeing short waits at the polls

In this section, IZGBS is applied in the context of election systems apportionment using the outputs from a discrete event simulation

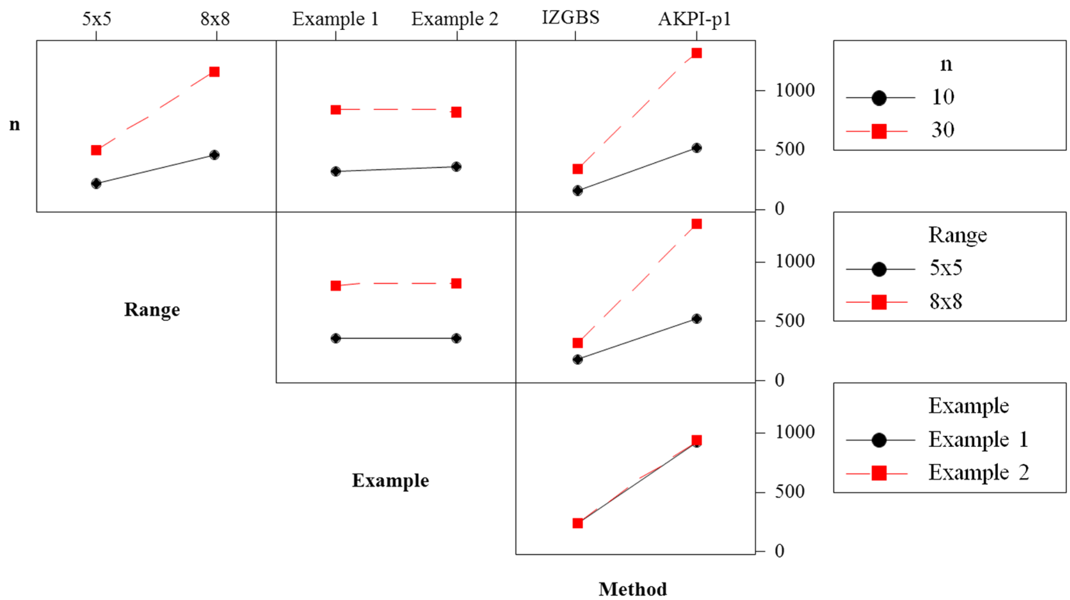

Fig. 2. Interaction plot for ASN of the full factorial design results.

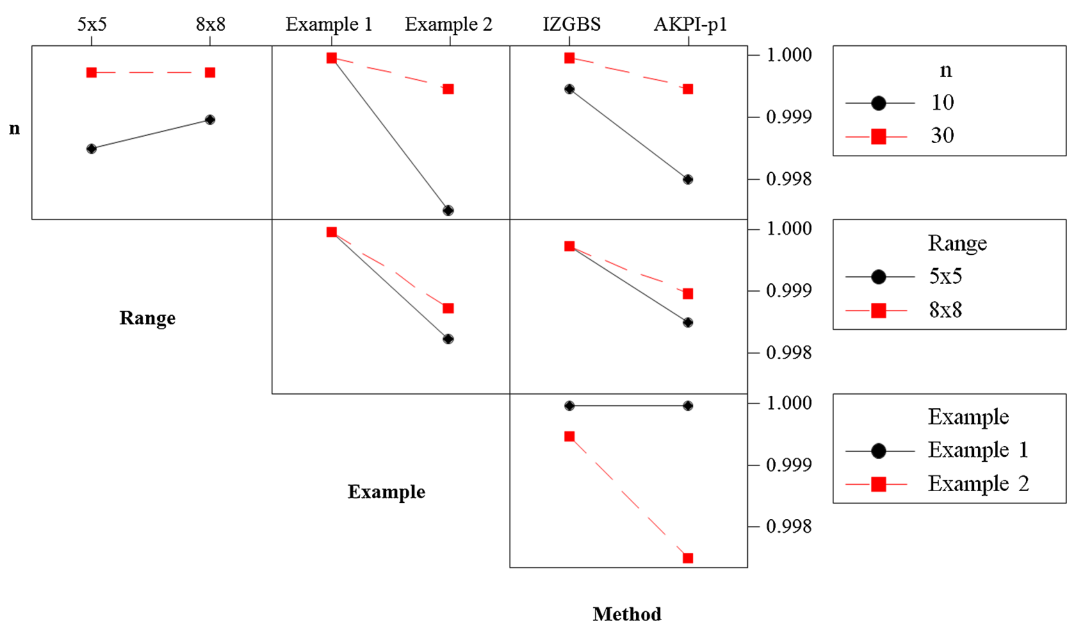

Fig. 3. Interaction plot for PCS of the full factorial design results.

model. In voting systems, IZGBS can derive a minimum number of machines needed to meet an objective, a relevant number for administrators who are purchasing and allocating the equipment. In U.S. election systems, apportionment occurs significantly in advance of the elections and often involves legislators. A related decision is allocation in which local officials determine the exact resource level at each location. IZGBS is mainly relevant to apportionment but it can also help in allocation through iterative application. For fixed resource levels, IZGBS can be applied iteratively to find the maximum attainable performance level.

The simulation model is constructed according to the voting process at a poll station on Election Day illustrated in Fig. 4, which can be considered as a two-stage tandem queue. The first process is voter check-in, where a voter's identity is validated; the second process is ballot casting, where a voter casts his ballot on a voting machine or in a voting booth. Generally, there are multiple poll books and multiple voting machines (or voting booths) at a poll station. Each process should ideally have a separate queue to avoid starving the bottleneck.

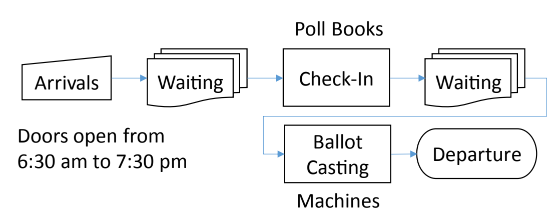

Fig. 4. Voting process.

As discussed in Section 3.2, the guarantee of IZGBS depends on the approximate normality of the performance metric data, which is the simulated waiting times in our case studies. However, raw simulation output, especially the associated order statistics (e.g. the longest waiting time), rarely follows a normal distribution. Therefore, two approaches are introduced as a necessary pre-step in Sections 6.1 and

Table 5
Computational results for the 8 issues service time case with 20 replicates and batch size = 10.

Expected voters	Number of issues on ballot	Resource	Exp. Quantile waiting time	SD of the mean	Avg. run time (seconds)
500	8	9	25.8	0.00	5.2
1000	8	17	28.6	0.00	14.2
1500	8	25	27.4	0.00	23.4
2000	8	33	29.1	0.00	45.4
2500	8	41	27.2	0.11	78.1
3000	8	48	28.2	0.17	108.3

6.2 to establish asymptotic normality. Computational comparisons are described to determine the appropriate batch size for the election system examples. Then, voting systems with a single resource, as well as multiple resources are studied in Sections 6.3 and 6.4. Resource levels associated with the objective that voters can expect to wait less than 30 min are provided for election officials together with information about the computational performance of the IZGBS method (Table 5 and Fig. 12).

6.1. Batch means

Batching is a typical simulation technique for approximating normal data. Let $Y_{l,j}$ denote a basic jth observation of the maximum waiting time for system l as listed in the AKPI procedure (appendix). When the distribution of $Y_{l,j}$ drifts significantly from normality, we replace individual observation $Y_{l,j}$ with batch means ($Y_{l,j}[b]$). $Y_{l,j}[b]$ denotes the averages of b basic observations of $Y_{l,j}$, which is assumed to be nearer normal compared to $Y_{l,j}$ due to the central limit theorem (Bekki, Fowler, Mackulak, & Nelson, 2007; Kim & Nelson, 2001; Law & Kelton, 2000). For example, with $j = 200$ simulation runs and a batch size$b = 5$, the 200 maximum waiting time will be grouped into 40 groups, and 40 batch means will be generated as the initial sample size. Therefore, $n_0 = j/b$ batch means $Y_{l,j}[b]$ are generated in the first state of sampling, and S_l^2 become the sample variance of $Y_{l,j}[b]$. Andradóttir and Kim (2010) evaluated the robustness of AKPI method to IID non-normal data through batching. For an extreme IID Bernoulli distribution, the minimum observed Probability of Correct Selection of the raw observation data ($b = 1$) equals 67%. As the batch size increases to 2, 5 and 10, the observed PCS quickly improves to 86%, 93% and 95%, respectively.

