Pure and Applied Mathematics Journal
2025, Vol. 14, No. 2, pp. 13-23
https://doi.org/10.11648/j.pamj.20251402.11

Research Article

Implementation of a Voting Method Based on Mean-Deviation Evaluation for a Large-Scale Election

Hadarou Yiogo*, Zo ïnabo Savadogo

Department of Mathematics, Laboratory of Numerical Analysis, Computer Science and Biomathematics, Joseph KI-ZERBO University, Ouagadougou, Burkina Faso

Abstract

Today, many countries around the world, particularly in Africa, are experiencing post-election difficulties due to unexpected election results. This sometimes provokes protests and revolt among the population. To overcome this major problem, several voting systems have been developed in the literature, but some of them are not lacking in shortcomings. It was with this in mind that the voting method based on the evaluation of the mean deviation was born. It's a voting system that seems to be appreciated because it respects a certain number of fundamental properties of a ranking method. On the other hand, we note in the literature that it is only applicable to small-scale data with an insignificant number of candidates and voters. For this reason, we set ourselves the goal of implementing this method in order to extend its use to large-scale problems. Thus, we proposed the computer program using python software, which takes as input the scores assigned to the candidates by each voter and displays as output the best candidate. To do this, we built sub-programs such as median, arithmetic mean and mean-spread functions, each of which plays an effective role in selecting the best candidate. We then studied the algorithmic time complexity theoretically, then graphically, and ended by applying our computer program to several voting examples containing a very large number of candidates and voters. Numerous applications enabled us to observe that, whatever the size of the data, we always obtained a conclusive and satisfactory result with polynomial-type time complexity.

Keywords

Implementation, Voting Method, Mean- Deviation, Election

1. Introduction

One of the principles of a country's development is based on transparency and democracy, i.e. working to guarantee transparent and participative governance to involve the population in decision-making [8]. In order to achieve virtuous and consensual management, theorists have turned to elections and put in place numerous voting methods that satisfy the properties of democracy [3]. This is why, theorists believe that that elections are the heart of democracy [12]. In today's society, voting is widely used to resolve certain social choices. Consequently, the search for a voting method that better reflects reality seems to be a top priority for the future [11]. Many voting methods have already been developed by a number of authors, enabling a better choice to be made that satisfies certain voting properties [9, 10]. However, many of these methods suffer from shortcomings, especially when applied to large-scale data. As a result, the use of computer

*Corresponding author: yiogohadarou95@gmail.com (Hadarou Yiogo)

Received: 7 March 2025; **Accepted:** 17 March 2025; **Published:** 10 April 2025

tools has become indispensable in social choice theory to overcome these difficulties. The voting method based on the evaluation of the mean deviation already exists in the literature, with satisfactory properties. However, its application in the context of large-scale problems is computationally very robust and difficult to obtain a choice, which motivates us to propose its implementation. In our work, we will first describe the voting method based on the evaluation of the mean deviation.

2. Literature Review

2.1. Single-Member Constituency Vote

This is the type of voting method generally used in Africa. Each voter votes for at most one candidate, and the candidate with the highest number of votes is the best candidate. This is the voting method used in BURKINA FASO [1, 4-7].

2.2. Two-round Uninominal Voting

Each elector votes for a maximum of one candidate. If a candidate obtains an absolute majority (> 50%), he or she is elected. If not, a second round is held between the two candidates with the most votes. This rule is the most widely used in the world for direct elections (universal suffrage) [6].

2.3. Approval Voting

Approval or assent voting is a voting system that allows each voter to approve one, two, all or none of the candidates. The voter is not forced to choose a single candidate. The voter has the choice of selecting several, all or none of the candidates, but cannot vote more than once for the same candidate. It's a voting system that's simple to study and championed by many theorists [2].

2.4. Social Choice Functions

A social choice function is a function C that associates a non-empty subset of A with any situation (election), i.e.:

$$C: \{\text{situation}\} \to K \subset A$$

$$(A, p) \mapsto C(A, p) \subseteq A \text{ with } C(A, p) \neq \varnothing$$

Presentation of the voting method based on the evaluation of the arithmetic mean and the mean deviation

In this section, we have drawn inspiration mainly from [10].

