Research Statement for Laurence L. Leff, Ph.D.[1]

Web Development, Human Factors Experiments and Simulations (Experimental Algorithmics) for some Techniques in Extreme Participatory Democracy

In all democracies[2], people are becoming disenchanted with the democratic system. They distrust their politicians. The citizens watch democracy as a spectator sport. Their participation, voting about once a year or so, seems insignificant to them.

In 2008, I began work on programming for participatory democracy. Based on ID3, Genetic Algorithms, and Approval Voting, the entire electorate iteratively develops and decides on complex statutes: the country's penal code, tax code, and the codes setting how people qualify for social benefits such as social security or a military pension. The other effort is to develop a Constitution Construction Kit[3]. (After several years, I intend to include the techniques for participatory development of a complex statute within the Constitution Construction Kit.)

As developing a budget has been a problem for the last two decades in the United States at both the federal level and for individual states, I am excited about two ideas I have by which the people directly, rather than a legislature, can deal with budget issues and social programs.

Polls show that the majority of United States citizens want a national initiative process. Citizens want more referendums. Twenty-eight states support some kind of direct initiative or referendum. But these are perceived, perhaps correctly, as captured by special interests. Those proposing an initiative are required to submit petitions with a large number of signatures of registered Voters; a small industry has developed to gather them.

I observed another problem: referendums are take-it-or-leave-it propositions. The citizens cannot participate in developing the proposition on which to vote. Hence I allow the entire electorate develop a complicated body of law: a tax plan and budget, a penal code, a Constitution.

Constitutions have particularly troubled me. Several countries in the Arab Spring have overthrown their dictators, and, of course, after September 11, 2001, the United States removed the dictators in Iraq and Afghanistan. An elite developed a Constitution to be ratified on a take-it-or-leave-it basis. The electorate in a country should be able to choose among several constitutions. But first they should participate in simulating them so they can make a meaningful choice.

Thus, I propose a Constitution Construction Kit—a multi-player role-playing game. Any voter or group could propose a Constitution. Then, Users would play the roles of Voters, those holding office, and those working in government. The "Dungeon Master," whom I call the Reality-Detail-Filler-in, would propose situations that the government would have to deal with. This would stress-test the proposed Constitutions. These could include a disaster such as an asteroid hitting the country, a boon such as the discovery of oil wealth, or a situation that would exacerbate conflict among ethnic or racial groups such as the shooting of a child of one Ethnic Group by a police officer of another.

---

1 Excellent assistance provided by Will Pittenger with this document. See Appendix II: Acknowledgment

2 Olsson, A. R "Electronic Democracy and Power" in *EGOV 2004* LNCS 3183, R. Traunmiller (Editor), Springer-Verlag, Berlin, Heidelberg, 7-14, 2004.

3 Leff, L. "Constitution Construction Kit Requirements Specification" available at www.wiu.edu/users/mflll/CR.odt and "Constitution Construction Kit Executive Summary" www.wiu.edu/users/mflll/CE.odt

Of course, one is aware of the whole concern over voters being ill-informed, prone to demagoguery and even “mob rule.” However, the Deliberation Project at Stanford shows that randomly-chosen groups of citizens can successfully deliberate on public issues. They can interact with experts and will change their opinions and become more interested in politics as informed voters.

1. Computational Social Choice Voting Background

We are fortunate that many fine researchers have studied voting systems with more than two candidates. As we know from the Nader-Bush-Gore election, many people are faced with the unpleasant choice of voting either for the lesser-of-two-evils who has a chance to win, or for the one they truly want to win, their first preference. Voting for a less-favored candidate to prevent a more-disliked candidate from winning is called manipulating the vote. Over the centuries, researchers and others have proposed techniques for selecting a winner given the expressed preferences. This is the social choice function. In the United States, we use plurality voting; each voter gets a single vote to assign to a single candidate; the one with the most votes wins. There have been many other proposed techniques. Several are what Conitzer termed scoring systems and Wally Smith termed COAF systems. In these voting systems, each voter assigns a number to each candidate. The social choice system sums the numbers for each one. The one with the highest total is the winner.

	A	B	C
	5	4	4
	1	4	2
	5	8	1
	5	2	2
	3	3	4
	4	3	1
Total	23	24	14

Table 1: COAF Example For Range Voting

	A	B	C
	1	0	0
	0	1	0
	0	1	0
	1	0	0
	0	0	1
	1	0	0
Total	3	2	1

Table 2: COAF Table For Plurality Voting

The most general version of this is Range Voting[5]. Here each voter may use any number between zero and ten, the range. More specific is approval voting, where each voter can only enter zero or one for each voter, but may vote for more than one candidate. For example: Assume there are three candidates, **A**, **B** and **C** and six Voters. Table 1: COAF example for Range Voting shows that candidate **B** would win with the highest total.

The most familiar voting system, usually the one used in the United States, is simple majority voting or plurality voting. In the COAF framework, each voter is able to enter a one for only one candidate. All other candidates receive zero from that voter. Table 2: COAF table for Plurality Voting shows the plurality voting system in which each voter votes for their most preferred candidate from Table 1—as you can see there is a different winner.

Another possible voting system is "bullet voting," in which Voters enter ones for all the candidates except one; in other words, they can knock the one they hate.

The other broad class of social choice functions has each voter rank the candidates in preference order. A popular one involves seeing if one candidate has a majority based on the first preference of the voters. That is, did 50$^{\%}$ of the voters give their first preference as candidate *X*? If yes, then that person is the winner. If not, then find which candidate has the least votes as first preference. Eliminate that candidate and shift the choices for the Voters who voted for them so their second preference is now the first, their third is the second, etc. Is there a candidate with 50$^{\%}$ of the vote based on the new first preference? If not, go back to the beginning and repeat. This is the Single Transferable Vote[6]. Assume, for example, that we have eight Voters and four

5 http://uthreee.blogspot.com/2010/11/wally-smith-on-range-voting-thoughtful.html

6 Conitzer, V., Sandholm, T., Lang, J. "When are Elections with Few Candidates Hard to Manipulate?" *Journal of the ACM*, Volume 54 Number 3

candidates: **A**, **B**, **C**, and **D**. Table 3: Single Transferable Vot votes. **D** has only one number one ranking, so is eliminated. T the second group shown in Table 3. Here, **A** has the fewest f eliminating that column, that leaves the assumed votes in the f winner is thus **C**, with five first-place Voters.

The techniques I propose use Multi-Candidate voting their processing. This is particularly true for the first technique Decision Tree System. Of course, I want to do experiment voting choice mechanisms work best in the contexts here.

Initial Rankings				Second Round			Final Round	
A	B	C	D	A	B	C	B	C
3	1	2	4	3	1	2	1	2
4	2	3	1	3	1	2	1	2
2	1	4	3	2	1	3	1	2
4	3	1	2	3	2	1	2	1
1	4	2	3	1	3	2	2	1
4	3	1	2	3	2	1	2	1

2. *The Systems*

2.1. *The ID3 Decision Tree System*

Decision Trees represent a penal code, a tax code or a social benefit code. There are three types, one where the result is simply true or false. For example, the citizens might simply vote on whether a person in a given situation may or may not legally possess the Type Of Gun indicated. The ID3 Decision Tree in Figure 1: A Decision Tree Representing Part of a Penal Code exemplifies the second type, where each Node is given a Result. As the ID3 Decision Tree Process forms the tree, Voters will choose a from a list of possible Result classes for each Node. For the third and most interesting type, Voters choose a mathematical Formula involving the four functions (+, –, ÷, x) and a set of parameters. For the United States tax code, they would include quantities such as the total income, percentage earned from capital gains, percentage earned from wages and the number of children.

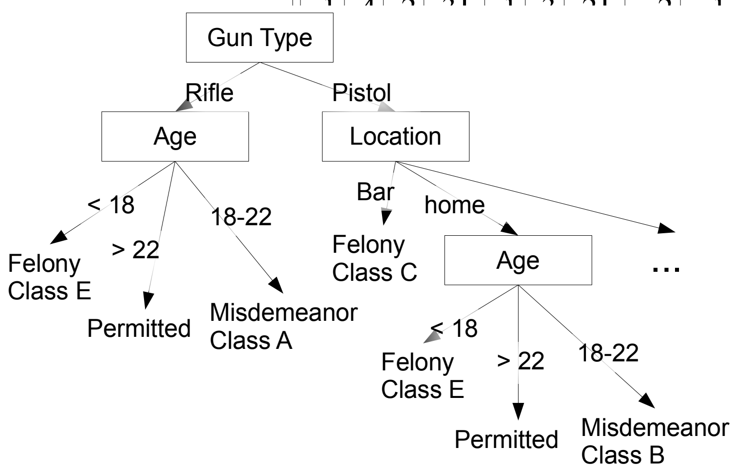


Figure 1: A Decision Tree Representing Part Of A Penal Code

As the Decision Tree is formed, I term the words from which the Users can choose Titles. In the penal code example, it might be Gun Type, Location of the gun and possessor, and Conviction status of the person possessing the gun. The categories for each Title would be the Attributes. For example, for Gun Type, they might be pistol, antique gun, bb-gun, rifle, shotgun, or assault rifle.