In the voting context, the waiting-time distribution for the voter who waits the longest depends on both the arrival and the service processes. Therefore, the batch size for approximate normality could depend on the resource level for a fixed arrival process. Here, we study two scenarios: scenario one with 2 machines, an average of 4.9 min as service time, and 76 expected voters ($NTotal = 76$) as the minimum count in the 2012 Ohio Franklin Presidential Election; scenario two with 34 machines, an average service time of 6 min, and the maximum number of expected voters ($NTotal = 2,976$). The observations of the waiting times for the voter who waits the longest are generated and grouped into batches of variable sizes (b=1, 2, 5, 10 and 20).

Figs. 5 and 6 illustrate the effect of batch size on the approximate normality of the batch means for each scenario with 250 replications. For the cases with large batch size ($b = 20$), the approximate normality is apparent in the plot. Using Kolmogorov-Smirnov testing, p-values are derived for the batch averages. For scenario one, the normality tests are rejected when $b = 1$ and 2 (p-values <0.01). For the other cases, the p-values are greater than 0.05, indicating some level of support for the normality assumption. Thus, we can conclude a batch size of 5 is needed for approximate normality for scenario one. For scenario two, approximate normality is only achieved when $b = 20$ (p-values >0.05).

6.2. Quantile estimators

The quantile-based approach is another choice to obtain approximately IID normal data. Compared to the basic observation (e.g. average/maximum waiting time), a nonparametric estimator of quantiles provides extra power by leveling the entire distribution of the performance metric, and is more robust against outliers and extremely skewed distributions.

Bickel and Lehmann (1975) studied three classes of quantile estimators: L-estimator is calculated as a linear function of order statistics; R-estimator is derived from rank tests; M-estimator is obtained as an optimizer of a criterion function (e.g. minimizing a sum function of the data). They found L-estimator is the only measure that provides high efficiency and robustness at the same time among all three estimators. Later, Dielman, Lowry, and Pfaffenberger (1994) evaluated ten nonparametric L-estimators by mean square error (MSE) and mean absolute deviation (MAD) for various distributions including skewed distributions, and recommended HD quantile estimator for its great performance over a broad range of cases ($0.98 \geq q \geq 0.02$), with some deficiencies for extreme percentiles. The HD estimator (Harrell & Davis, 1982) is formulated as a linear function by assigning a weight for each order statistic, and is given by:

$$HD_{l,j,q} = \sum_{n=1}^{NTotal} w_{NTotal,(n)} Y_{l,j,(n)} \tag{6}$$

where

$$w_{NTotal,(n)} = I_{n/NTotal}\{q(NTotal+1), (1-q)(NTtotal+1)\} - I_{(n-1)/NTotal}\{q(NTotal+1), (1-q)(NTtoal+1)\}$$

Here, $I_c(a, b)$ denotes the incomplete beta function. Let $Y_{l,j}$ denote a basic jth observation representing the maximum waiting time for system l as listed in the AKPI procedure. Assume system l has $NTotal$ expected voters, then we let $Y_{l,j,(1)} \leq Y_{l,j,(2)} \leq \cdots \leq Y_{l,j,(NTotal)}$ denote the order statistics of individual's voting time of all the voters collected in jth observation for system l. Then, let $HD_{l,j,q}$ denote the jth HD estimators of qth quantile for system l, and replace the individual observation $Y_{l,j}$ in the AKPI procedure (Appendix) with $HD_{l,j,q}$.

The asymptotically normal approximation of the HD estimator is judged to be adequate when $NTotal \geq 20$ and $q = 0.5$ or $50 \geq NTotal \geq 30$ and $q = 0.95$ for uniform and normal distributions; for asymmetric distribution such as exponential, $100 \geq NTotal \geq 80$ may be required when $q \geq 0.9$ (Brodin, 2007; Harrell & Davis, 1982). Batur (2010) applied the HD estimator in a two-stage quantile-based R&S procedure to address the issue of data non-normality.

In the context of voting, we perform similar batching exercise as in Section 6.1 for HD estimators to investigate its asymptotic normality. We are more concerned with the estimation of extreme right-tail quantiles (e.g. $q = 0.95$ or $q = 0.99$), because for most realistic voter waiting time distributions on Election Day, 0.7-quantile or below are usually minimal. Here, if we want to investigate the number of resources needed to guarantee that at least 95% of the voters wait for less than 30 min, then the HD estimators with $q = 0.95$ have to be determined. Figs. 7–10 illustrate the effect of batch size, number of expected voters and quantile on the approximate normality of data. Applying Kolmogorov-Smirnov testing, we find that when $NTotal = 76$, $q = 0.95$ *or* 0.99, a batch size of 10 is needed to achieve approximate normality ($p > 0.05$). When $NTotal = 2976$, $q = 0.99$, a batch size of 5 can satisfy the assumption of normality. With the same $NTotal$ of 2976, but a lower q of 0.95, $b = 2$ is sufficient for the approximate normality. In general, the normality condition can be more easily satisfied with a larger sample size (number of expected voters) and less extreme quantiles. Thus, for the remaining computational study, HD estimator with $b = 10$ and $q = 0.99$ is adopted since it represents relatively low computational costs with approximately normally distributed data.

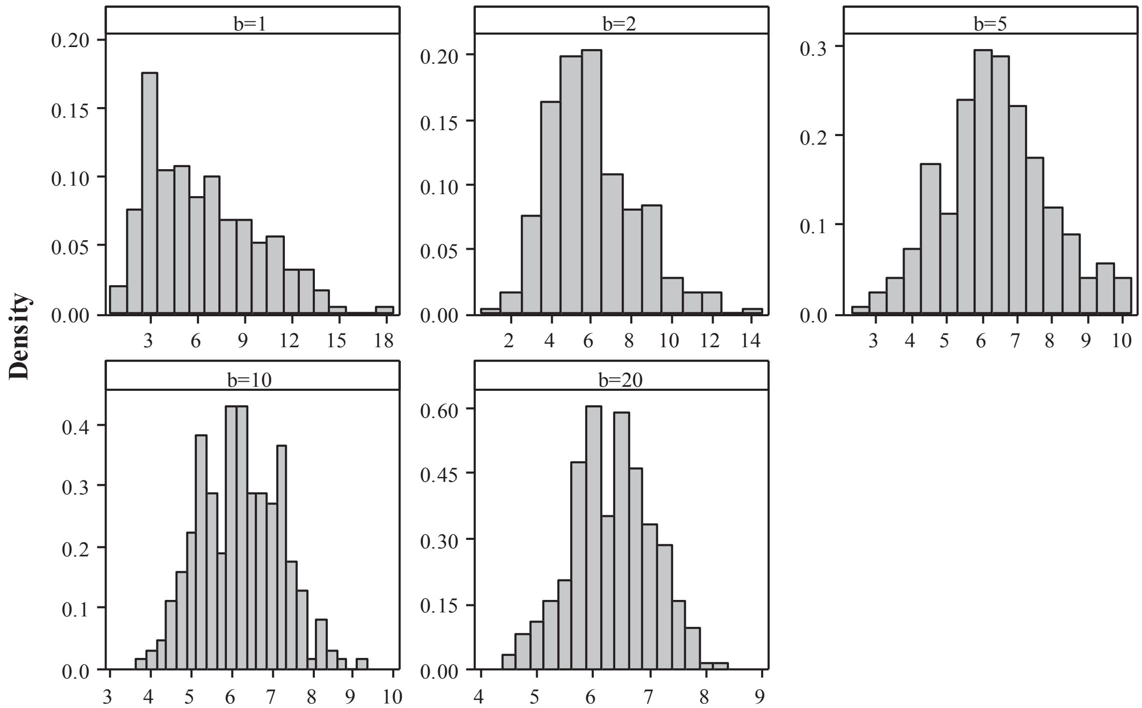

Fig. 5. The effect of batch size on approximate normality for maximum waiting time of scenario one.

6.3. Case study for single resource type

In this section, we apply IZGBS to a voting problem with one type of scarce resource. Local officials could appropriately apply the IZGBS method to allocate the bottleneck resource if other types of voting resources are sufficient. Section 6.4 illustrates the IZGBS method in the situation when multiple bottlenecks occur in a voting system.

Long voting lines were widely reported in central Ohio in 2004, in central Florida in 2012, and in Sandoval County of New Mexico in 2012. Many poll stations in these three elections involved bottlenecks of a single resource type, although the type of resource differed. Direct Recording Electronic (DRE) voting machines were used in central Ohio in 2004 (Allen & Bernshteyn, 2006); central Florida utilized voting booths in 2012 (Allen, 2013); and scanning machines were used in Sandoval County, New Mexico, in 2012 (Allen, 2014). The bottleneck can occur either at the voter check-in or at the voting machine/booth. When there is only one bottleneck in the process, the process can be approximately simplified to a single queueing node.