Definition 2.1 (Median)

Consider x_1, x_2, …, x_i,…, x_n the series arranged in ascending order. Thus the median is as follows:

If n is odd

$$x_{1/2} = x_{((n+1)/2)} \tag{1}$$

If n is even

$$x_{1/2} = \frac{1}{2}\{x_{(n/2)}+x_{(n/2\ +1)}\} \tag{2}$$

Definition 2.2 (Arithmetic mean)

Consider x_1, x_2, …, x_i,…, x_n a series of values, the arithmetic mean is calculated as follows:

$$M = \frac{\sum_{i=1}^{n} xi}{n} \tag{3}$$

Definition 2.3 (Mean deviation)

Let x_1, x_2, …, x_i,…, x_n be a note series and $\bar{x}|$ the arithmetic mean of this series. Its mean deviation is determined as follows:

$$EM = \frac{\sum |xi-\bar{x}|}{n} \tag{4}$$

This method consists of classifying candidates into five categories in order of preference. A candidate ranked in the best class obtains 5 points. If he or she is ranked in the next highest category, he or she gets 4 points, and so on down to the last category, where he or she gets 1 point. The next step is to rank the candidates' scores in ascending order. The median of the ordered candidates' scores is then calculated. When a candidate's median score is 1 or 1.5, depending on the parity of the number of voters, he or she is systematically eliminated.

To ensure the best choice, we calculate the arithmetic mean and the average deviation. In this way, the candidate with the best mean and the smallest average deviation is elected. If the mean difference is equal, the process is repeated.

Consider a set E of m candidates for an election, with m ≥ 2, i.e. Consider a set E of m candidates for an election, with m ≥ 2, i.e. $E = \{c_1, c_2, …., c_m\}$ and F

A set of voters with $s \geq 2$, c'est àdire $F = \{v_1, v_2, …., v_s\}$. So the method is as follows:

1. Each of the s voters uses disjoint elements of P(E) whose union gives E, in the following order: choice choice 1, choice 2, choice 3, choice 4, choice 5.
2. It assigns each element in each of its subsets a score of 1, 2, 3, 4 or 5 respectively.

Example: Let $E = \{c_1, c_2, c_3, c_4, c_5\}$ a set of 5 candidates. The following table represents the choice of a voter: $\{c_1\}$, $\{c_2\}$, $\{c_3\}$, $\{c_4\}$, $\{c_5\}$

Now let's calculate the mean deviation of the scores of the different candidates. Note that, in this table, we have also calculated the mean deviation of candidate 1. This is optional, as the median score of candidate 1 is 1 and, according to the method, candidate 1 is systematically eliminated.

3. Expérience Numérique

Let's consider a vote of 11 voters and 6 candidates:

Candidate 1: 1 1 1 1 1 1 1 5 5 5 5
Candidate 2: 1 3 3 3 3 3 3 3 3 3 3
Candidate 3: 1 1 1 1 1 5 5 5 5 5 5
Candidate 4: 1 2 2 2 2 3 3 3 3 3 3
Candidate 5: 1 1 1 1 1 1 5 5 5 5 5
Candidate 6: 1 1 1 2 4 4 4 4 4 5 5

Let's put these different scores into a table and determine the average for each candidate. This example is taken from and solved manually in [10].

Table 1. *Table of candidate scores.*

	C_1	C_2	C_3	C_4	C_5	C_6
X_1	1	1	1	1	1	1
X_2	1	3	1	2	1	1
X_3	1	3	1	2	1	1
X_4	1	3	1	2	1	2
X_5	1	3	1	2	1	4
X_6	1	3	5	3	1	4
X_7	1	3	5	3	5	4
X_8	5	3	5	3	5	4
X_9	5	3	5	3	5	4
X_{10}	5	3	5	3	5	5
X_{11}	5	3	5	3	5	5
$\bar{x}$	2.45	2.81	3.18	2.45	2.81	3,18

Table 2. *Average deviation table for candidate 1.*

Candidate 1												
x_i	1	1	1	1	1	1	1	5	5	5	5	EM
$\|xi-\bar{x}\|$	1,45	1,45	1,45	1,45	1,45	1,45	1,45	2,55	2,55	2,55	2,55	1,85

Table 3. *Average deviation table for candidate 2.*

Candidate 2												
x_i	1	3	3	3	3	3	3	3	3	3	3	EM
$\|xi-\bar{x}\|$	1,81	0,19	0,19	0,19	0,19	0,19	0,19	0,19	0,19	0,19	0,19	0,33

Table 4. *Average deviation table for candidate 3.*

Candidate 3												
x_i	1	1	1	1	1	5	5	5	5	5	5	EM
$\|xi-\bar{x}\|$	2,18	2,18	2,18	2,18	2,18	1,82	1,82	1,82	1,82	1,82	1,82	1,98

Table 5. *Average deviation table for candidate 4.*

Candidate 4												
x_i	1	2	2	2	2	3	3	3	3	3	3	EM
$\|xi-\bar{x}\|$	1,45	0,45	0,45	0,45	0,45	0,55	0,55	0,55	0,55	0,55	0,55	0,57

Table 6. *Average deviation table for candidate 5.*

Candidate 5												
x_i	1	1	1	1	1	1	5	5	5	5	5	EM
$\lvert xi - \bar{x}\rvert$	1,81	1,81	1,81	1,81	1,81	1,81	2,19	2,19	2,19	2,19	2,19	1,98

Table 7. *Average deviation table for candidate 6.*

Candidate 6												
x_i	1	1	1	2	4	4	4	4	4	5	5	EM
$\lvert xi - \bar{x}\rvert$	2,18	2,18	2,18	1,18	0,82	0,82	0,82	0,82	0,82	1,82	1,82	1,47

We note that the calculation of average deviations gives a consensus for candidate 3 and candidate 6. Indeed, according to the method, the best candidate is candidate 6, as it obtains a smaller mean deviation than candidate 3.

