

# Indepth analysis of a variant of Majority Voting and relation to Zermelo-Fraenkel Set Thoery With Axiom of Choice (ZFC) - Updated Draft

*SrinivasanKannan(alias)Ka.Shrinivaasan(alias)ShrinivasKannan  
IndependentOpenSourceDeveloper, ResearcherandConsultant*

*Ph : 9789346927, 9003082186, 9791165980*

*KrishnaiResearchOpenSourceProducts : [http : //sourceforge.net/users/ka\\_shrinivaasan](http://sourceforge.net/users/ka_shrinivaasan),*

*[https : //www.ohloh.net/accounts/ka\\_shrinivaasan](https://www.ohloh.net/accounts/ka_shrinivaasan)*

*ResearchWebsite : [https : //sites.google.com/site/kuja27/](https://sites.google.com/site/kuja27/)*

*(ka.shrinivaasan@gmail.com, shrinivas.kannan@gmail.com, kashrinivaasan@live.com)*

February 8, 2014

## Abstract

As a generalization on Majority voting and pseudorandom choice, a BPNC majority voting circuit with oracle access to a choice function is presented and the choice function (which in the infinite case would be a choice function for ZFC) is defined in this article.

## 1 Majority voting circuit and pseudorandom choice are in BPNC

It is a known result that  $BPNC^k$  is contained in BPP. Earlier a majority voting circuit with BPP voter oracle circuit lying in  $NC1^{BPP}$  was explained. It was also mentioned that instead of an oracle voter, the majority voting circuit can be formulated alternatively as a non-uniform BPNC circuit. But pseudorandom choice was shown to be in BPP using Nisan's Pseudo random generator. Since BPNC is contained in BPP, this probably gives a counterintuitive result that majority voting is more efficient compared to a pseudorandom choice but both have a same bounded error derived using P(Good) expression. But recently there are pseudorandom generators constructed in NC0 with linear stretch. Thus there is a symmetry for the LHS and RHS of the P(good) equation with both Pseudorandom choice with bounded error and Majority voting with bounded error lying in BPNC class. But the caveat is that the majority voting is non- uniform. This is an improved bound as against  $NC1^{BPP}$  oracle circuit for majority voting and BPP machine for pseudorandom choice.

## 2 Majority and Parity

By the result of Furst,Saxe and Sipser Parity is not in AC0. It is also proved by Hastad's switching lemma implying Parity cannot be computed by constant depth circuits of polynomial size and unbounded fanin. There is a simple reduction from parity to majority function where sum of bits are computed by a 3-to-2 way addition of 3 numbers and parity (XOR) is used in the computation. Thus parity function forms the building block of entire majority voting circuit. Since parity is not computable in AC0, the majority circuit with bounded error cannot be computed in constant depth and this justifies the BPNC bound obtained above.

### 3 Zermelo-Fraenkel-Axiom-of-Choice (ZFC) and its relevance to majority voting

Axiom of choice in set theory postulates that given a set  $X$  (possibly infinite) of non-empty sets, there always exists a choice function that chooses one element from each element-set in  $X$ . Though existence of choice function for infinite case is controversial, AOC has been pushed into ZermeloFraenkel Set theory to make it ZFC set theory (ZF with AOC). Now let us consider a majority voting scenario where the voting population is partitioned into subsets called constituencies and there exists a choice function that chooses one representative element from each constituency. Thus set of representatives from each constituency chosen by choice function form an electoral collegium denoted as EC. This EC can be modelled as a majority voting circuit with each chosen representative of EC as an input to the majority circuit. Thus we have a voting system with two levels.

### 4 Definition of a choice function and circuits for choice function computation

Let  $m$  be the size of voting citizens in a constituency and  $q \ll m$  be the number of candidates.

Each of the  $q$  candidates get votes  $v(i)$  for  $i=1,2,3,\dots,q$ .  
Thus  $v(1) + v(2) + v(3) + \dots + v(q) = m$

Above is a partition of voting population. The choice function  $C$  is defined as follows:

$C(m,q) = \text{index maxI of the biggest valued part in the integer partition of } m \text{ as defined above, where } 1 < \text{maxI} < q$

By Axiom of Choice, there exists a choice function for choosing a representative from each constituency in infinite case. But it needs to be well-defined that the above choice function always returns a unique index of biggest part in the partition. Unfortunately partition of votes as defined by the choice function can have multiple similar valued biggest parts (which requires a tie- breaker criterion) and might return a set in the absence of a tiebreaker. Thus any voting system similar to above choice function allowing a tie has an inherent limitation. A turing machine computing the choice function for a constituency takes as input the voting population and candidates, initiates a voting on candidates, computes a partition of the voting population into votes for individual candidates, sorts the parts in the partition and outputs the index of the candidate with largest vote (assuming a tiebreaker). Aggregating votes for candidates involves addition NC1 circuits for each of the candidates where voters cast their vote by giving inputs into the NC1 circuit of the corresponding candidate. Outputs of each of the addition circuits form a partition of the voting population. These outputs are parts in the partition which can be sorted using a sorting network to get the largest indexed candidate part. For voting with bounded error these addition NC1 circuits are BPNC circuits with error and feed to a sorting network of polynomial size and logdepth (AKS sorting circuits). Thus the choice function oracle circuit is in BPNC of polylog depth and polynomial size.