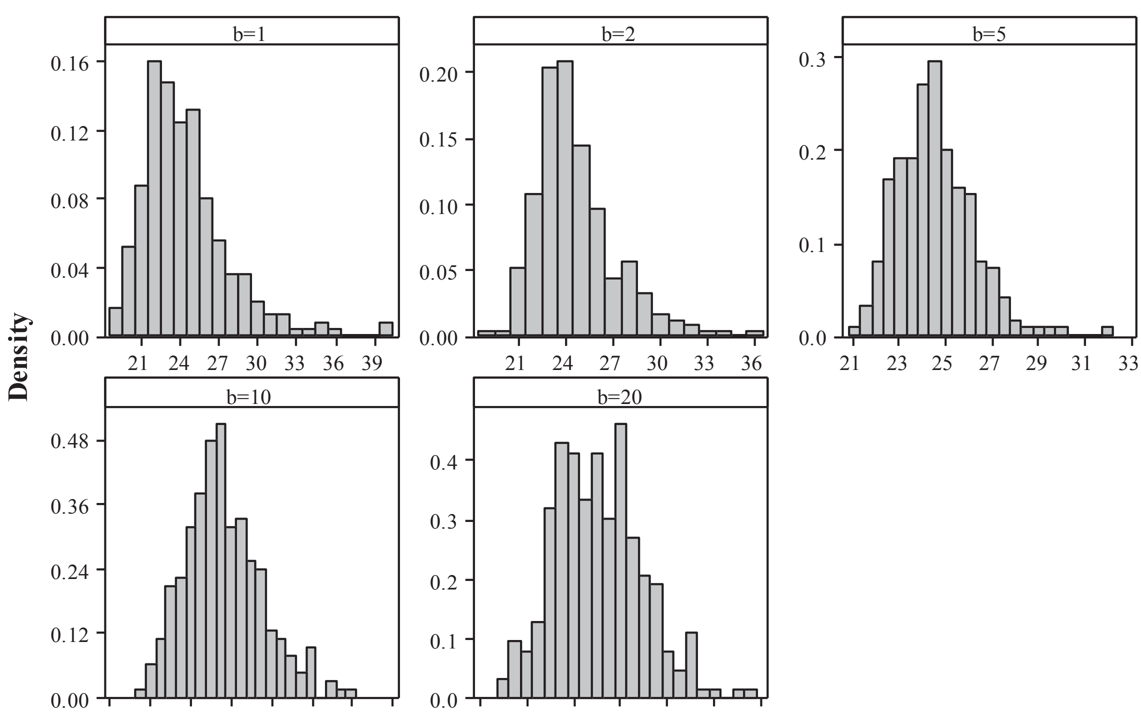

Fig. 6. The effect of batch size on approximate normality for maximum waiting time of scenario two.

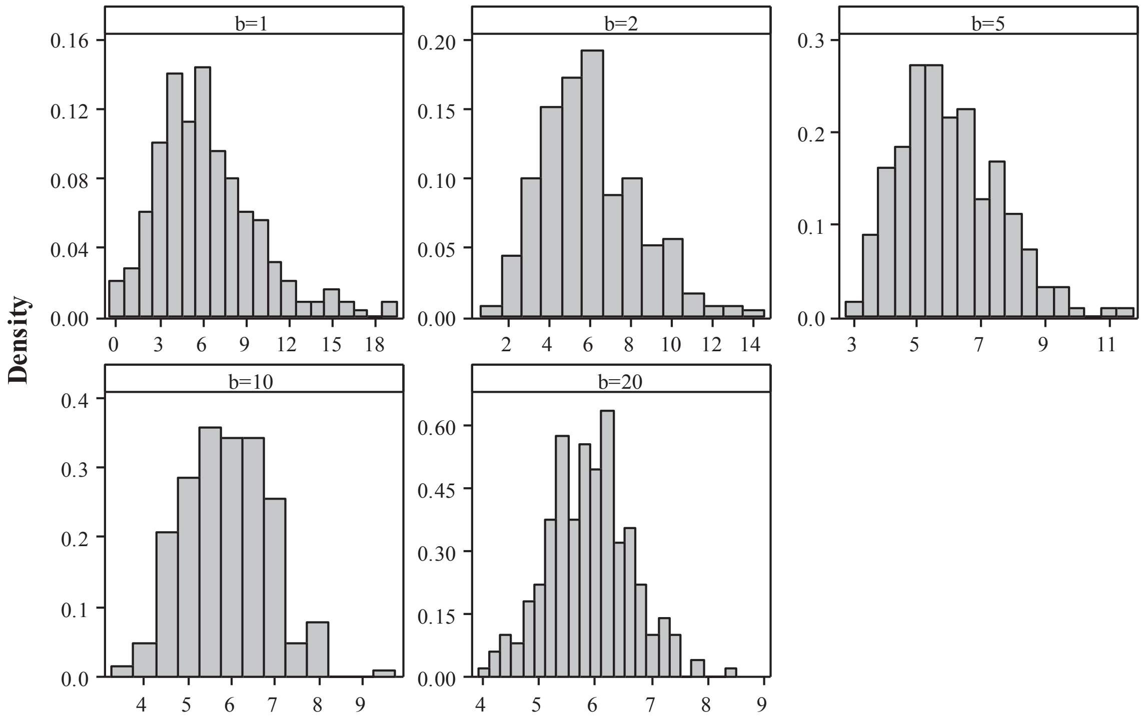

Fig. 7. The effect of batch size on approximate normality when $NTotal = 76$ and.$q = 0.99$

The simulation model includes a 13-hour election day with two rush periods: 6:30–8:30, and 17:30–19:30. The "thinning" method for the nonhomogeneous Poisson process is used so that the average arrival rate is doubled during the rush periods. The measure of service times across voting locations is correlated with the ballot lengths. In general, service time distributions do not differ greatly by the method of voting. For example, Fig. 11 shows measured service times for 20 voters during a mock election using DRE voting machines in Franklin County, Ohio and 24 voters in Saginaw County, Michigan in 2017.

This follows because methods ranging from DRE voting machines to voting booths with paper ballots may have approximately similar service distributions, which relate to the time during which the voter monopolizes the resource.

With a batch size $b = 10$, repeated applications of the IZGBS method using the simulation model can provide insights. The simulation model and IZGBS methods are coded in Visual Basic for Applications (VBA) in Excel for election officials to use. In this example, three service time distributions are considered motivated by Fig. 11: *triangular*(2, 7, 4) for

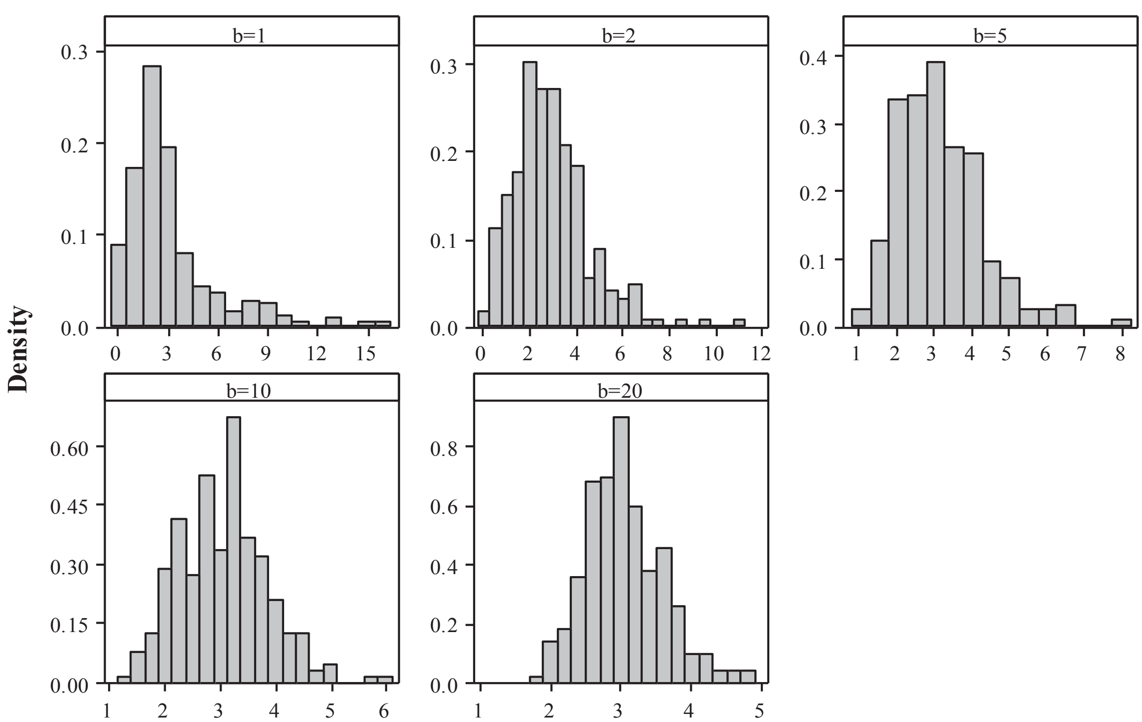

Fig. 8. The effect of batch size on approximate normality when $NTotal = 76$ and.$q = 0.95$