This voting method is used to solve certain social choice problems and more generally, to solve certain group decision problems. This recurrent use is due to the satisfactory results it provides and the fundamental properties it verifies.

Like the other methods, this voting system based on average deviation evaluation has the following properties following properties according to [10]:

1. It verifies Condorcet's criterion.
2. Binary actions are independent:

To rank, for example, 2 candidates among several others, it is sufficient to know the preferences of each voter for these two candidates; their choices for the others do not change the ranking between these two candidates.

3. Monotonicity:

If x is elected in one election and in a second an elector who voted against x changes his mind in favor of x, then x is always elected.

4. Unanimity or Pareto

4. Principle of Implementation

4.1. Proposal of the Method Algorithm

Following the presentation of the method, we'll now describe its computer algorithm, which takes as input an integer table containing the candidates' scores and returns the smallest mean deviation, i.e. the best candidate. The algorithm is shown below.

ALGORITHMS

Input: L representing candidate scores: Integer

Output: minimum(EM) i.e. the minimum of the average deviations belonging to the best candidate

```
            N number of rows in L
            M number of columns in L
         Median[n]: list of real numbers
         Moy[n]: a list of real numbers
         EM[n]: a list of real numbers
         MaxL [n]: a list of real numbers
         m1, m2, cm1, cm2, k1, a1, a2, mn1, mn2 variables of real type
         Ind[n]: An empty list
ListMin[n]: An empty list
ListMax[n]: An empty list
candNd[n]: An empty list
Ecart[n, m]:: A null matrix of dimension N×M
Start
      L← sort table L
%Calculate the median of each row in the table, which is equivalent to the median of each candidate.
Median ← median(L),
Moy ← mean(L),
        For p ranging from O to length(Mediane)-1 do
              If Mediane[p]=1 or Mediane[p]=1.5 then
                   Add p+1 to list candNd
              End If
End For
For i from 0 to N-1 do
             List ←L[i]
             For j from 0 to M-1 do
             a ← (List[j]-Average[i])
             Deviation[i,j] ← |a|
           End for
           EM ← Average(Deviation),
           MaxL ← maximum(Moy),
End for
For i from 0 to length(Moy)-1 do
            If Moy[i]=MaxL then
              ListMax ← Moy[i]
              add Moy[i] to ListMax list
```

```
add i+1 to Ind list
ListMin ← EM[i]
add EM[i] to ListMin list
End If
End For
Display("ascending filtering of candidate scores", L)
Display("number of voters", M)
Display("number of candidates", N)
For n from 0 to length(candNd)-1 do
    n1 ← candNd[n]
    Display("candidate n°",n1)
End for
For m from 0 to N-1 do
    m1 ← m+1
    m2 ← Mediane[m]
    Display ("the median of candidate n° ", m1)
    Display("is ", m2)
End for

For cm from 0 to N-1 do
    cm1 ← cm+1
    cm2 ← Average[cm]
    Display("the average of candidate n°", cm1)
    Display ("is ", cm2)
End for
If length(ListMax)=1 then
    max ←maximum(Average),
    Add index of max to variable t
    Display ("The lucky winner is candidate n°", t+1)
Otherwise
    Display("There are ties. The list of ties is:")
    For k from 1 to length(ListMax)-1 do
        k1 ← Ind[k]
        Display("candidate n°", k1)
    End For
    display("and candidate no.", length(ListMax))
    Display("the deviation table is: ", Deviation)
    Display("average deviation of each candidate is:")
    For a from 0 to length(EM)-1 do
        add a+1 to variable a1
        a2 ← EM[a]
    Display("the average deviation of candidate n°", a1)
    Display("is ", a2)
    Display("the average gaps of the tied candidates")
    For u from 0 to length(ListMin) do
        u1 ← Ind[u]
        u2 ← ListMin[u]
    Display("the average deviation of candidate n° ", u1)
    Display("current is:", u2)
    End for
End if
Add the minimum of the ListMin list to mn
If length(mn)=1 then
    Add the index of mn in mn1
    mn2 ← Ind[mn1]
    Display("So, the lucky winner of this election is candidate n◦", mn2)
Otherwise
    Display("The following candidates have the same mean deviation:")
    For v ranging from 0 to length(Ind) do
        v1 ← Ind[v]
    Display("the candidate n◦", v1)
    End For
    Display("so the vote is to be taken again. Thank you )
    End if
```