This is an extensive game with perfect information and simultaneous moves[7]. At every stage of forming the tree, the Voters can choose, for any of the open Nodes, to expand the Node. They can cast a vote for each possible Title. (Those are the Titles not chosen for any of the

7 Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory* MIT Press, Cambridge Massachusetts, 1994, page 102-3

Nodes on the path from the Root Node to the Node in question. For each Node, this is a Multi-Candidate election, as described above.)

For all except the Root Node, there is one more option. That is to have a vote for the Result. If the option to vote for the result does not win, when everyone has voted for the Title for that Node, the Node will expand and new Nodes will be created. Of course, there will be another Multi-Candidate election for each of these new Nodes.

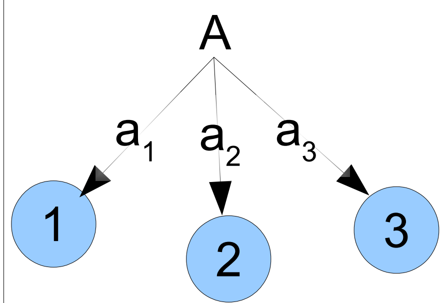


Figure 2: Interactively Voting On An ID3 Decision Tree

For example, assume that there are three Titles: **A**, **B**, and **C**. Also assume they have three Attributes each, a_1, a_2, a_3 and b_1, b_2, b_3 and c_1, c_2, c_3. Error: Reference source not found shows this tree in progress. At Node 1, the Users would be presented with a ballot with **B** and **C**. Those Users would then "vote for Result." **B** won. So the system would expand that Node with b_1, b_2, b_3. See Figure 3: After the vote on expanding Node 2 in an ID3 Decision Tree. For each of these three Nodes, there would be a vote with two options, expand, or vote for a Result.

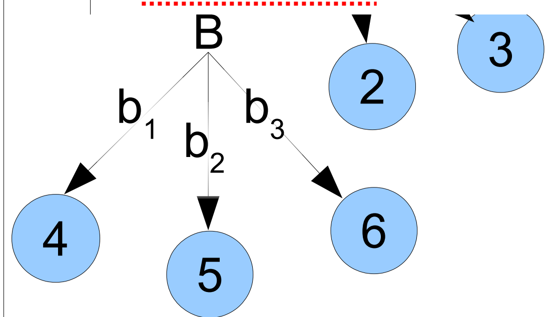


Figure 3: After The Vote On Expanding Node 2 In An ID3 Decision Tree

2.1.1. ID3 Decision Tree Process With Formulae

As mentioned earlier, the most interesting thing to do with ID3 is to have Formulae at the edge. Voters can propose their own Formulae. The tax or benefit would use the median of the Result of all the different proposed Formulae. Of course, some Voters would just vote for another proposed Formula—each Formula would be weighted by the number of votes, as if each person who voted for it had proposed it themselves.

Each Formula can include Median Parameters: e.g., the tax could be $m_1 I - m_2 C$, where I is the income, C is the number of children and m_1, m_2 are Median Parameters.

Whenever they want, Voters enter the values they prefer for the coefficient. The value used is the median of all the ones entered. (Of course, as citizens pass on or leave the country or jurisdiction, their values are deleted from consideration.)

Hysteresis is provided, so the tax structure doesn't change too fast. The maximum change per year is limited, say to 5$^{\%}$ per year. Thus, even if all the Users were to change the coefficient to zero, the value used would only go down by 5$^{\%}$.

2.2. Genetic Algorithms

Another approach is inspired by Genetic Algorithms, a technique used to optimize. For example, Genetic Algorithms have been used to engineer the size of the I-beams in a bridge and the location of the joints connecting them. There the program seeks to optimize a Formula based on the ability of the bridge to withstand loads as well as on its cost—in other words, it finds the cheapest bridge that will still stay up.[8]

		position<***S***-1									***S***-1	*S*	position>***S***		
		Genes that aren't swapped										*Swapped Genes*			
Generation #	**Gamete**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
1	**A**	1.7	2.3	9.9	8.2	11	9.2	4.2	5.6	3.7	2.2	3.5	4.8	6	4.6
	B	3.2	0.21	67.2	8.1	3.2	2.5	2.6	7.2	6.2	6.4	4.3	4.2	1.2	9.4
2	**A′**	1.7	2.3	9.9	8.2	11	9.2	4.2	5.6	3.7	2.2	4.3	4.2	1.2	9.4
	B′	3.2	0.21	67.2	8.1	3.2	2.5	2.6	7.2	6.2	6.4	3.5	4.8	6	4.6

Table 4: Illustration Of Crossover In Genetic Algorithms

In each cycle, the Genetic Algorithm combines some features of the bridges that have high fitness—are closest to satisfying the desired criteria. These are analyzed against those criteria, and in turn features from both parts are randomly mixed. In engineering a bridge, the Genetic Algorithm designer assigns the thickness of each beam and the x and y position of each connection point to a position on a Chromosome. Each Chromosome represents one possible design. Each of the values on the Chromosome is termed a Gene. For example, a Gene might represent the *x*-value for the position of Node 17 or the thickness of girder 22.The process of

8 Adeli H. Kumar S. "Distributed Genetic Algorithms for Structural Optimization" *Journal of Aerospace Engineering* 8(3) 156-163.

combining the Chromosomes is termed "crossing over" and is inspired by the meiosis process in sexual reproduction that sets the Chromosomes that will appear in spores or gametes such as sperm and eggs. To cross over, the system chooses a position randomly in the Chromosome. That is, if there are *n* Genes in the Chromosome, the system chooses a random number, ***S***, from 1 to *n*. The new Chromosome contains the first ***S*** Genes from one high-fitness Chromosome and Genes $S+1$ through *n* from the other Chromosome. Table 4: Illustration of Crossover in Genetic Algorithms shows a possible crossover between two Chromosomes at position ***S***.

In engineering a bridge, the Genetic Algorithm designer assigns the thickness of each beam and the x and y position of each connection point to a position on a Chromosome[9]. Each Chromosome represents one possible design. Each of the values on the Chromosome is termed a Gene. For example, a Gene might represent the *x*-value for the position of Node 17 or the thickness of girder 22. For example, a Gene might represent the *x*-value for the position of Node 17 or the thickness of girder 22.

Item name	Percentage voting for Item	Amount allocated
Air Force	60.00%	$2,000,000.00
Conventional	70.00%	$1,000,000.00
Nuclear	50.00%	$1,000,000.00
Submarine Defense	60.00%	$500,000.00
Surface Defense	45.00%	$600,000.00
Defend East Coast	25.00%	$2,000,000.00
Defend West Coast	20.00%	$1,000,000.00
Defend Both Coasts	90.00%	$1,000,000.00
	83.00%	$2,000,000.00
Air Craft Carrier	85.00%	$500,000.00

Table 5: Sample Allocation For The Defense Budget

The fundamental operation in Genetic Algorithms is creating a new Chromosome by crossing-over. two Chromosomes. Assume there are ***K*** Genes in the Chromosome. A random value, ***S***, is chosen between 1 to $K-1$. The new Chromosome consists of Genes 1 to ***S*** from one of the two Chromosomes and Genes to $S+1$ to ***K*** from the other. This was inspired by the meiosis process in genetic reproduction.

Each cycle, the Genetic Algorithm crosses over the Chromosomes from two that have high rating.

To apply Genetic Algorithms to the budgeting process, Voters first propose their own budgets. The numbers on the Chromosome represent the parameters in the tax code and benefit structures, or simply the amounts budgeted for various governmental activities, say foreign aid to a specific country or grants for breast cancer research. Thus, the numbers from each User's ideal budget become the first Generation of Chromosomes.

Each of the subsequent rounds involves each User entering a rating between zero and one for each Chromosome in that Generation. The Fitness Function is simply the average of the ratings.

My voting scheme differs from the usual use of the Genetic Algorithms in how the fitness is computed. Each Voter would have a randomly-chosen set of Chromosomes or tax/budget

9 Chee-Kiong Soh and Jiaping Yang, "Optimal Layout of Bridge Trusses" " *Computer-Aided Civil and Infrastructure Engineering* 13 247-254,.

plans to rate. That is, there would be millions of Chromosomes, and each User would rate ten or twenty of them.

The cross-over and combination process would result in a new set of Chromosomes for a new Generation and a new round.

2.3. *Approve Budgeting*

This was inspired from Brams' work on Approval Voting[10], see Section 1. Computational Social Choice Voting Background. The voters choose the priority for how to distribute the revenue. (Usually this would be taxes, but for a few fortunate countries it would be from an exportable natural resource like oil.) All the possible Items on which they could spend the money would be available for vote. The Voters vote for as many as they Approve. The Items with the most votes are Approved. There are three types of Items: the simplest are ones which are either funded at a specific amount of money or not funded at all, for example, an air craft carrier or monument. The second type would be a grant-like funding program. Each User specifies the amount that he or she would be willing to fund. Finally, the third type would be benefit programs which would pay a specific amount of money to each person who qualifies—who met a specific set of conditions. (As discussed later in this section, this will use the ID3 Decision Tree process above.)