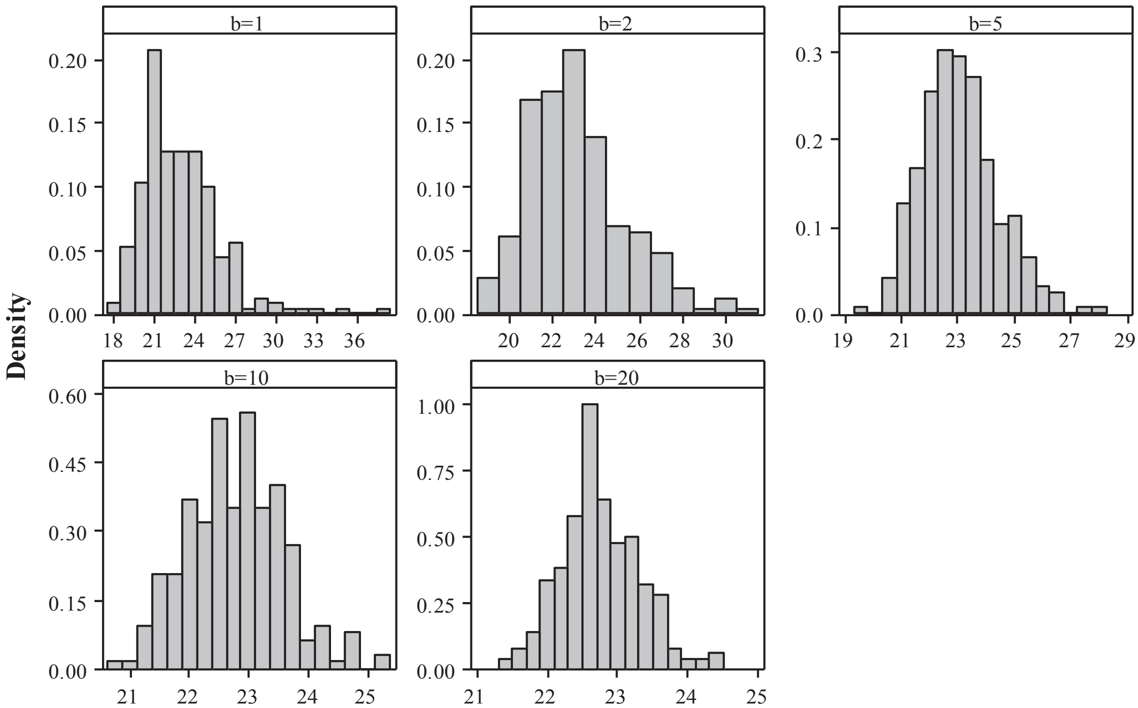

Fig. 9. The effect of batch size on approximate normality when $NTotal$ = 2976 and.q = 0.99

a ballot length of 4 issues, *triangular* (4, 9.5, 6) for a ballot length of 6 issues, *triangular* (6, 12, 8) for a ballot length of 8 issues. These service times correspond to short, medium, and long ballots cast on DREs or, alternatively, in voting booths with paper ballots. The simulation model assumes a known number of voters. In practice, the precise number of voters is unknown, but, for a conservative estimate, a relatively high, locally appropriate fraction of the registered voters can be used. In our experience, election officials are able to predict turnout within a few percentage points based largely on historical turnout for similar elections.

In this example, we apply an indifference zone of δ = 0.5 minute, a quantile of q = 0.99, and a performance objective μ_0 = 30 minutes. The objective of the IZGBS method is to derive the minimum number of resources needed to guarantee that 0.99-quantile of the voter waiting-time distribution is expected to be less than 30 min, with 0.5 as the significant difference to be detected.

The results of applying 20 repeated applications of IZGBS are shown in Table 5 corresponding to the case with 8 issues on the ballot. Six

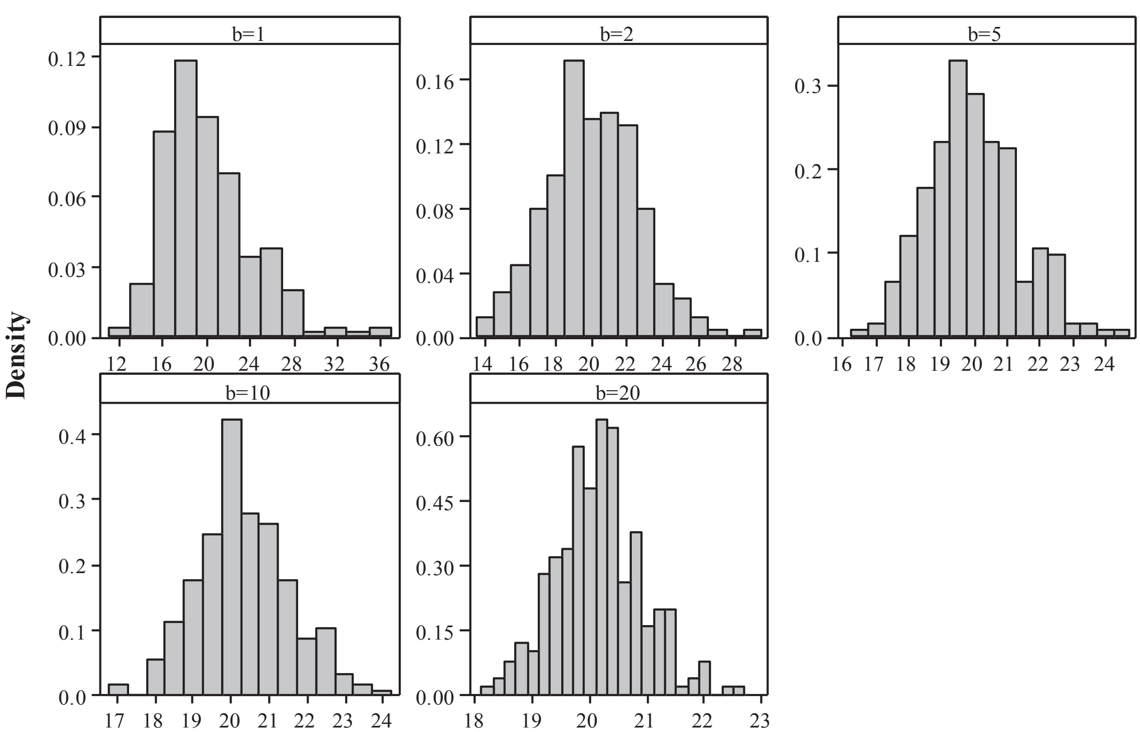

Fig. 10. The effect of batch size on approximate normality when $NTotal$ = 2976 and.q = 0.95

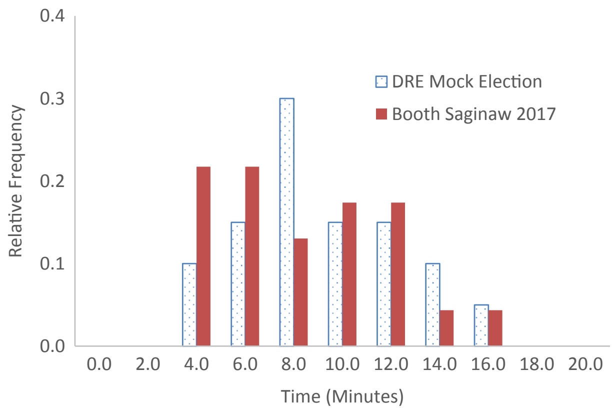

Fig. 11. Relative frequency histograms for service times for 20 electronic (DRE) voters in Franklin County, Ohio 2006 and 24 booth voters in Saginaw, Michigan 2017.