4.2. Python Program

Having worked out the algorithm for this voting system, we're now going to focus on the essentials, which include describing the code. To do this, we have used python version 5.3.1 to create the program. This can be described as follows:

Program 1 Voting method based on evaluation of mean deviation

```
@author: YIOGO Hadarou(yiogohadarou95@gmail.com)
""""
import numpy
import pandas as pd
import time as tps
import matplotlib.pyplot as plt

def MethodedeVote(L):
print("List of candidate ratings by voter \n\n",L)
Ind = []
ListMin = []
ListMax = []
CandNd = []#List of unwanted candidates
Ecart=numpy.zeros((len(L),len(L[0])))
L.sort(True)
Mediane=numpy.median(L, axis=1)# Calculating the median for each candidate

Mean =numpy.around(numpy.mean(L,axis=1),2)# Average for each candidate
for p in range(len(Mediane)):
if Mediane[p]==1 or Mediane[p]==1.5:
CandNd.append(p+1)

for i in range(len(L)):# Number of candidates
Liste=L[i]

for j in range(len(L[0])):
a=numpy.around(Liste[j]- Mean [i],2)
Ecart[i,j]=abs(a)
EM=numpy.around(numpy.mean(Ecart,axis=1),2)#    Calculation of average deviations
MaxL =max(Mean)
for i in range(len(Mean)):
```

```
if Mean [i] == MaxL:
ListMax.append(Mean [i])
Ind.append(i+1)
ListMin.append(EM[i])# List of average gaps between tied candidates
# ListCandEx.append(EM[Ind])
print("\nIncreasing filtering of candidate ratings\n\n", L)
print("\nThe number of voters is: ",len(L[0]))
print("\nThe number of candidates is: ",len(L))
print("\nThe following candidates:")
for n in range(len(CandNd)):
print("\nCandidate number",CandNd[n])
print("\nare not “loved” by the majority\n")
print("\nCalculation of the median for each candidate")
for m in range(len(L)):
print("\nThe median of candidate number{} est: {}".format(m+1,Mediane[m]))
print("\nCalculation of the average for each candidate")
for cm in range(len(L)):
print("\nThe average of candidate number{} is: {}".format(cm+1,Mean[cm]))
print("\nThe minimum average for this vote is: ",min(Mean))
print("\nThe maximum average of this vote is: ",max(Mean))

if len(ListMax)==1:
convtab=Mean.tolist()
T=convtab.index(max(convtab))
print("The lucky winner of this election is candidate number ", T+1)
else:
print("\nThere is a tie in this vote.\n\nlist of the ties is:\n")
for k in range(len(ListMax)-1):
print("\nCandidate number{};".format(Ind[k]))
print("\net\n\nlCandidate number{}.".format(Ind[len(ListMax)-1]))
print("\nThe deviation table is:\n\n",Ecart)
print("\nThe average deviation for each candidate is:")
for a in range(len(EM)):
print("\n The mean deviation of candidate numero{} is: {}".format(a+1,EM[a]))
print("\n The mean differences of the tied candidates")
for u in range(len(ListMin)):
C=ListMin[u]
print("\nThe mean deviation of candidate numero{} is {}".format(Ind[u],C))

if ListMin.count(min(ListMin))==1:
G=Ind[ListMin.index(min(ListMin))]
print("\nSo the lucky winner is candidate number", G)
else:
print("The following candidates have the same mean deviation:\n")
for v in range(len(Ind)):
print("\nCandidate number",Ind[v])
print("\nSo the vote must be taken again.\nThanks")
tps1 = datetime.now()
#d=  pd.read_csv(r"C:\Users\YIOGO\Desktop\projet\excel 1.csv", sep=";")
d=pd.read_csv(r"C:\Users\YIOGO\Desktop\projet\excel 2.csv",sep=";")
#d=pd.read_csv(r"C:\Users\YIOGO\Desktop\projet\excel 3.csv",sep=";")
#d=#pd.read_csv(r"C:\Users\YIOGO\Desktop\projet\excel 4.csv",sep=";")
#d=pd.read_csv(r"C:\Users\YIOGO\Desktop\projet\excel 5.csv",sep=";")
#d=pd.read_csv(r"C:\Users\YIOGO\Desktop\projet\excelperspectives.csv",sep=";")
d = d.values
# randomly generate a table
#d=numpy.random.randint(1,6,size=(5,1000))
# function to display the maximum averages corresponding to the tie-breakers
print(d)
# this function takes a list of values as input
MethodedeVote(d)
print("\n Machine time is:",tps.time()-tps1)
def Complexite(data):
Debut=tps.time()
MethodedeVote(data)
T=tps.time()-Debut
return T
def Temps(NbrCand,NbrVotant):
VX,TY=[],[]
for i in range(10,NbrVotant+10,10):
data=numpy.random.randint(1,6,size=(NbrCand,i))
VX.append(i),TY.append(Complexite(data))
plt.plot(VX,TY)
plt.xlabel(The number of voters)
plt.ylabel(The time machine '+str(NbrCand)+' candidate')
plt.show()
return VX,TY

NbrCand=20
for i in range(5,NbrCand+5,5):
print(Temps(i,3000))
```