Rank	Number to divide pensions by	Years Of Service	Output class
Private	1	3	Q
Private	1	6	Q
NCO	3		R
Officer	2		S

("Years of service" doesn't matter for officers and NCOs.)

Table 6: After Pension Application Board Classifies The Pension Applications

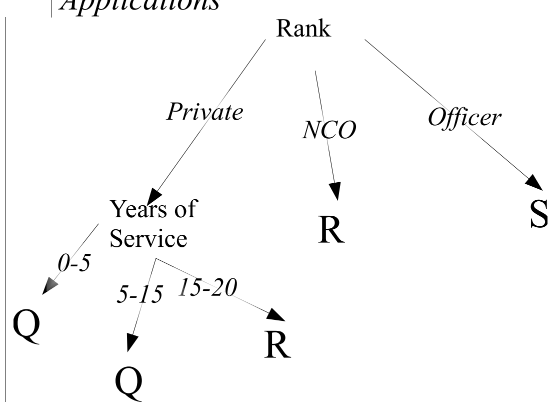


Figure 4: Decision Tree For Classifying Service People For Their Pensions

In addition, Items can be Merged and Split. For example, the group could vote to Split the Item for cancer research into several Items, one for each of the different types of cancer (such as lung cancer or breast cancer). They could create new combinations to get more support. A facetious but instructive example would be Merging funding the East Coast defense with funding the West Coast defense. Voters from the entire country would be more likely to vote for funding a complete coastal defense program than either program that only takes care of one region.

Table 5: Sample allocation for the Defense budget shows a sample set of votes from the Electorate. Assume $1.5 million were available. Then defending the coasts would be done at the one million dollar level and the aircraft carrier would be purchased.

The third type of Item is a benefit program; each Voter indicates the amount per person for each classification. The classification is given by an ID3 Decision Tree. An, obviously oversimplified, example for military service people pensions is given in Table 6: After Pension Application Board classifies the Pension Applications. and Figure 4: Decision Tree for Classifying Service People for Their Pensions. Assume that we have the number of pensioners in each class as given in Table 6. Then, assume that revenue is available in the steps given in Table 7: How Pensions would be allocated for a combination of ID3 and Approve Budgeting. Each output class with the most votes gets all the votes until they reach the amount allocated by that

10 Brams, Stephen, Fishburn, Peter C., *Approval Voting* Cambridge MA Birkhauser Boston 1983.

percentage voting. Then the class which received the next smallest percentage receives money. Notice how output class Q is listed three times in the Results tables. This means that the class gets money at three different points.

Incoming amount	Class(es) getting income	Classes for amounts to right	New amount for each class member	New balance for each class member
$3,000.00	Q	Q	$1,500.00	$1,500.00
		R	$0.00	$0.00
		S	$0.00	$0.00
$7,000.00	Q	Q	$3,500.00	$5,000.00
		R	$0.00	$0.00
		S	$0.00	$0.00
$12,000.00	R	Q	$0.00	$5,000.00
		R	$4,000.00	$4,000.00
		S	$0.00	$0.00
$6,000.00	R	Q	$0.00	$5,000.00
		R	$2,000.00	$6,000.00
		S	$0.00	$0.00
$14,000.00	Q and S	Q	$2,000.00	$7,000.00
		R	$0.00	$6,000.00
		S	$5,000.00	$5,000.00
$6,000.00	S	Q	$0.00	$7,000.00
		R	$0.00	$6,000.00
		S	$3,000.00	$8,000.00

Table 7: How Pensions Would Be Allocated For A Combination Of ID3 And Approve Budgeting

2.4. The Constitution Construction Kit

This is a massive on-line role-playing game built upon WorkFlow concepts and technology. More details are found in the Systems Requirement Specification[11].

In our system, there are multiple Constitutions proposed. Each Constitution is a set of Workflows, and legislative objects or judicial cases or administrative matters move through the Workflows. Every member of the group who could be Voters in the nation under the new Constitution can participate in simulating any of them. Then there is a Ratification process. It generalizes the above in choosing from many Constitutions so that Voters choose the best one; "best" being defined on the basis of having the support spread over all groups or geographic regions of the country as well as a simple count of all individuals.

Our system shows Constitutions and the legislative process as a flow diagram. Figure 5 shows a simplified diagram of the path of a bill in the United States from the House to the Senate to the President, including a possible veto override. The Constitution Construction Kit construction mode creates this, or adds to it. Voters add new steps and legislative bodies such a Constitutional Court or a Council of Experts; a smaller change would be to reconfigure a WorkFlow. A simple example would be changing the percentage needed to override a veto or ratify a treaty.

11 http://www.wiu.edu/users/mflll/CR.odt

But more than a simple percentage, a voting rule can ensure the representation of all Ethnic Groups. Our system supports a multi-way voting rule so that the legislative body can choose from among several alternative bills. Also, rather than a simple victory to the candidate with the highest percentage of votes, the system can be configured to use rules to give an advantage to bills with support over several Ethnic Groups or regions.

As a Constitution is designed, the moderator of the game, termed an RDF, puts the system into simulation mode. This is the true power of the Constitution Construction Kit (CCK). The Demos sign up to run for elections and participate in the political parties. They elect members, and they role-play the legislators. The legislators pass laws—the RDF can pose crises such as natural disasters or the sudden windfall of finding new natural resources, that is, the Oil Curse.

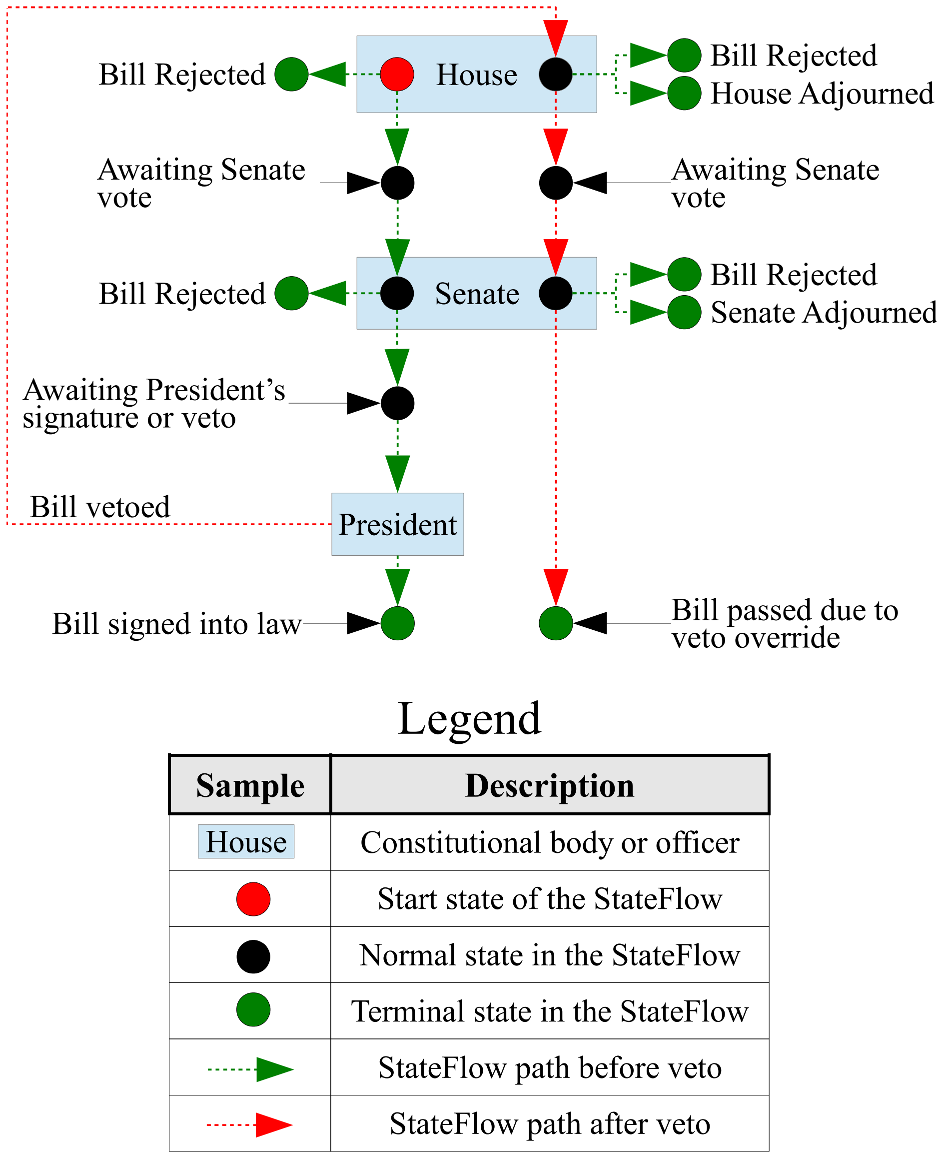


Figure 5: StateFlow For A Bill In The United States Constitution

I designed several techniques that can be designed in to ensure that bills would enjoy support over all or many Ethnic Groups or geographic groups in addition just winning a majority vote. The United States federal government has examples of that—bills have to pass the Senate where each state gets the same two votes regardless of population. And the electoral college has been designed to ensure that the president gets a more geographically-dispersed support than if the president was simply elected by the majority of voters.