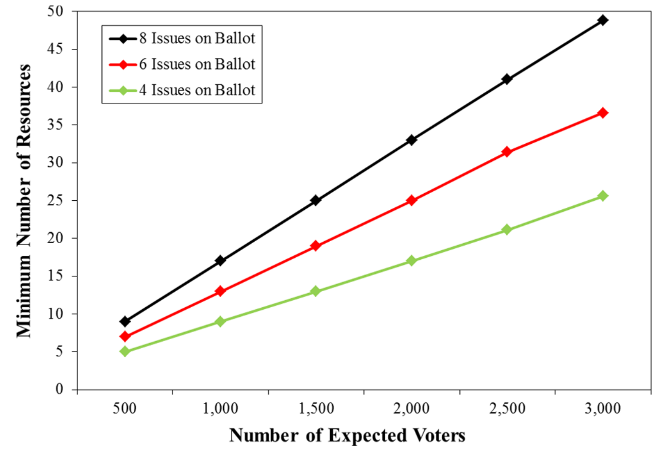

Fig. 13. Numbers of machines needed by numbers of voters and issues (single resource) for an expected 30 maximum minute or less wait.

scenarios with different numbers of voters are considered. Also, the average number of resources derived from IZGBS is shown as well as the standard deviations (SD) of the mean. The true minimum resource levels are likely to be the nearest rounded-up integer. For example, with 500 voters 9 resources are needed. The average computation run times are derived using a PC computer with Intel Core i5-5300U 2.30 GHz 2.30 GHz, 8.00 RAM, and 64-bit operating system. Our experiences indicate that C++ codes are between one and two orders of magnitude more efficient.

In Fig. 13, we consider six different voter numbers and three different processing times to illustrate the correlations. Fig. 13 may be immediately relevant for election officials and lawmakers. The results illustrate a clear correlation between the ballot lengths (average service times) and the numbers of resources needed.

In our experience, the numbers of machines required and displayed in Fig. 13 exceed the levels provisioned for voters. This occurs in part because many officials underestimate average service times. In 2016, we worked with Franklin County, Ohio and used our software to generate apportionments for 379 locations around central Ohio. The largest precinct had 9,237 registered voters and 6,114 expected voters. The IZGBS algorithm suggested that 34 more machines would be needed to ensure that 99% of voters in that precinct could expect to wait less than 30 min. This was the largest discrepancy. The officials found 16 more available machines and moved them to this precinct. Even with this change, waits of more than 3 h at that location were reported. Still, we conclude that thousands of voters were saved hours of waiting.

Software was created to determine the allocation and apportionment of resources, incorporating user input on the number of resources to be explored for just voting or for both voting and check-in (see Fig. 12). The user also determines whether to consider an average time, a maximum time, or a quantile for the waiting time. Details of the election can also be entered, such as start and end time. The distribution of the arrival rate is also entered. The user can also specify time periods with higher arrival rates, such as rush periods. And the desired indifference-zone is input as well. The number of items on the ballot are also entered. Based on the user's time constraints to get a response from the software, they can also decide if they want to achieve a faster result without a guarantee (probabilistic on solution quality), or a slower response with a guarantee.

As another example, in the 2008 election for Franklin County, Ohio locations with 1,000 voters had only 5 machines, far below the recommended number of 17 machines (for an 8 issue ballot) shown in Table 6. Of course, that election and the relevant locations became notorious with documented waiting lines longer than 5 h (Yang et al., 2014). In this example, focusing on the waiting time of the extreme

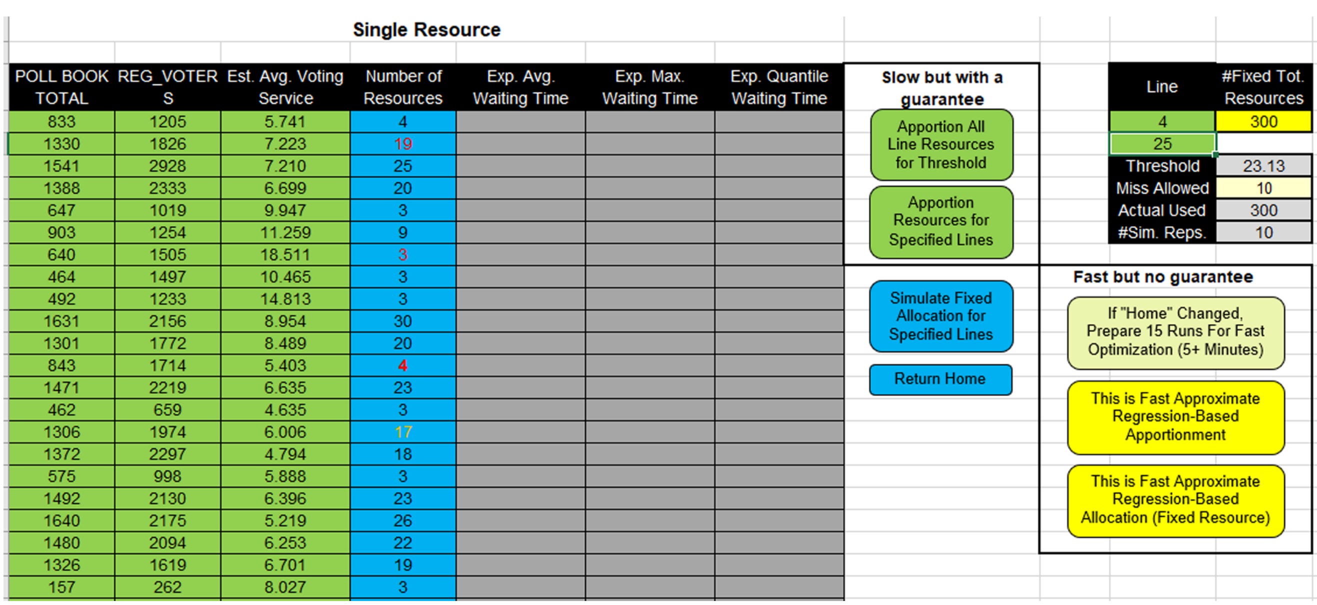

Fig. 12. Screen capture of the software with single resource example for allocation.

Table 6
Computational results for the voting problem with multiple resources.

Expected voters	Number of issues on ballot	No. resources needed (poll workers, machine)
500	8	(2,9)
1,000	8	(4,18)(5,17)
1,500	8	(9,25)
2,000	8	(8,35)(11,33)
2,500	8	(10,42)
3,000	8	(13,51)(15,50)
500	6	(2,7)
1,000	6	(4,13)
1,500	6	(6,20)(8,19)
2,000	6	(8,26)(11,25)
2,500	6	(9,32)(13,31)
3,000	6	(11,39)
500	4	(2,5)
1,000	4	(4,9)
1,500	4	(6,14)
2,000	4	(8,17)
2,500	4	(9,22)
3,000	4	(11,26)

right-tail quantiles, the recommended resource levels would effectively eliminate waiting for most voters.

6.4. Case study for multiple resource types

This section considers a voting process involving multiple types of resources. Although casting ballots is usually the bottleneck of the entire voting process, a real voting process could be more complicated. According to Spencer and Markovits (2010) and Stewart and Ansolabehere (2015), it is important also to consider the balance between the upstream service at the check-in tables and its cascading effects at the voting venues: congestions and lines could occur at both places. For example, some complicated voter-check-in cases may take a long time to complete before the voter starts to cast a ballot, which can create waiting lines at check-in even when the voting machine is idle.

In this section, we apply IZGBS to a more complicated voting problem with $q = 2$ resource types: poll books at check-in desks (i.e., service I) and voting machines to cast ballots (i.e., service II). We add an upstream check-in process with the service time distribution of $triangular(1, 3, 1.5)$ to the simulation model used in the previous section. We select this distribution based on the observation data collected in the 2016 Primary Election in Franklin County, which is consistent with a time baseline of 1.83 min observed by Spencer and Markovits (2010). The objective of this problem becomes finding the combinations of poll books and voting machines to ensure that the total waiting time at both services is less than μ_0=30 min with the indifference zone $\delta = 0.5$ minutes. The result is displayed in Table 6, which lists all the contour optimal solutions of resource combinations in the right column.

In Table 6, we verify the dependent relationship between multiple resources. Taking the scenario with 1,000 voters and 8 issues on the ballot as an example, we see 17 voting machines are suggested when 5 poll books (devices for efficient check-in) are required at the same time to ensure the standard of 30 min. When the number of poll books is reduced, an additional voting machine must be added accordingly to guarantee the same objective. Election officials may then determine which solution, (4,18) or (5,17), is subjectively the best. The computational results of contour solutions are also displayed in Fig. 14, where the size of the bubble represents the number of expected voters, and the color represents the number of issues on the ballot. From this, election officials could use their preferences to decide the best strategy for resource allocation from the contour solutions as shown in Fig. 14.

Remark 1. Note that, for lower computational cost, the results here pertain to the voter who waits at the 0.99-quantile in a single location. The simulation can also be used to derive resource needs if the performance metric is the absolute longest wait by adjusting the batch size as illustrated in Section 6.1. Additionally, in a typical county there might be hundreds of locations in parallel. Yet, simulation permits the exploration of the voter who waits the longest or 0.99-quantile at each individual location, as well as across multiple locations.