4.3. Complexity Study

4.3.1. Theoretical Study

In sum, the worst-case time complexity of our Python code is $O(n^2)$. The cost of executing our code is therefore quadratic.

4.3.2. Graphical Analysis

After having analyzed the theoretical complexity of our program, we proceed to a graphical study by visualizing the results. To do this, we set the number of voters at 3000, then the number of candidates at 20. We then vary the number of candidates from 5 to 20 in steps of 5, and for each number of candidates selected, we vary the number of voters in turn from 10 to 3000 in steps of 10. For each fixed number of candidates, we obtain 300 tables of the candidates' scores, which we extract from their execution times and arrange in a Table called TY. Finally, four (04) different curves are represented of the Ty execution time as a function of the number of voters, as follows:

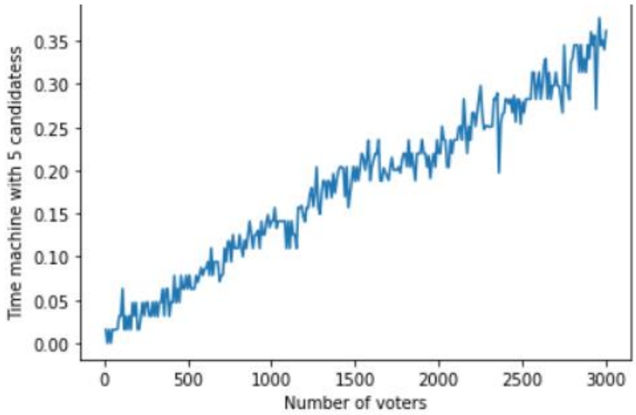

Figure 1. *Graphical complexity of an election with 5 candidates and numbers ranging from 10 to 3,000.*

The curve in Figure 1 is obtained by setting the number of candidates to 5 and then varying the number of voters from de 10 to 3000 in steps of 10. So, we have a voter simulation of 10 voters and 5 candidates, 20 voters and 5 candidates, 30 voters and 5 candidates and so on, up to 3000 voters and 5 candidates.

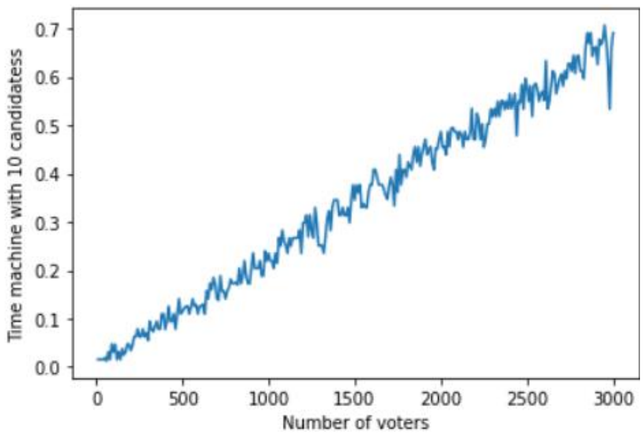

Figure 2. *Graphical complexity of an election with 10 candidates and numbers ranging from 10 to 3,000.*

The curve in Figure 2 is obtained by setting the number of candidates to 10 and then varying the number of voters from de 10 to 3000 in steps of 10. So, we have a voter simulation of 10 voters and 10 candidates, 20 voters and 10 candidates, 30 voters and 10 candidates and so on, up to 3000 voters and 10 candidates.

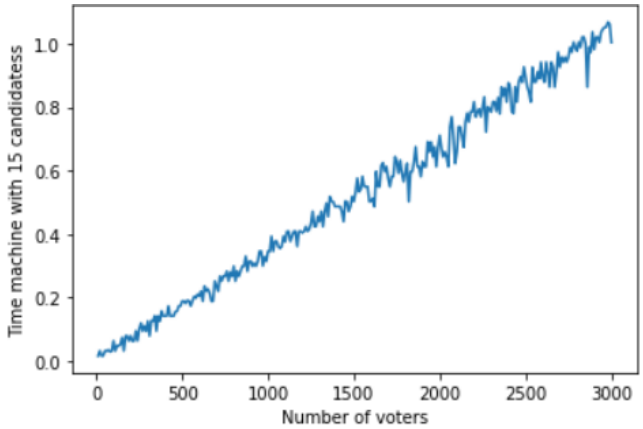

Figure 3. *Graphical complexity of an election with 15 candidates and numbers ranging from 10 to 3,000.*

The curve in Figure 3 is obtained by setting the number of candidates to 15 and then varying the number of voters from de 10 to 3000 in steps of 10. So, we have a voter simulation of 10 voters and 15 candidates, 20 voters and 15 candidates, 30 voters and 15 candidates and so on, up to 3000 voters and 15 candidates.