The most important, albeit one-time, use of these voting rules is the way of selecting which Constitution is ultimately chosen—a more powerful alternative to a simple Ratification vote. Assume three candidates **X**, **Y**, and **Z** were proposed and simulated by large segments of the population, the Results of which were, of course, reported in the country's news media. Also, assume there are seven Provinces. Table 8: Nth Smallest Method of Selecting A Constitution gives the votes for each Constitution in each Province. Let's assume that the rule chosen was that the third smallest vote by Province determines which Constitution is ratified and used. Observe from Table 5 that the highest number is Province 4 for Constitution **Y**.

		Candidates		
		X	**Y**	**Z**
Province	**1**	35	25	25
	2	38	33	38
	3	* 36	42	30
	4	49	* 38	42
	5	80	47	* 35
	6	21	49	47
	7	50	50	50

Table 8: Nth Smallest Method Of Selecting A Constitution

The Nth smallest is one of the rules that the people configuring the bill approval process can specify as Workflows are proposed for consideration.

One can simply add that before a bill is Approved at a step, it must get the support of a certain percentage of the Ethnic Group or geographic regions involved. Such rules may include:

- That a super-majority is needed
- That at least percentage *x* in *y* percent of the geographic regions Approves the bill
- That at least percentage *x* in *y* percent of the Ethnic Groups Approves the bill
- That at least *x* percent of each of the two genders Approves the bill

Ethnic Group	**Candidate X**	**Y**	**Z**	**% of the population**
A	3	2	1	15%
B	3	6	1	15%
C	15	18	25	70%
Total	21	26	27	

Table 9: Sample Results In A Multi-Candidate, To Show Norm-Adjusted Voting

The more interesting requirements would be for Multi-Candidate elections, where one is choosing one of *n* candidates for a position or choosing among *n* alternatives for a bill that was proposed.

We can adjust the sum of the votes by a norm of each Ethnic Group (or geographic region) votes. For example, assume there are three bills and only one is to be selected, or three people running for office. I will refer to them as **X**, **Y**, and **Z**. Assume the total votes are as Error: Reference source not found, and the votes by each of the three Ethnic Groups **A**, **B** and **C**. The system also allows votes to be COAF or Score voting, such as approval where each Voter can vote for as many choices as they like, or range-voting, where each Voter can assign a number from zero to ten for each choice.

The system computes a norm for each candidate based upon the deviation from the average score in each Ethnic Group See Table 10: Averages and Percentages for Norm-Adjusted Voting to Protect Minorities. When the Users configure how voting will occur in the transition, they choose the weighting factor. The adjusted vote that is used to determine which candidate won is weighting $\mathbf{factor} \times \mathbf{norm} + \mathbf{vote}$. Table 11: The Norms for Norm-Adjusted Voting Example shows the calculation of the norm of the differences from the difference. The candidate with a high norm represents one whose support is mostly from a single Ethnic Group (see Table 12: Adjusted Score).

3. Deliverables

So what do we do?

3.1. Just Build Them

After the system is built and tested using GUI regression testing tools such as Selenium, I will have a large number of people use the system. The system will be coded to record in a log the actions each User takes, as well as a time stamp. This will, obviously, generate a large amount of data for statistical analysis. In the informed consent document, I will request permission to put the data anonymously on the internet for other political scientists to data mine. Of course, the web program will collect demographic information and at the end, ask for evaluations including free-form responses.

Ethnic Group	Candidate		
	X	Y	Z
A	50.00%	33.33%	16.67%
B	30.00%	60.00%	10.00%
C	25.86%	31.03%	43.10%
Average	35.29%	41.46%	23.26%

Table 10: Averages And Percentages For Norm-Adjusted Voting To Protect Minorities

The sample might simply be those on the internet who care to try the project. We will ask people on the internet to just try it. Or we might bring in a specific group or groups such as senior citizens from a senior citizens' center. The convenience sample of a large number of college students comes to mind[12].

In addition to the specific experiments outlined below, various communication facilities will be provided. For example, the participating group could:

1) be physically together in the same room and encouraged to talk and chat normally
2) have video and voice communication
3) have voice communication
4) use text communication.[13]

I anticipate collecting relevant library materials, including recidivism statistics and articles about tax policy. This will allow library faculty to develop studies on how to present this information in participatory democracy situations.

Ethnic Group	Candidate		
	X	Y	Z
A	14.71%	-8.12%	-6.59%
B	-5.29%	18.54%	-13.26%
C	-9.43%	-10.42%	19.85%
Squares	216.5	66	43.4

Table 11: The Norms For Norm-Adjusted Voting Example

3.1.1. ID3 System

The development phase is to pay students to prototype an ID3 Decision Tree process. The first one will be for a penal code. A good sample problem would be setting the circumstances under which it is prohibited to possess a gun. Another example will be setting the type of felony sentences for various homicides and related crimes. This includes the capital

12 Every four years, my home institution, Western Illinois University, simulates the Presidential election process. 5000 students hold the "mock presidential election." See http://www.wiu.edu/news/newsrelease.php?release_id=9262. A search on Google shows many "lesson plans" and resources for high school students to form a mock legislature. Certainly, these techniques and software can be used that way as well.

13 Dr. Alphonse Chapanis compared these types of communication where two students had to collaborate to solve various problems. In -person communication and voice took approximately half an hour while exchanging text took 67 to 69 minutes. *Scientific American* 75, Volume 232(3) 36-42.

X	Y	Z
14.3	15.6	14.7

Table 12: Adjusted Score

punishment category. What happens when a person plans to kill over ten people, but only manages to injure one person? Under what combination of Title values affects should the penal code distinguish between whether or not this was a murder for hire, whether the purpose of the murder is financial gain such as killing a person for insurance funds, for inheritance or due to animosity towards a racial group, Ethnic Group, religious group or sexual orientation?

New York State penal code classifies crimes as Class C, B, and A misdemeanors or Class E, D, C, B, and A felonies. A separate class X means that the crime is subject to capital punishment. These would be the Results on which crime situations will be classified. I will refer to this as the "crime classification." The Decision Tree will assign a crime classification to any set of circumstances, including one for no penalty.

3.1.1.1. Experimental Comparison

When this is run with real people, there will be example data. As is done in experimental work in machine learning, some of the examples will be the training set and the remainder will be a test set. As the experimental participants vote in the system, they will be encouraged to use the training set to see how each element would be classified. That is, at any Node in the Decision Tree in progress, the software shows the participants how the example would be classified.

The experimental control group will simply assign a Result to each example in the training set. Then a conventional Decision Tree learning algorithm[14] will generate a Decision Tree for each, using as examples each rating.

The participants will then be asked to classify each element of the test set and to state their satisfaction with other classifications of that training set. Thus, if the test set included a person convicted of a misdemeanor, having a BB-gun in a Bar, they will be asked to give their best classification for that crime. Assume they said it should be a misdemeanor of type B. They would be asked what they would think if it was in fact classified as a misdemeanor of type C or type A. Both the voting-generated and conventionally-generated Decision Tree would be run on the test set and the examples compared.

Later, we could try categorizing people for tax or benefit purposes. One group would use the ID3 categorization system to assign each person a letter A, B, C, D, E; another group would vote on the amount of benefit for A, B, C, D, and E.

3.1.2. Genetic Algorithm System

The first implementation of the Genetic Algorithm system, Section 2.2, is budgeting. Some of the the parameters would represent the marginal tax rate for each of the quintiles—bottom twenty percent of the income earners, next twenty percent, those who earn between the bottom 40% and the top 40%, as well as some top tax rates such as the top one percent and those earning more than one million dollars per year. The Internal Revenue Service Statistics of Income division published tables for the amounts earned, as well as deductions[15].

From these, I will have my group develop a model, with the emphasis on its feeling realistic to the participants rather than its being accurate. This will allow the Users to set the tax

14 Quinlan, J. R. *C4.5 Programs for Machine Learning*, Morgan Kaufman, 1992.
15 http://www.irs.gov/taxstats/

rates on each quintile, the corporate income tax rate, the tax rate on capital gains as well as the amount of deductions or exclusions for R&D, taxes, etc.[16]

On the spending side, the Chromosome will have parameters for the total spending on non-entitlement programs such as highways and National Institute of Health grants. Benefit programs will also be controlled by parameters. In the Social Security System[18], when a person retires, each past year's income is multiplied by a factor to adjust for inflation. (That is, for a retiree now, the $5000.00 earned in 1950 counts more than the same amount earned in 1975.) The top 35 years are summed and a monotonic function is applied to this to determine the monthly benefit. (For those retiring at 65, 90% of the first 767 dollars in indexed earnings, and 32% of the amount between $767 and $4624, is the benefit.) In Genetic Algorithm calculations, the parameters would be:

1. percentage of the first 767 dollars in indexed earnings for people retiring at 65
2. percentage of the amount between $767 and $4624 in indexed earning retiring at 65
3. percentage of the first 767 dollars in indexed earnings for people retiring at 62
4. percentage of the amount between $767 and $4624 in indexed earnings for people retiring at 62

The Users would be given boxes to fill in the numbers they would want in their ideal budget. As they do so, a Javascript implemented in the browser will display the deficit. When the Users indicate they are satisfied, this is entered as a Chromosome in first Generation. Each round, each User will be presented with a sampling from that Generation.