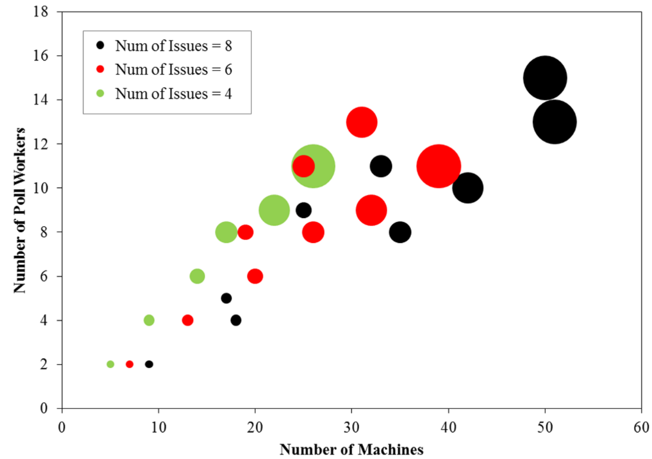

Fig. 14. Numbers of machines needed by number of voters and number of issues (multiple resources).

Remark 2. Our algorithm is fast enough to be solved repeatedly with iterated thresholds (μ_0) so that the total resource constraints are met over hundreds of predicts for minimax optimal performance.

7. Conclusions

This article presents a rigorous method to determine the minimum resource levels required to satisfy a given service level objective. The proposed IZGBS method addresses a class of problem, where the system performance level is non-decreasing in the amount of resources and can only be evaluated via simulation. The IZGBS method has potentially wide applicability, particularly in cases where the standard is written into law. It could provide a defensible approach to satisfy a given standard with proven probability bounds. The efficiency of IZGBS permits it to play a "building block" role in the development of optimization methods. The method can be extended to a "dual" problem with simulated parallel systems to derive resource allocation, with the objective of achieving the best possible service level for the worst performing single system, while constraining the total amount of resources below a specified level. The case studies presented in the context of voting resource allocation provide specific insights of potentially immediate relevance of IZGBS to law makers.

The direct application of IZGBS allows elections officials to predict waiting times (or line lengths) and create optimal allocation strategies with solution quality guarantees. Long voting lines have played an important role in several election lawsuits in recent years. For example, the long lines during the Arizona primary election in 2016 brought Arizona election officials into court (Feldman, 2016). Election officials are accused of not using scientific methods to allocate sufficient voting resources, which therefore resulted in long voting lines and suppressed voter turnout. Election officials in Ohio were also charged with disproportionally allocating fewer machines to precincts with predominately minority voters during the 2004 presidential election (Tanner, 2005). Clearly, election officials need a scientific and defensible method to allocate voting resources. Election officials are generally open to optimization methods for resource allocation as long as they comply with constraints from their state secretary of state directives. Our method, scientific and defensible, can not only meet such needs but also avoid the type of discrimination that has occurred in recent elections (Allen, 2013, 2014), as well as be used to support laws that eliminate in-person waiting.

Moreover, our method is relatively computationally efficient compared to the alternatives that we consider. Election officials usually need to plan for the upcoming elections several months in advance. For example, election officials start to make the tentative plans for polling stations in August for a November election. Their forecast for Election Days could be improved as more information is known, especially after the early voting period. Our easy-to-use method allows election officials to run the program whenever necessary, so that they can create the voting resources allocation plan using the most updated information. We are posting Excel macros on our personal website, which are available to all officials with instructions. We selected Excel because it is currently in wide use by officials for allocation. As described in Section 6.3, our application of the software and methods likely saved hours of waiting for thousands of voters in 2016.

As shown in Tables 3 and 4, resource requirements are provided such that 99% voters are expected to wait less than 30 min. This would critically change existing provisions that mandate resources per registered voter and ignore non-stationary arrivals and service time variability. As ballot lengths differ from county to county and precinct to precinct (sometimes by over 6 pages) (Allen & Bernshteyn, 2006) these results can help to account for service time differences when allocating resources in a way that is rarely done in practice. Even for short and/or constant length ballots, the results can aid in predicting waiting times before they occur and prevent potential voter disenfranchisement caused by long lines.

Several topics remain for future research. First, even faster methods could increase the usability of the software and its adoption by election officials for allocation and legislators for apportionment. Such methods could, conceivably, use metamodeling since the service and waiting distributions across hundreds of locations are likely similar. Second, methods that explicitly minimize the maximum waiting time over hundreds of locations or some other measure of equity with constrained resources are also of interest. In our Remark 2, we describe how our software can be used to iteratively set the threshold or maximum value to employ all available resources, but the process is arguably too slow for at least some potential users. Third, if the monotone assumption is known to be violated, the logarithm bound on the number of necessary steps may not apply. A new way of preprocessing needs to be developed so that it can eliminate hypotheses using any known structure, and a more sophisticated approach to reduce the available hypotheses at each step can be implemented. Fourth, infeasibility alerts could be developed for cases in which legislators are apportioning too few resources or space constraints making satisfying relevant thresholds impossible.

CRediT authorship contribution statement

Theodore T. Allen: Conceptualization, Funding acquisition, Project administration. **Muer Yang:** Methodology, Validation. **Shijie Huang:** Formal analysis, Software. **Olivia K. Hernandez:** Writing - review & editing.

Acknowledgements

Enhao Liu provided key edits and feedback. We want to acknowledge Kirk Cherep and Kelly Parriman from Integrated System Engineering of the Ohio State University for their important contribution in collecting the check-in processing times at the 2016 Franklin County Primary Election. Edward Leonard, David Payne, Zach Manifold, and Carolyn Gorup contributed ideas and encouragement. NSF grant #1409214 provided partial support for this work.

Appendix. Review of methods from Andradóttir and Kim

In this appendix, a comparison with a standard procedure presented by Andradóttir and Kim (2010) is described. A special case of this procedure is incorporated into the "Indifference-Zone Generalized Binary Search" (IZGBS) method proposed in Section 3.

Formulation of Andradóttir and Kim's Method

Andradóttir and Kim (2010) presented a constrained R&S method to select the system with the best primary performance measure from a finite number of parallel systems. Meanwhile, the selected system must satisfy a stochastic constraint of a second performance measure.

Assuming there is a total of $r \geq 2$ unordered alternative systems indexed by $l = 1, \cdots, r$. Let $Z_{l,j}$ and $Y_{l,j}$ be two univariate real-value random variables derived from simulation, which associate with the primary and secondary performance measures from replication (or batch) j of system l, respectively. Then, the mean response values for the alternative system l become $E[Z_l]$ and $E[Y_l]$ for $l = 1, \cdots, r$. The problem seeks to find system l that satisfies:

$$\min_{l=1,\cdots,r} E[Z_l]$$

$$s.\ t.\ E[Y_l] \leq \mu_0 \tag{A.1}$$

The Andradóttir and Kim Phase I Method

Andradóttir and Kim's method (AKPI) is a two-stage procedure with a screening procedure in the first stage to identify all the feasible systems. In the first phase, AKPI attempts to identify a subset including all the feasible solutions. This effectively solves the formulation in Eq. (A.1) because once the feasible set is identified, determining the Pareto optimal solution set is trivial. In the second phase of their method, AKPI searches among these solutions for the single best solution.

The AKPI screening procedure starts with a set of candidate solutions, $\mathbf{M} = \{1, 2, \cdots, r\}$ and ends with a set $\mathbf{F}$ which consists of all the feasible solutions. Consider the event of a non-increasing fitness function, the objective becomes to seek the solution set $\mathbf{F}$ containing all feasible systems satisfying $E[Y_l] \leq \mu_0 - \delta$ but no infeasible solutions with $E[Y_l] \geq \mu_0 + \delta$. The rest of the systems satisfying $\mu_0 - \delta < E[Y_l] < \mu_0 + \delta$ are considered as acceptable solutions, which may be declared as either feasible or infeasible. Andradóttir and Kim (2010) proved that AKPI achieves this objective with a probability greater than $1 - \alpha$ for all systems under the normality assumption.

AKPI is efficient in terms of sample size (Chen, 2006). At each stage, it only takes one more sample from each remaining system whose feasibility is undetermined. Once clear evidence is shown, the undetermined system will be moved to either feasible solution set or infeasible system set.