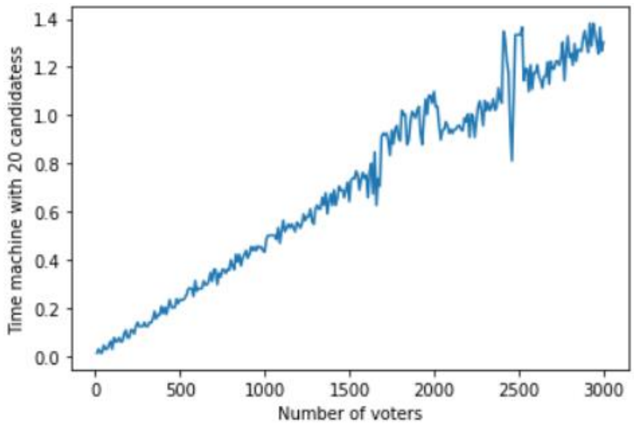

Figure 4. *Graphical complexity of an election with 20 candidates and numbers ranging from 10 to 3,000.*

The curve in Figure 4 is obtained by setting the number of candidates to 20 and then varying the number of voters from de 10 to 3000 in steps of 10. So, we have a voter simulation of 10 voters and 20 candidates, 20 voters and 20 candidates, 30 voters and 20 candidates and so on, up to 3000 voters and 20 candidates.

The four figures represent curves that shows a linear trend as the number of candidates and voters varies. This shows the accuracy of the study of the program's theoretical complexity.

In short, based on these results and on data of reasonable size, our algorithm is satisfactory, with a linear trend in the representation of the various curves. Whatever the increase in the number of voters and electors, the algorithm will provide us with a conclusive result with an almost linear curve.

5. Numerical Application

5.1. Example 1

In this section, we present examples with the votes cast by voters for each candidate, and implement the method efficiently by using our program to solve the problem.

Let's consider Table 1 from [10], where we have 11 voters and 6 candidates. After feeding this data into our code program, we obtain the results below:

The number of voters is: 11

The number of candidates is: 6

The following candidates:

Candidate n◦ 1

Candidate n◦ 5

Are not “liked” by the majority.

Calculate the median of each candidate.

The median of candidate n◦1 is: 1.0

The median of candidate n◦2 is: 3.0

The median of candidate n◦3 is: 5.0

The median of candidate n◦4 is: 3.0

The median of candidate n◦5 is: 1.0

The median of candidate n◦6 is: 4.0

Calculation of the average for each candidate

The average of candidate n◦1 is: 2.45

The average of candidate n◦2 is: 2.82

The average of candidate n◦3: 3.18

The average of candidate n◦4: 2.45

The average of candidate n◦5: 2.82

The average of candidate n◦6 is: 3.18

The minimum average for this vote is: 2.45

The maximum average of this vote is: 3.18

There are ties in this vote.

The list of ties is:

Candidate n◦3 and candidate n◦6.

In these results, we see that the median of Candidate 1 and Candidate 5 is equal to 1, which means that these two candidates are not appreciated by at least 50 percent of voters, so they can be eliminated from the electoral list.

These results also show two maximum averages, belonging to candidates 3 and 6, so the arithmetic mean cannot be used to decide between these candidates, so we need to calculate the mean deviation of these two candidates. Whoever has the smallest mean deviation is the best candidate.

The table of gaps is:

Table 8. *Candidate gap table.*

Candidate 1	1.45	1.45	1.45	1.45	1.45	1.45	1.45	2.55	2.55	2.55	2.55
Candidate 2	1.82	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
Candidate 3	2.18	2.18	2.18	2.18	2.18	1.82	1.82	1.82	1.82	1.82	1.82
Candidate 4	1.45	0.45	0.45	0.45	0.45	0.55	0.55	0.55	0.55	0.55	0.55
Candidate 5	1.82	1.82	1.82	1.82	1.82	1.82	2.18	2.18	2.18	2.18	2.18
Candidate 6	2.18	2.18	2.18	1.18	0.82	0.82	0.82	0.82	0.82	1.82	1.82

Table 8 shows the standard deviation table and is used to calculate the mean deviation for each candidate.