## 5 Circuit for majority voting with constituencies as described above

The non-uniform NC1 majority voting circuit can be modified to query a choice function oracle described above and get the inputs(representatives) for majority gate collegium. Using BPNC choice function described above as oracle for electoral collegium, we obtain a  $BPNC^{BPNC}$  circuit. It is not known if BPNC is low for itself like BPP. Thus a BPNC circuit with oracle access to a BPNC choice function has been arrived at for the special case of majority voting with constituencies and electoral collegium.

## 6 Future directions

Handwritten notes on perfect voter decidability and relation to Arrow's Impossibility Theorem, NP-Completeness of Democracy, Relation between hash functions and integer partitions by the author in references below might further strengthen the above result. For instance, for above Choice function, a hash function that unequally distributes is needed (no two keys have equally long buckets of values hashed into them). Thus number of such hash functions indirectly give all possible voting patterns with unique elected choice. If coalitions are also mathematically defined, more fine-grained complexity model for practical multiparty majority voting can be obtained.

## 7 Acknowledgements

I dedicate this article to God.

## 8 Bibliography

### References

- [1] On Type-2 Probabilistic Quantifiers, Automata, Languages and Programming: 23rd international colloquium, ICALP
- [2] Pseudorandom generators with linear stretch in NC0, Benny Applebaum, Yuval Ishai, Eyal Kushilevitz, Computer science department, Technion, Israel.
- [3] Various resources on Choice functions and Zermelo-Fraenkel-Axiom-of-Choice set theory
- [4] TAC 2010 - Update summarization through interview algorithm - [http : //www.nist.gov/tac/publications/2010/participant.papers/CMI.IT.proceedings.pdf](http://www.nist.gov/tac/publications/2010/participant.papers/CMI.IT.proceedings.pdf)
- [5] Few Algorithms for ascertaining merit of a document and their applications - [http : //arxiv.org/abs/1006.4458](http://arxiv.org/abs/1006.4458)
- [6] Various resources on Integer Partitioning
- [7] Parity not in AC0 - Furst, Saxe and Sipser
- [8] Hastad's Switching Lemma, John Hastad's PhD thesis, MIT
- [9] Majority is in non-uniform NC1, Mix Barrington
- [10] Sorting networks of Ajtai-Komlos-Szemerédi

- [11] Integer Partitions and Hash Functions - [https : //sites.google.com/site/kuja27/IntegerPartitionAndHashFunctions](https://sites.google.com/site/kuja27/IntegerPartitionAndHashFunctions)
- [12] Informal notes - 3 : on Minimum Convex Hulls of Implication Random Growth Networks and Perfect Voter Decidability - [https : //sites.google.com/site/kuja27/ImplicationRandomGraphConvexHullsAndPerfectVoterProblem2014-01-11.pdf?attredirects = 0](https://sites.google.com/site/kuja27/ImplicationRandomGraphConvexHullsAndPerfectVoterProblem2014-01-11.pdf?attredirects=0)
- [13] Informal notes - 2 : on Minimum Convex Hulls of Implication Graphs and Hidden Markov Model on class nodes of Concept Hypergraph - [https : //sites.google.com/site/kuja27/NotesOnConceptHypergraphHMMandImplicationGraphConvexHulls2012-12-30.pdf?attredirects = 0](https://sites.google.com/site/kuja27/NotesOnConceptHypergraphHMMandImplicationGraphConvexHulls2012-12-30.pdf?attredirects=0)
- [14] Informal notes - 1 : on Implication Graphs, Error probability of Majority Voting and P Versus NP Question [http : //sourceforge.net/projects/acadp drafts/files/ImplicationGraphsPGoodEquationAndPNotEqualToNP](http://sourceforge.net/projects/acadp drafts/files/ImplicationGraphsPGoodEquationAndPNotEqualToNP)
- [15] Lower Bounds for Majority Voting and Pseudorandom Choice - [https : //sites.google.com/site/kuja27/LowerBoundsForMajorityVotingPseudorandomChoice.pdf?attredirects = 0](https://sites.google.com/site/kuja27/LowerBoundsForMajorityVotingPseudorandomChoice.pdf?attredirects=0)
- [16] Circuit For Computing Error Probability of Majority Voting and Pseudorandom Choice - [https : //sites.google.com/site/kuja27/CircuitForComputingErrorProbabilityOfMajorityVoting.pdf?attredirects = 0](https://sites.google.com/site/kuja27/CircuitForComputingErrorProbabilityOfMajorityVoting.pdf?attredirects=0)
- [17] Interview Algorithm is in  $IP=PSPACE$  - [https : //sites.google.com/site/kuja27/InterviewAlgorithmInPSPACE](https://sites.google.com/site/kuja27/InterviewAlgorithmInPSPACE)