The Andradóttir and Kim Phase I (AKPI) Procedure

STEP 1. Select $n_0 \geq 2$, and nominal Probability of Correct Select decision $1 - \alpha$. Select c as any positive integer, although in practice, $c = 1$ is recommended (Kim & Nelson, 2001). Compute $\eta > 0$, which is the solution to the equation $g(\eta) = 1 - (1 - \alpha)^{\frac{1}{r}} = \sum_{i=1}^{c} (-1)^{i+1}\left(1 - \frac{1}{2}\mathfrak{T}(i = c)\right)\left(1 + \frac{2\eta(2c - i)i}{c}\right)^{-(n_0-1)/2}$, where $\mathfrak{T}$ is the indicator function.

STEP 2. Initialize $\mathbf{M} = \{1, 2, \cdots, r\}$ and $\mathbf{F} = \{\varnothing\}$, which is the set of systems whose feasibility is not determined yet and the set of systems declared to be feasible, respectively. Let $h_1^2 = 2c\eta(n_0 - 1)$. Generate n_0 observations of $Y_{l,j}$ for $j = 1, 2, \cdots, n_0$ from each system l, and compute the standard deviation S_l^2. Set the stage counter to $u = n_0$ and go *Step 3*.

STEP 3. For each system $l \in \mathbf{M}$, if $\sum_{j=1}^{u}(Y_{l,j} - \mu_0) \leq -R(u;\delta, h_1^2, S_l^2)$, then move system l from $\mathbf{M}$ to $\mathbf{F}$; else if $\sum_{j=1}^{u}(Y_{l,j} - \mu_0) \geq +R(u;\delta, h_1^2, S_l^2)$, then eliminate l from $\mathbf{M}$. Here, $R(u;\delta, h_1^2, S_l^2) = \max\left\{0, \frac{h_1^2 S_l^2}{2c\delta} - \frac{\delta}{2c}u\right\}$.

STEP 4. Let $|\mathbf{M}|$ denote the number of elements in set $\mathbf{M}$. If $|\mathbf{M}| = 0$, then return $\mathbf{F}$ as a set of feasible systems. Otherwise, take one additional observation $Y_{l,u+1}$ from each system $l \in \mathbf{M}$, set $u = u + 1$, and go to *Step 3*.

Remark 1. Logically, assuming the alternatives are systems with different combinations of resources $\mathbf{x}$, and the feasible set is correctly identified, then it is trivial to directly apply AKPI to solve Eq. (A.1) for the single resource type case. For multiple resource type case, one can simply select all the Pareto Optimal solutions in the feasible solution set $\mathbf{F}$. Therefore, AKPI effectively seeks to solve the formulation in equation (A.1).

Remark 2. A special case of the AKPI method involves a single system being compared with a standard, i.e., $r = 1$. This is the method incorporated into the Indifference-Zone Generalized Binary Search (IZGBS) introduced in Section 3. There, a single system characterized by resource level $\mathbf{x}$ is compared with the standard. If the system evaluated is found to be feasible, then the resulting set $\mathbf{F}$ is not empty.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.cie.2019.106243.

References

Akan, M., Oguzhan, A., Baris, A., Fatih, S. E., & Adnan, S. (2012). A broader view of designing the liver allocation system. *Operations Research, 60*(4), 757–770.

Allen, T. T. (2013). Delving into the reasons for long lines can bring solutions. *Orlando Sentinel* Retrieved from https://www.orlandosentinel.com/opinion/os-xpm-2013-01-08-os-ed-long-lines-voting-florida-010813-20130107-story.html.

Allen, T. T. (2014). Fleming v. Gutierrez: Expert witness report. Dist. Court, D. New Mexico (No. 13-CV-222 WJ/RHS).

Allen, T. T., & Bernshteyn, M. B. (2006). Mitigating voter waiting times. *Chance: A. Magazine of the American Statistical Association, 19*(4), 25–36.

Alvarez, R. M., Ansolabehere, S., Bernsky, A., Lenz, G., Stewart III C., & Hall T. (2008). 2008 Survey of the performance of american elections: Final report, white paper. Retrieved from http://elections.delaware.gov/pdfs/SPAE_2008.pdf.

Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2016). Simulation optimization: A review of algorithms and applications. *Annals of Operations Research, 240*(1), 351–380.

Andradóttir, S., & Kim, S. H. (2010). Fully sequential procedures for comparing constrained systems via simulation. *Naval Research Logistics (NRL), 57*(5), 403–421.

Batur, D. (2010). A quantile-based approach to system selection. *European Journal of Operational research, 202*(3), 764–772.

Bechhofer, R. E., Goldsman, D. M., & Santner, T. J. (1995). *Design and analysis of experiments for statistical selection, screening, and multiple comparisons.* Cambridge, MA: Wiley.

Bekki, J. M., Fowler, J. W., Mackulak, G. T., & Nelson, B. L. (2007). Using quantiles in ranking and selection procedures. 2007 Winter simulation conference, Washington, DC, pp. 1722–1728.

Bertsimas, D., Farias, V. F., & Trichakis, N. (2011). The price of fairness. *Operations Research, 59*(1), 17–31.

Bertsimas, D., & Gupta, S. (2011). A proposal for network air traffic flow management incorporating fairness and airline collaboration. Working Paper, MIT. Retrieved from http://www.mit.edu/~dbertsim/papers.html.

Bickel, P. J., & Lehmann, E. L. (1975). Descriptive statistics for nonparametric models II. Location. *Annals of Statistics, 3*(5), 1045–1069.

Bose, S. K. (2002). *An introduction to queueing systems.* US: Springer.

Brennan Center for Justice (2019). New voting restrictions in America. Retrieved from https://www.brennancenter.org/our-work/research-reports/new-voting-restrictions-america.

Brodin, E. (2007). Extreme value statistics and quantile estimation with applications in finance and insurance. (Ph.D. Thesis). Chalmers University of Technology and University of Göteborg.

CBS News. (2016). Official's comments on long voting lines in Phoenix spark uproar. Retrieved from https://www.cbsnews.com/news/phoenix-long-voting-lines-arizona-presidential-primary-blame-voters-election-2016/.

Cassidy, C. A., Long, C., & Balsamo, M. (2018). Machine breakdowns, long lines mar vote on election day. *AP News* Retrieved from https://www.apnews.com/6fb6de6fdb034b889d301efd12602e21.

Chen, C. H., Lin, J., Yücesan, E., & Chick, S. E. (2000). Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic systems, Theory and Applications, 10*, 251–270.

Chen, E. J. (2006). Performance evaluations of comparison-with-a-standard procedures. *Proceedings of the 38th conference on Winter simulation. Winter simulation conference* (pp. 360–368). .

Chick, S. E., & Inoue, K. (2001). New two-stage and sequential procedures for selecting the best simulated system. *Operations Research, 49*, 732–743.

Cohen, A. (2008). No one should have to stand in line for 10 hours to vote. *The New York Times* Retrieved from. http://www.nytimes.com/2008/08/26/opinion/26tue4.html.

Dasgupta, S. (2005). Analysis of a greedy active learning strategy. *Advances in neural information processing systems 17* (pp. 337–344). MIT Press.

Dielman, T., Lowry, C., & Pfaffenberger, R. (1994). A comparison of quantile estimators. *Communications in Statistics - Simulation and Computation, 23*(2), 355–371.

Edelstein, W. A., & Edelstein, A. D. (2010). *Queuing and elections: Long lines, DREs and paper ballots. Proceedings of the 2010 electronic voting technology workshop/workshop on trustworthy elections, Washington, DC, August 9-10*.

Erhardt, E. B., Huang, S., Huerta, G. & Allen T. T. (2015). Did "effective disenfranchisement" affect Sandoval County NM outcomes on Election Day during the 2012 election? (Working Paper).

Famighetti, C. (2016). Long voting lines: Explained. Brennan center for justice. Retrieved from https://www.brennancenter.org/sites/default/files/analysis/Long_Voting_Lines_Explained.pdf.

Feldman, L. (2016). Feldman et al v. Arizona Secretary of State's Office et al, United States District Court District of Arizona. (No. CV-16-01065-PHX-DLR).

Frazier, P. I. (2014). A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct election. *Operations Research, 62*(4), 926–942.

Gao, S., Chen, W., & Shi, L. (2017). A new budget allocation framework for the expected opportunity cost. *Operations Research, 65*(3), 787–803.

Gini, C. (1912). Variabilità e Mutabilità. C. Cuppini, Bologna.

Grant, F. H., III (1980). Reducing voter waiting time. *Interfaces, 10*(2), 19–25.

Gurumurthy, A., & Rambabu, K. (2011). Design of lean manufacturing systems using value stream mapping with simulation: A case study. *Journal of Manufacturing Technology Management, 22*(4), 444–473.

Harrell, F. E., & Davis, C. E. (1982). A new distribution-free quantile estimator. *Biometrika, 69*(3), 635–640.