The data from each Chromosome will be displayed, along with the deficit. The Users will enter their ratings. The ratings for each Chromosome will be averaged and this entered into the crossover phase of the Genetic Algorithm to generate the next Generation.

The most basic question is do we have convergence. If 1000 people enter their initial budgets, each member rates five budgets randomly chosen from the individually-proposed budgets. The highest-rated from that set are crossed over to generate new budgets. Then, the participants rate five of the budgets from this set. The highest rated from that set are crossed over—repeat. Do we end up with any sort of convergence at what appears to be the most satisfactory possible budget to the participants?

---

16 Nathan Newman nicely developed the National Budget Simulation, http://www.nathanneman.org/nbs/longbudget06.html, for people to do what-if games. It has many categories such as natural resources development, agriculture and social security old age and survivor insurance (OASI). Users entr "hold even" or raise or lower by 10%, 20%, 30%…100%. (Lowering by 100% eliminates the program). On the tax side, the users spec wh to change the 2001 and 2003 tax cuts and if so, by what percentage to increase or decrease them. However, they do not set the actual amount of money that a social security annuitant receives each month or the percentage in taxes that a person earning in the top ten percent of the nation should pay.
Recently, the Committee for a Responsible Federal Budget put out a similar system. (http://crfb.org/stabilizethedebt)

18 ww.ssa.gov/pubs/10070.html, SSA Publication No 05-10070. Also, the Social Security administration has data on recipients, http://www.socialsecurity.gov/policy/docs/statcomps/supplement/2011/index.html

3.2. Approve Budgeting

Implementing this is quite simple. We woulc States federal budget or the budget for state in the Users would enter an amount that they Approve spe percentage of the average of the first $767.00 that a sc the percentage of the amount between $767 and $4624 Social Security works under Genetic Algorithms.)

1 $\mathbf{A} \rightarrow \text{True}(0.13, -0.13)$
2 $(\neg \mathbf{A}) \rightarrow \text{False}(0.24, 0)$
3 $(\mathbf{A} \wedge \mathbf{B}) \rightarrow \text{True}(0.17, 0)$
4 $(\mathbf{A} \wedge (\neg \mathbf{B})) \rightarrow \text{False}(0.5, 0.25)$
5 $\mathbf{B} \rightarrow \text{True}(0.15, -0.05)$
6 $(\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C}) \rightarrow \text{True}(0.16, -0.02)$
7 $(\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C} \wedge \mathbf{D}) \rightarrow \text{True}(0.18, -0.04)$
8 $(\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C} \wedge (\neg \mathbf{D})) \rightarrow \text{True}(0.14, 0.03)$

List 1: Voter P

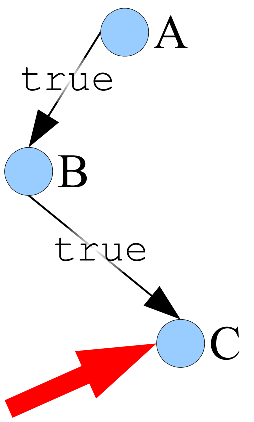


Figure 7: Node Voting Example HA

Users would enter the pa the tax system. That is, they wo percentage of tax paid by each percentile of income, the percentage tax on capital gains, the percentage paid on dividends, the corporate income tax rate, etc. The computer would take the median for each of these numbers. From these the revenue would be computed. Then, the computer program would display the resulting budget.

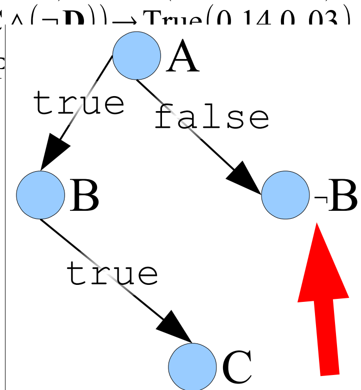


Figure 6: Node Voting Example HB

3.2.1.1. Possible Later Additions

We would compute a new budget in each simulated year. That is, the Voters could change their votes based upon the previous year's deficit and their views of the revenue collected and amounts paid out by the benefits programs. The parameters would change slightly, perhaps under the control of a "Dungeon Master," or as I term it to avoid trademark restrictions, the Reality-Detail-Filler-in.

However, in the Constitution Construction Kit, I proposed two interesting additions involving hierarchies of Approve Budgeting. One of these is for grant programs such as the National Science Foundation or the National Institute of Health. The main group would select a total level of funding for the granting program. Then a subgroup would allocate the funding to specific grant proposals. The problem is getting lists of possible things to fund. Of course, grant proposals are confidential, especially those that were not funded. However, I would hope the program directors would solicit permission from the applicants to use their proposals, possibly redacted. For National Aeronautics and Space Administration (NASA), I assume they could provide a list of missions on the drawing board.

4. Simulations—Experimental Algorithms

4.1. ID3

Algorithms to play games are simulated. For example, when one is developing algorithms and techniques to play chess, one plays both opponents. In the same spirit, we will do the same thing with the ID3 techniques. However, we will simulate with a large number of groups. As described below, zero, one or two groups will be strategic. The others will vote truthfully. By strategic I mean that they will set up a game tree in an attempt to determine the best move, considering the moves of the other populations.

We will simulate the creation of a tree for four Titles: **A**, **B**, **C**, and **D,** with two Attributes (either true or false). For simplicity, we will also assume only two Results—also, true and false. Input data will be the set of preferences for each of P simulated populations. A preference is a conjunction of literals in the Titles, a Result value (`true` or `false`), and real number satisfaction if that is classified correctly or incorrectly. In the gun-control example (see section 2.1), a preference might correspond to (**location**=*Bar*, **gun-type**=*Pistol*, **person-occupation**=*Security-guard*, **person-conviction**=*Misdemeanor*), Permitted, (0.37,-0.25). That means if a penal code says that if a person who previously committed a misdemeanor was now working as a security guard, and was permitted to have a pistol in a bar, that population would view that situation positively at 0.37 and if they were not permitted under those circumstances, the population would have a dissatisfaction represented by the negative number, -0.25. List 1: Voter Preferences shows a sample set of preferences for a particular population.

Title	Preference from List 1: Voter Preferences	Difference	Total
A	1	0.26	1.43
	2	0.24	
	3	0.17	
	4	0.25	
	6	0.18	
	7	0.22	
	8	0.11	
B	3	0.17	1.13
	4	0.25	
	5	0.2	
	6	0.18	
	7	0.22	
	8	0.11	
C	6	0.18	0.51
	7	0.22	
	8	0.11	
D	7	0.22	0.33
	8	0.11	

Table 13: Deciding Which Title To Vote For To Expand The Root Node

Assume that the Voters are voting on a Node **C** in Figure 7, indicated by a big red arrow. Of course, the options would be to expand on **D** or just choose a Result. Since this population has preferences recorded for both $\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C} \wedge \mathbf{D}$ and $\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C} \wedge (\neg \mathbf{D})$, they will vote to expand (at least if they are not voting strategically). Let us look at the same population confronted by the Node corresponding to the conjunction $\mathbf{A} \wedge (\neg \mathbf{B})$ in Figure 6: Node Voting Example HB. Of course the choices would be to expand on Title **C**, Title **D**, or to just choose a Result. Since that population has no preferences for $\mathbf{A} \wedge (\neg \mathbf{B}) \wedge \mathbf{C}$ versus $\mathbf{A} \wedge (\neg \mathbf{B}) \wedge (\neg \mathbf{C})$ or $\mathbf{A} \wedge (\neg \mathbf{B}) \wedge \mathbf{D}$

versus $\mathbf{A} \wedge (\neg \mathbf{B}) \wedge (\neg \mathbf{D})$, they would vote to just choose a Result.

A Non-Strategic Population would vote to expand on the Title for which there is the greatest difference in satisfaction between True and False for that Title. This would be at the Root Node, for the example population. So assume the population has the preferences given in List 1 and is non-strategic. Table 13: Deciding Which Title to Vote for to Expand the Root Node gives the differences that the population sees at the Root Node.

Thus, at the root, this population would vote to expand on Title **A**, whose total difference is 1.43.

If after voting, the Root Node were in fact expanded in Title **A**, then this population would vote to expand the Node for **A** true by **B**. (Its difference would be 0.38 from $\mathbf{A} \wedge \mathbf{B}$ and $\mathbf{A} \wedge (\neg \mathbf{B})$.) Table 14: Deciding how to expand the Node When A is true shows the differences that it sees at that Node; as **B** has a difference of 0.93, it would vote to expand by **B.**

Title	Preference from List 1: Voter Preferences	Difference	Total
B	3	0.17	0.93
	4	0.25	
	6	0.18	
	7	0.22	
	8	0.11	
C	6	0.18	0.51
	7	0.22	
	8	0.11	
D	7	0.22	0.33
	8	0.11	

Table 14: Deciding How To Expand The Node When ***A*** *Is* **`true`**

Also, of course, each population would have an associated size, such as 55,323 people. Its vote would be weighted by the population size. Then, one of the Multi-Candidate social choice techniques would be used to determine which Title was used to expand. (See Figure 6: Node Voting Example HB.)