The average deviation of each candidate is:

The mean deviation of candidate n◦1 is: 1.85

The mean deviation of candidate n◦ 2 is: 0.33

The mean deviation of candidate n◦ 3 is: 1.98

The mean deviation of candidate n◦ 4 is: 0.6

The mean deviation of candidate n◦ 5 is: 1.98

The mean deviation of candidate n◦ 6 is: 1.41

Mean deviations of tied candidates

The mean deviation of candidate n◦3 in coure is: 1.98

The mean deviation of candidate n◦ 6 in coure is: 1.41

Therefore, selected candidate based on the computed criteria of this election is candidate n◦6

Machine time is: 0:00:00.015629

Note that when using the same table of candidate scores, the result is identical to that solved manually in [10] (see section 3). This proves the efficiency and accuracy of our computer program. These satisfactory results lead us to carry out several tests on other examples to confirm the effectiveness of our work.

Let's now test our program on an example of a vote involving 4 candidates and 100 voters, to obtain the following table of candidate scores awarded by voters:

Note that when using the same table of candidate scores, the result is identical to that solved manually in [10] This proves the efficiency and accuracy of our computer program. These satisfactory results lead us to carry out several tests on other examples to confirm the effectiveness of our work.

Let's now test our program on an example of a vote involving 4 candidates and 100 voters, to obtain the following table of candidate scores awarded by voters:

5.2. Example 2

This example is a data simulation.

Table 9. *Candidate scores awarded by 100 voters and arrange in ascending order.*

Candidate 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Candidate 2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Candidate 3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Candidate 4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

So, after entering the data in our study, the program will, as before, provide the results after execution. We obtain the following results:

Number of voters: 100
The number of candidates is: 4
Candidate n◦ 1 is disliked by the majority
Calculation of the median for each candidate
The median of candidate n◦1 is: 1.0
The median of candidate n◦ 2 is: 3.5
The median of candidate n◦ 3 is: 4.0
The median of candidate n◦ 4 is: 2.0
Calculation of each candidate's average
The mean of candidate n◦ 1 is: 1.0
The average of candidate n◦ 2 is: 3.5
Candidate n◦ 3's average is: 3.5
The average of candidate n◦ 4 is: 2.5
The minimum average for this vote is: 1.0
The maximum average of this vote is: 3.5
There are ties in this vote.
The list of ties is:
Candidate n◦ 2 and candidate n◦ 3.
The average deviation of each candidate is:
The mean deviation of candidate n◦1 is: 0.0
The mean deviation of candidate n◦2: 0.5
The mean deviation of candidate n◦3 is: 0.75
The mean deviation of candidate n◦4 is: 1.5
Mean deviations of tied candidates
The mean deviation of candidate n◦2 is: 0.5
The mean deviation of candidate n◦3 in coure is: 0.75

Therefore, selected candidate based on the computed criteria of this election is candidate n◦2

Machine time is: 0:00:00.046881.

5.3. Example 3

This example is taken from and solved with a VMAVA voting method computer program in [13]. The aim is to compare the results obtained with those provided by our computer program.

Table 10. *Example of 4 Candidates and 5 voters.*

	C1	C2	C3	C4
V1	3	4	2	4
V2	4	1	3	2
V3	3	2	4	2
V4	3	2	1	2
V5	4	1	3	1

By implementing the VMAVA method, Candidate 1 is elected see [13]. This method has also been studied in [14] where the accuracy of these results has been demonstrated.

Using our computer model, we obtain the following results (see tables 11 and 12):

Table 11. *Ascending filtering of candidate scores.*

	V1	V2	V3	V4	V5
C1	3	3	3	4	4
C2	1	1	2	2	4
C3	1	2	3	3	4
C4	1	2	2	2	4

Table 11 will enable us to calculate the median for each candidate. If a candidate's median is equal to 1 or 1,5 depending on the parity of the number of voters, then this candidate is note appreciated by at the least 50 percent of the population and can therefore be removed from the list of candidates.

After running the program, we obtain the following table:

Table 12. *Table of medians and averages for each candidate.*

C1	mediane	Arithmetic mean
C2	3,0	3,4
C3	2,0	2,0
C4	3,0	2,6
C5	2,0	2,2

The maximum average of this vote is 3, 4 which is that of candidate 1.

We can see from the results that the median con not be used to distinguish between candidates, so we have to calculate their arithmetic mean. All candidates have a distinct arithmetic mean, so the one with the highest arithmetic mean is the best candidate. Therefore, selected candidate based on the computed criteria of this election is candidate n◦1.

We no longer have to calculate the mean deviation. In this example, the method coincides with the MMCM method developed in [15].