Herron, M. C., & Smith, D. A. (2016). Precinct resources and voter wait times. *Electoral Studies, 42*, 249–263.

Highton, B. (2006). Long lines, voting machine availability, and turnout: The case of Franklin County, Ohio in the 2004 presidential election. Retrieved from *PS: Political Science & Politics, 39*(1), 65–68. https://doi.org/10.1017/S1049096506060148.

Hunter, S. R., & Pasupathy, R. (2013). Optimal sampling laws for stochastically constrained simulation optimization on finite sets. *INFORMS Journal on Computing, 25*(3), 527–542.

Hyafil, L. (1976). Constructing optimal binary decision trees is NP-complete. *Information processing letters, 5*(1), 15–17.

Inoue, T., & Katsunori, O. (2012). Grinding mechanism of centrifugal mills — A simulation study based on the discrete element method. *Comminution, 1994*, 425.

Jackson, J. R. (1963). Jobshop-like queueing systems. *Management Science, 10*(1), 131–142.

Kim, S. H., & Nelson, B. L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation, 11*(3), 251–273.

Kim, S. H., & Nelson, B. L. (2006). Selecting the best system. *Handbooks in Operations Research and Management Science, 13*, 501–534.

Kozanidis, G. (1991). Solving the linear multiple-choice knapsack problem with two objectives: Profit and equity. *Computational Optimization and Applications, 43*, 261–294.

Kumar, S., Yang, M., & Goldschmidt, K. (2018). Will aging voting machines cause more voters to experience long waits? *International Journal of Production Economics, 198*, 1–10.

Lam, S. S. W., Ng, Y. S., Lakshmanan, M. R., Ng, Y. Y., & Ong, M. E. H. (2016). Ambulance deployment under demand uncertainty. *Journal of Advanced Management Science, 4*(3).

Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). New York: McGraw-Hill.

Lee, L. H., Chew, E. P., Teng, S., & Goldsman, D. (2010). Finding the non-dominated Pareto set for multi-objective simulation models. *IIE Transactions, 42*(9), 656–674.

Marist poll (2008). Election Security. Retrieved from http://maristpoll.marist.edu/9-18-election-security.

Marsh, M. T., & Schilling, D. A. (1994). Equity measurement in facility location analysis:

A review and frame-work. *European Journal of Operational Research, 74*, 1–17.
Nelson, B. L., & Goldsman, D. (2001). Comparisons with a standard in simulation experiments. *Management Science, 47*, 449–463.
Ni, E. C., Ciocan, D. F., Henderson, S. G., & Hunter, S. R. (2017). Efficient ranking and selection in parallel computing environments. *Operations Research, 65*(3), 821–836.
Niu, S. (1981). On the comparison of waiting times in tandem queues. *Journal of Applied Probability, 18*(3), 707–714. https://doi.org/10.2307/3213324.
Norden, L. (2013). How to fix long lines. The Brennan center for justice at New York University School of Law, New York, New York, USA. Retrieved from http://www.brennancenter.org/sites/default/files/publications/How_to_Fix_Long_Lines.pdf.
Nowak, R. D. (2011). The geometry of generalized binary search. *IEEE Transactions on Information Theory, 57*(12), 7893–7906.
Pastrana, L. (2012). M-D early voter: I waited "over six" hours to vote. CBS Miami. Retrieved from http://miami.cbslocal.com/2012/11/03/last-day-of-early-voting-opens-in-miami-dade/.
The Pew Charitable Trusts. (2014). New Hampshire elections performance index. Retrieved from http://www.pewtrusts.org/~/media/assets/2014/04/07/2012_election_performance_index_new_hampshire.pdf?la=en.
Pool, A., Wijngaard, J., & Zee, D. V. D. (2011). Lean planning in the semi-process industry, a case study. *International Journal of Production Economics, 131*(1), 194–203.
Powers, S., & Damron, D. (2013). Analysis: 201,000 in Florida didn't vote because of long lines. Orlando Sentinel. Retrieved from http://articles.orlandosentinel.com/2013-01-29/business/os-voter-lines-statewide-20130118_1_long-lines-sentinel-analysis-state-ken-detzner.
Presidential Commission on Election Administration. (2014). The American voting experience: Report and recommendations of the presidential commission on election administration. Retrieved from https://www.supportthevoter.gov/files/2014/01/Amer-Voting-Exper-final-draft-01-09-14-508.pdf.
Robinson, S., Radnor, Z. J., Burgess, N., & Worthington, C. (2012). Simlean: Utilising simulation in the implementation of lean in healthcare. *European Journal of Operational Research, 219*(1), 188–197.
Rohleder, T. R., Lewkonia, P., Bischak, D. P., Duffy, P., & Hendijani, R. (2011). Using simulation modeling to improve patient flow at an outpatient orthopedic clinic. *Health Care Management Science, 14*(2), 135–145.
The Schaefer Center for Public Policy. (2014). Voting and the administration of elections in Maryland, Retrieved from http://www.elections.state.md.us/press_room/documents/Schaefer%20Center%20Final%20Report.pdf.
Seo, K. M., Song, H. S., Kwon, S. J., & Kim, T. G. (2011). Measurement of effectiveness for an anti-torpedo combat system using a discrete event systems specification-based underwater warfare simulator. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 8*(3), 157–171.
Spencer, D. M., & Markovits, Z. S. (2010). Long lines at polling stations? Observations from an election day field study. *Election Law Journal, 9*(1), 3–17.
Stein, R. M., Mann, C. B., Stewart, C., Birenbaum, Z., Fung, A., Greenberg, J., et al. (2019). Waiting to vote in the 2016 presidential election: Evidence from a multi-county study. *Political Research Quarterly*. https://doi.org/10.1177/1065912919832374.
Steuer, R. E., & Na, P. (2003). Multiple criteria decision making combined with finance: A categorized bibliographic study. *European Journal of Operational Research, 150*(3), 496–515.
Stewart III, C., & Ansolabehere, S. (2013). Waiting in Line to Vote - White paper. Retrieved from https://www.eac.gov/documents/2017/02/24/waiting-in-line-to-vote-white-paper-stewart-ansolabehere/.
Stewart, C., III, & Ansolabehere, S. (2015). Waiting to vote. *Election Law Journal, 14*(1), 47–53.
Sun, Z., Lee, L. H., Chew, E. P., & Tan, K. C. (2012). Microport: A general simulation platform for seaport container terminals. *Advanced Engineering Informatics, 26*(1), 80–89.
Tanner, J. (2005). Letter from U.S. Department of Justice, Civil Rights Division to Nick Soulas, Jr., Assistant Prosecuting Attorney of Franklin County, Ohio. Retrieved from https://www.npr.org/documents/2005/oct/doj_docs/doj_franklin_oh.pdf.
Tay Y.C. (1992). A monotonicity theorem for tandem queues (Technical Report). Princeton University. Retrieved from https://www.cs.princeton.edu/research/techreps/TR-362-92.
Tizzoni, M., Bajardi, P., Poletto, C., Ramasco, J. J., Balcan, D., Goncalves, B., et al. (2012). Real-time numerical forecast of global epidemic spreading: Case study of 2009 a/h1n1pdm. *BMC Medicine, 10*(1), 165.
Ury, B. (1969). Ury v. Santee. Dist. Court, N. D. Illinois (No. 69 C 1146).
Walker, J. (2013). Va. Senate rejects bills addressing long voting lines. *The Virginian Pilot* Retrieved from https://pilotonline.com/news/government/politics/virginia/article_56c4cada-bee1-5741-9ba3-d5c31fb79912.html.
Wu, K., Shen, Y., & Zhao, N. (2017). Analysis of tandem queues with finite buffer capacity. *IISE Transactions, 49*(11), 1001–1013. https://doi.org/10.1080/24725854.2017.1342055.
Yang, M., Allen, T. T., Fry, M. J., & Kelton, W. D. (2013). The call for equity: Simulation-optimization models to minimize the range of waiting times. *IIE Transactions, 45*, 1–15.
Yang, M., Fry, M. J., Kelton, W. D., & Allen, T. T. (2014). Improving voting systems through service-operations management. *Production and Operations Management, 1*, 1–15.
Yang, M., Wang, X., & Xu, N. (2015). A robust voting machine allocation model to reduce extreme waiting. B *Omega - The International Journal of Management Science, 57*, 230–237.
Young, H. P. (1995). *Equity in theory and practice.* Princeton, New Jersey: Princeton University Press.
Zhang, Y., Puterman, M. L., Nelson, M., & Atkins, D. (2012). A simulation optimization approach to long-term care capacity planning. *Operations Research, 60*(2), 249–261.