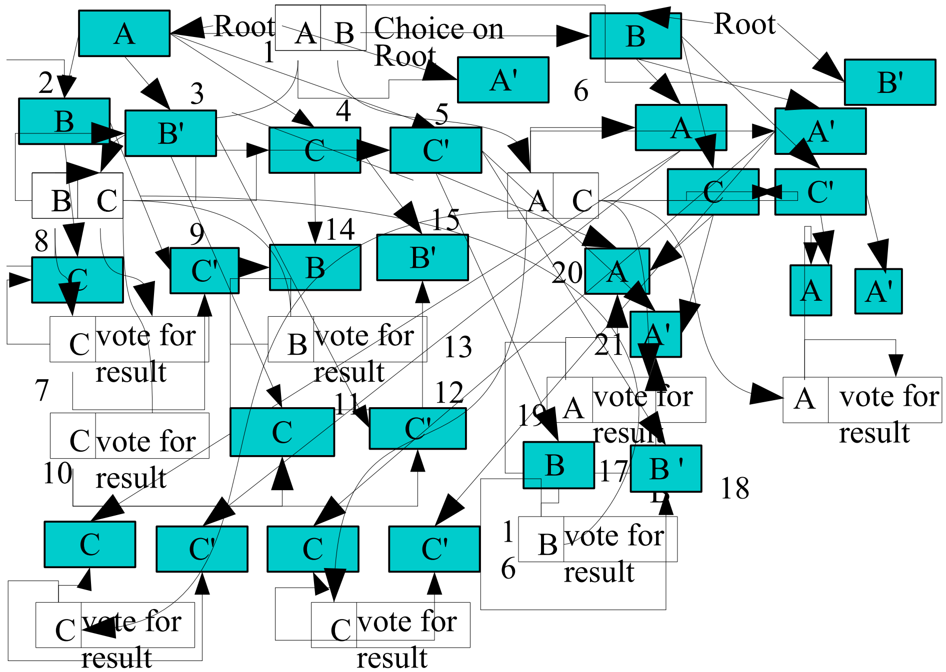
Keep in mind that at each Node the alpha-beta tree contains a pointer to the sub-tree in the ID3 Decision Tree being formed. I describe below four simulation types, followed by experimentation with desires for the tree.

4.1.1.1. One Population Competing, The Max-Min Case

In Figure 8, assume that the manipulator does not have enough weight to force the Root Node to be expanded on **C.** (I show only one manipulator as the diagram is already complicated enough.) Thus, there is a forest of two trees, one rooted at **A**, the other rooted at **B.** For the left hand tree, rooted at **A**, observe that is pointed to by the alpha-beta tree entry (Choice of root) side for **A**. (The term alpha-beta as used here is of course a misnomer. I want the same term for the Node in both this type of simulation and the one with two Strategic Populations.)

The other part of this points to the forest rooted with the Title **B**. The simulation program observes when a voting choice has no effect on the output. Thus, if the manipulating Population votes to expand the root with Title **C**, the non-manipulating populations have sufficient votes to prevail to expand with either **A** or **B**. Thus, the diagram does not have **C** in the Root Node of the alpha-beta tree.

Once the Root Node is expanded in **A**, the Voters have to decide whether to expand **A** in either **B** or **C**, or vote for Result. (Assume that the manipulator is less satisfied with the vote on



The Legend:

The Nodes of the Decision Tree

The alpha-beta tree

Lines connecting a Node in the Decision Tree to its descendent

Lines connecting the Error: Reference source not found in the alpha-beta tree to the corresponding Node in the Decision Tree

Lines connecting a Node of the alpha-beta tree

Figure 8: The Very Complicated Way An Alpha-beta Tree Relates To A Decision Tree In Experimental Algorithmics For The ID3 Algorithm

just **A** than any possible expansion of the tree. Thus, there is no entry in that alpha-beta tree Node for "vote for Result.") Alpha-beta Node 1 connects to the two Nodes (2 and 3) for the expansion for **B.** Also, there is a connection from **A** to the expansion for **C** and **¬C** (4 and 5) in the sub-tree for **A**. When the original Root Node is expanded in **A**, the system will create a Node, 6, for $\neg\mathbf{A}$. Due to space constraints, I do not show its expansion. When $\mathbf{A}\wedge\mathbf{B}$, Node 2, is expanded, alpha-beta tree Node 7 shows the choice of creating Nodes for $\mathbf{A}\wedge\mathbf{B}\wedge\mathbf{C}$ and $\mathbf{A}\wedge\mathbf{B}\wedge(\neg\mathbf{C})$ (8 and 9). Similarly, $\mathbf{A}\wedge(\neg\mathbf{B})$ (Node 10 of the alpha-beta tree) is expanded into $\mathbf{A}\wedge(\neg\mathbf{B})\wedge\mathbf{C}$ and $\mathbf{A}\wedge(\neg\mathbf{B})\wedge(\neg\mathbf{C})$ (Nodes 11 and 12).

Node 4 ($\mathbf{A}\wedge\mathbf{C}$) is expanded by alpha-beta Node 13 into $\mathbf{A}\wedge\mathbf{C}\wedge\mathbf{B}$ and $\mathbf{A}\wedge\mathbf{C}\wedge(\neg\mathbf{B})$ (Nodes 14 and 15). And Node 5 ($\mathbf{A}\wedge(\neg\mathbf{C})$) is expanded by alpha-beta Node 16 into $\mathbf{A}\wedge(\neg\mathbf{C})\wedge\mathbf{B}$ and $\mathbf{A}\wedge(\neg\mathbf{C})\wedge(\neg\mathbf{B})$ (Nodes 17 and 18).

I won't describe all the Nodes for the part of the Decision Tree corresponding to the possibility that the original vote expands the Root Node in **B**. However, one of them is the expansion of Node for $\mathbf{B}\wedge\mathbf{C}$ into $\mathbf{B}\wedge\mathbf{C}\wedge\mathbf{A}$ and $\mathbf{B}\wedge\mathbf{C}\wedge(\neg\mathbf{A})$ (alpha-beta Node 19 and Nodes 20 and 21 in the Decision Tree).

4.1.1.2. Two-Populations Competing, The Alpha-Beta Pruning Case

Here, two populations will be selected to be Strategic Populations. Both of these populations perform a look-ahead which will be four levels deep. For the Root Node, the four options will be:

- vote to expand by Title **A**
- vote to expand by Title **B**
- vote to expand by Title **C**
- vote to expand by Title **D**

If this Strategic Population has sufficient size to outweigh the Non-Strategic Populations, there would be four resulting Nodes in the alpha beta tree:

- the ID3 Decision Tree was expanded at the Node by Title **A**
- the ID3 Decision Tree was expanded at the Node by Title **B**
- the ID3 Decision Tree was expanded at the Node by Title **C**
- the ID3 Decision Tree was expanded at the Node by Title **D**

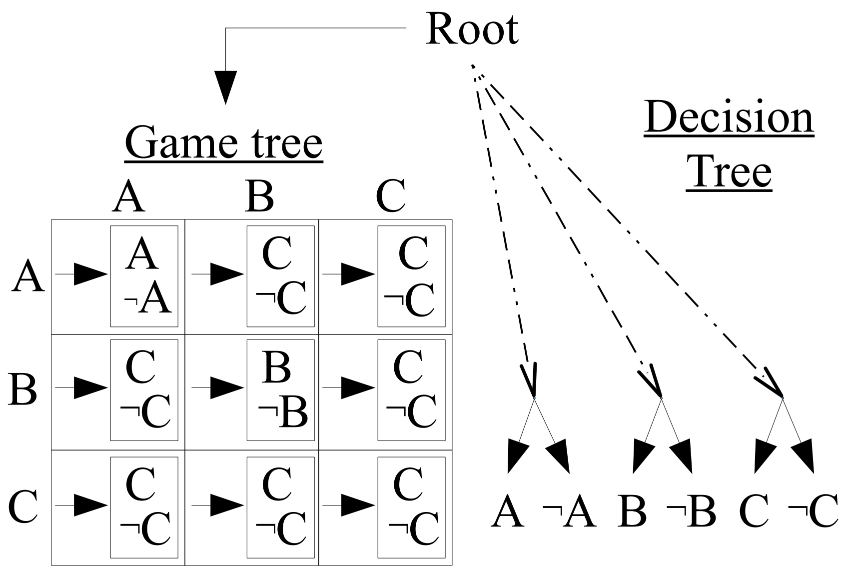


Figure 9: The Beginning Of The Two Trees For Simulating Two Manipulating Populations