6. Conclusion and Outlook

In this work, we have presented the literature on different voting methods, the social choice function and the desirable properties of a social choice function. This has helped to strengthen our knowledge of the various existing voting systems. We then described a voting method based on the evaluation of mean deviation. Following this, we proceeded to an implementation that represents the main goal of our work, where a few examples of choice problems were solved through this code in order to demonstrate its accuracy. The comparison of these results with those of the voting method based on the evaluation of the mean deviation and the VMAVA method not only showed the accuracy of our work but also presented a more interesting result than that of these existing methods. As with any research work that has its shortcomings, ours has not been spared. A number of difficulties were encountered during our programming, as described below.

Our code doesn't eliminate candidates disliked by at least half the population, whereas removing them could reduce the computations and could also reduce the algorithm's execution time. When we have candidates with the same maximum averages and mean deviations which is a fairly rare case, elections must necessarily be repeated. how to get the best candidate without re-running the elections? Our future research will not only address these shortcomings in order to improve the efficiency of our computer program and the efficient handling of very large data sets, but will also also study time complexity in greater depth by comparing execution time with a benchmark and including a regression equation to confirm the linear nature of the execution time curves.

Abbreviations

VMAVA	Voting Method Based on Approval Voting and Arithmetic Mean
MMCM	Mean-Median Compromise Method

Author Contributions

Hadarou Yiogo: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing

Zo ïnabo Savadogo: Resources, Supervision, Validation

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] N. Bakhta. Multi-agent model for the design of collective decision support systems. thesis, University of Oran. pages 542-569, 2013-2014.

[2] P. Blanchenay. Voting paradoxes and voting methods in france. the master's thesis,' high school commercial studies major 'economie. 2004.

[3] Adama Coulibaly. Group decision, facilitation aid: adjusting voting procedures according to the decision context. artificial intelligence [cs.ai]. university toulouse 1 capitole (ut1capitole). 2019.

[4] Veera P. Darji and Ravipud V. Rao. Application of ahp/evamix method for decision making in the industrial environment; s.v. national institute of technology, surat, india. (3): 542-569, 2013.

[5] Kangashe Jean-Louis Esambo. Strategic vote choice in one round and two round elections. Political research quaterly. 4(3): 637-645, 2010.

[6] Kangashe Jean-Louis Esambo. Congolese electoral law. academia-harmattan., louvain-la-neuve. 2014.

[7] Benny Geys. Rational theories of voter tournout: A review. political studies review. (4): 16-35, 2006.

[8] Jacobs M. Sustainable development as a contested concept, in dobson, a. fairness and futurity: Essays on environmental sustainability and socialjustice. oxford: Oxford university press. pages 27-59, 1999.

[9] Zo ïnabo Savadogo. Contributions to collective aggregation in the multicriteria decision support problem. page 19.

[10] Zo ïnabo Savadogo, Abdoulaye Compaore, and Pegdwindé Ousséni Fabrice Ouedraogo. Voting method based on an average gap. 12(3): 1176-1186, 2019.

[11] Zo ïnabo Savadogo, Sougoursi Jean Yves Zaré, Wambie, Zongo, Somdouda Sawadogo, Blaise Somé. New Innovative Method in the Field of Social Choice Theory. *Pure and Applied Mathematics Journal*. Vol. 10, No. 6, 2021, pp. 121-126. https://doi.org/10.11648/j.pamj.20211006.11

[12] Zo ïnabo SAVADOGO, Sougoursi Jean Yves ZARE, Wambie ZONGO, Blaise SOME. New Voting Method Adapted to Developing Countries (NoMePaVD). *Pure and Applied Mathematics Journal*. Vol. 12, No. 1, 2023, pp. 12-15. https://doi.org/10.11648/j.pamj.20231201.12

[13] Koumbèbarè Kambiré, Zo ïnabo Savadogo, Frédéric Nikiéma. Implementation of the VMAVA Method in Order to Make Applications with a Large Number of Candidates and Voters. *Pure and Applied Mathematics Journal*. Vol. 12, No. 3, 2023, pp. 49-58. https://doi.org/10.11648/j.pamj.20231203.12

[14] Wambie Zongo, Zo ïnabo Savadogo, Sougoursi Jean Yves Zare, Somdouda Sawadogo and Blaise Some, VMAVA+: (Voting Method based on Approval Voting and Arithmetic mean) +, Advances and Applications in Discrete Mathematics 35 (2022), 87-102. http://dx.doi.org/10.17654/0974165822054

[15] Ngoie, R.-B. M.; Kasereka, S. K.; Sakulu, J.-A. B.; Kyamakya, K. Mean-Median Compromise Method: A Novel Deepest Voting Function Balancing Range Voting and Majority Judgment. Mathematics 2024, 12, 3631. https://doi.org/10.3390/math12223631