For the first level Nodes, there will be three remaining Nodes. Thus, at an alpha-beta node corresponding to the Decision Trees having been expanded to **A**, there will be two branches in the Decision Tree (**A** and ¬**A**). At each of these, there will potentially be three branches:

- vote to expand by Title **B**
- vote to expand by Title **C**
- vote to expand by Title **D**

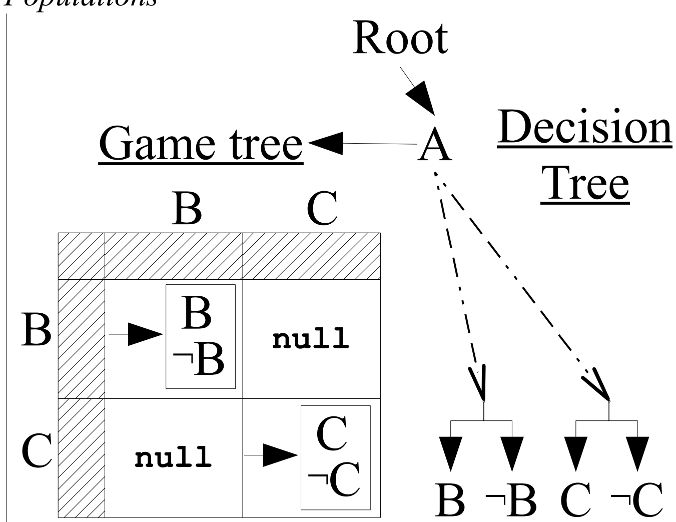


Figure 10: Two Manipulator Alpha-Beta/Decision Tree Examples For Node A

(If this population was totally indifferent between all conjunctions with **A** and any of the Titles, it would vote to simply make the decision at this point, as it would not care what happened below that Node. Similarly, the population would be totally indifferent at ¬**A** and any of the possible decisions for **B**, **C**, and **D**; it would vote to just make the decision at that point.)

Figure 10: Two Manipulator Alpha-Beta/Decision Tree Examples for Node A shows a situation for Node A. Assume that the Non-Strategic Populations are such that only if both manipulators vote to expand this by **B** will it expand by that Title, and only if both manipulators vote to expand this Node by **C** will it expand by that Title. Otherwise, the votes for just voting the Result for the Node corresponding to any case with **A** will prevail. In other words, the

Decision Tree would have no Nodes below **A**.

If this were a simulation of two manipulators competing against each other, each has simultaneous moves at each Node, casting a vote on how to expand this. Thus at the Root Node, there is a matrix with rows for **A**, **B**, and **C** corresponding to the possible votes for the first manipulator at the Root Node, and columns for **A**, **B**, and **C** for the second manipulator's votes.

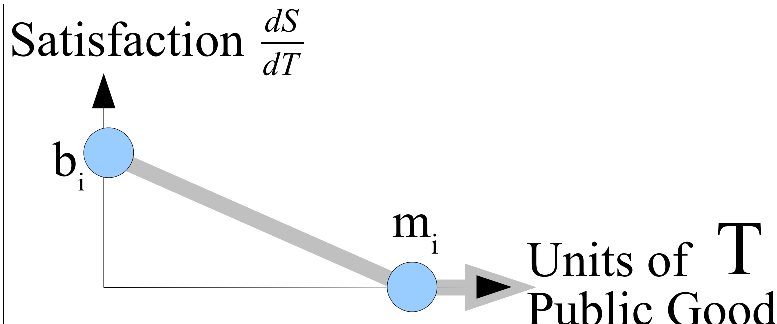


Figure 11: Satisfaction From The Provision Of A Public Good

Assume the non-manipulating populations favor expanding in **C** so much that only if the two manipulators vote for **A**, will the root be expanded in **A**, and only if the two manipulators vote for **B**, will the root be expanded in **B**. Then Figure 9 shows the alpha-beta Node for the root.

4.1.1.3. The Perfect Information Case

This is the benchmark. The computer program would compute the satisfaction for each possible Decision Tree. (There will be a relatively small number if there are only four levels—so exhaustive enumeration is possible). The best or ideal Decision Tree from a social choice perspective is the one that maximizes the total satisfaction for all the populations weighted by their population size. The experimental algorithmic goal is to find out the social cost of strategic behavior in this voting context.

4.1.1.4. Greedy

Each population simply votes to expand a tree by the Title that gives the greatest difference in satisfactions. It votes to just make a decision only when it sees no difference between all sets with any remaining Titles. See Section 4.1.1.5.. (We will also look at simplicity desires.)

4.1.1.5. Satisfaction

The first set of simulations will have each population have no preference for simplicity. The only time they would vote not to vote on a Result is when they have the same preference for all of the possible sub-trees.

However, some populations may prefer trees with fewer Nodes; that is, as a tree gets larger, they may get fatigued and be more likely to vote to just classify. This will be modeled as a monotonically increasing function of the number of Nodes. Assume the value of that function is f for the number of Nodes in the tree. When that population is asked to vote, it computes the maximum gain in satisfaction if all the expansions on the sub-tree voted the way that population wanted it to. This will be compared to the, probably negative, satisfaction when all the nodes on that sub-tree were voted opposite to the desire of the population. If that difference is less than f, it will vote to classify the Node and not to expand it further.

The other issue is that a population may prefer simpler shallow trees. That is, there is a preference to classify $\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C}$ rather than $\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C} \wedge \mathbf{D}$ and $\mathbf{A} \wedge \mathbf{B} \wedge \mathbf{C} \wedge (\neg \mathbf{D})$. This is modeled as a monotonically increasing function of the depth of a sub-tree. Assume the value of that function is g for the depth of a particular Node. When that population is asked to vote, it computes the maximum gain in satisfaction if all the expansions on the sub-tree were voted to be classified the way that the population wanted it to. If that is less than g, then it will vote to classify this Node and not to expand it further.

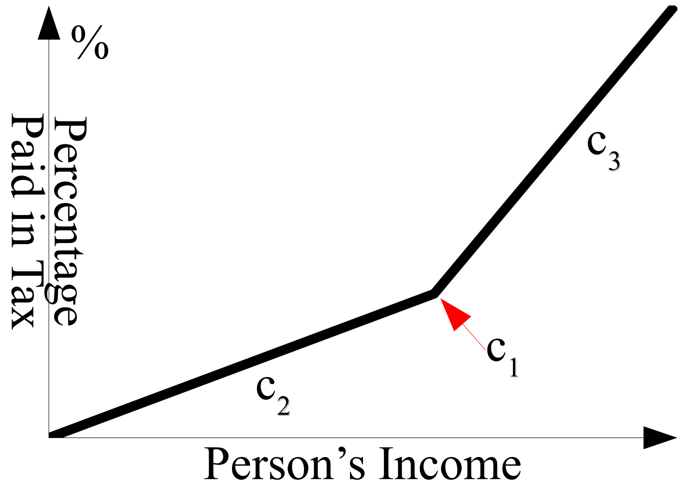


Figure 12: What The Genes On The Chromosome Mean For A Simple Progressive Tax

```
Solicit initial Chromosomes from each population:
for each Generation, i
  for each population
    for each Chromosome in that Generation
      Assign Rating (See Below)
  for each Chromosome
    Sum the Ratings over all populations
  Let Set X be the W highest-rated Chromosomes from Generation i
  for 1 to e×G // the number of cross-overs
    Choose one weighted randomly-chosen from X as Wa
    Choose another weighted randomly-chosen from X as Wb
      Choose a cross-over location, k, randomly from 1 to K−1
     Create a new Chromosome consisting of Genes 1 to k from Wa
      plus Genes k+1 to K from Wb
    Put that Chromosome in the forming Generation i+1
```

Listing 1: Source For Chromosome Evaluation

4.2. Genetic Algorithms

First simulate finding a tax plan to fund one public good. This tax plan will apply to all the P, populations. Each population has as input:

1. an Income distribution, g(%), a function of the percentage of the income
2. a preference for the public good based upon the following curve. Each population, i, has its own value for m_i and b_i. This curve comes from Heyman[19]. The curve in Figure 11: Satisfaction from the provision of a public good gives the marginal benefit or satisfaction from adding one more unit of the public good. Each population wants to maximize its total satisfaction. Let T be the total revenue collected, summing all the populations. Thus, the satisfaction for this population can be given as $\int_0^{min(b_i m_i, T)} m_i T - b_i \; dT - taxItPaid$.

The first example will be a simple progressive tax, Figure 12: What the Genes on the Chromosome mean for a simple progressive tax. The tax on each person's income is determined by three values on the Chromosome: c_1 – placement of the kink in the curve, c_2 – the tax rate on all income below c_1, c_3 – the tax rate on each dollar of any income above c_1. We compute the tax on the population for a given Chromosome as $\int_0^{c_1} c_2 g(\%) d\% + \int_{c_1}^{100} c_3 g(\%) d\%$.

Name	Definition
G	Size of each Generation
W	Take the W highest-rated elements of Generation *i* before crossing over
e	Cross-over percentage. e% of the new Generation is given by cross-overs. The remainder is taken from Generation *I*.
K	Length of Chromosome

Table 15: Parameters For Listing 1

First, we will simulate where each population has a constant income, I_i. For these simulations, the tax on that population is multiplied by the size of the population.

$$\begin{cases} c_2 I_i & I_i \le c_1 \\ c_2 c_1 + c_3 (I_i - c_1) & I_i > c_i \end{cases}$$

The text in Listing 1: Source for Chromosome evaluation gives the pseudo-code of the Genetic Algorithm simulation. It uses the parameters shown in Table 15. Observe that in this set of simulations, the Chromosome will have three Genes, the values for c_1, c_2, and c_3.

19 Heyman, *Public Finance*, Fifth Edition, Dreyden Press

```
DoGeneration(CS:ChromosomeSet, W:SetOfWeights)
RS:ChromosomeSet
RS=null
for each Chromosome in CS
  Sum that column in W.
  Let Set X be the W highest-rated Chromosomes
  for 1 to e×G  // the number of cross-overs
    Choose one weighted randomly-chosen from X as Wa
    Choose another weighted randomly chosen from X as Wb
    Choose a cross-over location S randomly from 1 to  K−1
     Create the new Chromosome consisting of Genes 1 to S in Wa
      followed by Genes  S+1  from Wb to K
    add that Chromosome to RS
  end for
return RS
```

Listing 2: Doing One Generation Of A Genetic Algorithm

Experiments will be done with zero, one and two Strategic Populations—the remaining populations will be Non-Strategic Population. Two techniques for these populations:

- Each population rates the Chromosome which gives it the highest satisfaction at one and the remaining Chromosomes in that Generation will be rated at zero.
- Each population rates the Chromosome which gives it the highest satisfaction at one. All other Chromosomes are rated proportional to their satisfaction. Assume there are four Chromosomes and population one has satisfaction for them at 25, 10, 7, and 4. That means they would be rated at 1, 0.4, 0.28, and 0.16, respectively.

```
StrategicFloat(CS:ChromosomeSet, I, J) returns float
NS:ChromosomeSet
W1:SetOfWeights
max1:float
temp1:float
In W1, assign Weights for the rows for Populations Two through P
  as discussed at end of section of 4.2
max1 = very negative number
For N times
  generate a random set of weights for Population One in the
     first row of W1
  NS = DoGeneration(CS, W1)
  if (J<DT) then
    temp1 = StrategicFloat(NS, I, J+1 )
    if (I=0)∧(temp1>max1) then
      max1 = temp1
      maxweight = NS
    end if
  else
    Let temp1 = the average of the satisfaction of NS
  if (temp1>max1) then
    max1 = temp1
  maxweight = NS
end for
if (J=0)∧(I != DT) then
  maxweight is the weight used at this level and we then call
  strategic(max1, I+1 , 0) to do the next Generation of the
  Genetic Algorithm.
end if
return max1
```

Listing 3:One Strategic Population

When experimenting with one Strategic Population, it will calculate the expected value for a given set of weights from this population. That is a greedy strategy. The simulation will calculate for the population which weights will optimize the expected value for the next Generation. Note, in this experiment, the Strategic Population will not look ahead more than one Generation.

```
Strategic(CS ChromosomeSet, I, J) returns SatisPair
NS:ChromosomeSet
W1:SetOfWeights
max1:SetofWeights
GameTheoryData:matrix with P rows and G columns/* contains two
  numbers per element */
in W1, assign weights for  populations three through P as
  discussed above
for Q=1 to N // Population One
  for R=1 to N // Population Two
    generate a random set of weights for population one and put
      it in the first row of W1
    generate a random set of weights for population two and put
      it in the second row of W1
    NS[Q,R] = DoGeneration(CS, W1)
    if (J<D_T) then
      temp1 = StrategicFloat(NS, I, J+1 )
      fill in GameTheoryData[Q,R] with the two "payoff" numbers
        from temp1
    else
      fill in GameTheoryData[Q,R] with average value for
        satisfaction for Population One and average value for
        satisfaction for Population Two of all Chromosomes in NS
    end if
  end // for R
end // for Q
Get the Nash Equilibrium for GameTheoryData using a tool such as
  GAMUT.
if (J=0) then
   if GameTheoryData has one Pure Solution let [q,r] be the
      values of Q and R corresponding to the pure solution
    return Strategic( NS[q,r], I+1, 0)
  end if
  if GameTheoryData has multiple pure solutions
    choose one of the pure solutions randomly, let [q,r] be the
      values of Q and R corresponding to the pure solution
    return Strategic( NS[q, r], I+1, 0)
```

Let w_i be the rating from this population for Chromosome i. Let $S(c_1, c_2, c_3)$ be the satisfaction from a tax plan from a Chromosome whose three Genes are c_1, c_2, and c_3. $c_{1,i,j}$, $c_{2,i,j}$, and $c_{3,i,j}$ are the values for the Genes on the Chromosome$_j$ from the Generation$_i$.

Let $W_{(T,j)}$ be the total weights from the all the Non-Strategic Populations for Chromosome j. Let W_{TT} be the sum of $W_{(T,j)}$ for all populations, both this one and the

Non-Strategic Populations. $c_{1,i,j}$, $c_{2,i,j}$, and $c_{3,i,j}$ are the values for the Genes on the Chromosome$_j$ from the Generation$_i$.[20]

For a particular weighting scheme, the number: $\sum_{j=1}^{G}(1-e)\frac{W_{T,j}}{W_{TT}}S(c_{1,i,j},c_{2,i,j},c_{3,i,j})+\sum_{j=1}^{G}\sum_{k=1}^{G}e\frac{W_{T,j}W_{T,k}}{W_{TT}^{2}}S(c_{1,i,j},c_{2,i,j},c_{3,i,j})$ is the expected value for the Chromosomes in the next Generation. Assume the system allows range-voting; that is, each population can enter a real number between zero and one for the weight for each range. Then, finding the optimal set of weights is calculated by linear programming.

For two Strategic Populations, I hope to generate a game theory equilibrium, that is, the min-max weighted scheme. Each ply of the game tree would have many elements. There are many Chromosomes in each Generation. Thus, if the only possible strategy is to rate a single Chromosome at one, with zero for all other Chromosomes in the Generation, there would be G rows and columns in the game-theory payoff matrix. If there are more complicated voting systems or decisions or social choice mechanisms, the payoff matrix would be even larger or possibly infinite.

4.2.1. Strategic Game Tree Approaches

We assume each Strategic Population looks ahead D_1 steps and that the Genetic Algorithm proceeds for D_T Generations. Both D_1 and D_T are parameters to the simulation.

We begin by taking Listing 1 and making a subroutine to be invoked for generating each Node of a game tree. A `ChromosomeSet` represents a Generation of Chromosomes. The subroutine returns the Result of one round of the Genetic Algorithm if the weights were as specified by W. A `WeightSet` represents the a possible assignment of weights by each member of the population. It will be a two-dimensional matrix where the rows represent the populations and the columns represent the ratings for each Chromosome. Thus, it is P rows by G columns (P is the number of populations). See Listing 2: Doing One Generation of a Genetic Algorithm.

Obviously, in most cases, it will not be possible at each level of the game tree to explore every possible alternative. For example, if the populations can assign a zero or one weight to each Generation there would be 2^G possible branches. And, of course, if the system allows the weights to be real numbers, proportional to the satisfaction given by each Chromosome, then there would be an infinite number of branches. The parameter N is the number of branches examined at each play.

Each Strategic Population will choose randomly possible strategies to apply. I will explore sampling from a uniform distribution and searching in the vicinity of the solution that leads to the best expected value for satisfaction in the next Generation. I represents the Generation being processed. J represents the position in the look-ahead.

Hence, if the program passes $J=0$ **to** `Strategic`, this is a signal that this is the actual application of preferences in the Genetic Algorithm. The values of J range from 1 to D_1.

20 This may require normalization so the denominator W_{TT} is constant.

The routine returns the best satisfaction it finds for the first population. This will be the average value for the Chromosomes in the Node of the game tree whose average is highest. See Listing 3:One Strategic Population.

The second simulation will involve two Strategic Populations. It is similar to the pseudo-code for one Strategic Population, above. At each Generation, we assume that the populations will choose a Nash equilibrium based upon the look-ahead Result. The first two rows of the `SetOfWeights` represent the weights for the two populations. The subroutine will set up the payoff matrix *`GameTheoryData`*. This, of course, is a two-dimensional $N \times N$ array, each element containing the satisfaction for each of the two Strategic Populations. Then it solves this using existing software[21] to find the Nash equilibrium. It returns, as a Result, the expected value for the satisfaction for each of two values. The type set will be `SatisPair`. See Listing 4:Two Strategic Populations.

5. Reasons From Theory Why I Think The ID3 And Genetic Algorithm Voting Techniques I Propose Would Be Difficult For Strategic Voters To Manipulate

As mentioned in Section 1. Computational Social Choice Voting Background, in a Multi-Candidate voting system, there are situations such as the Nader-Bush-Gore election where it would be rational for some voters to vote for someone other than their true preference. One would think that is just an artifact of the plurality system we use in the United States. Unfortunately, it is not. The Gibbard-Satterthwaite Theorem says that whenever the voter must choose from among three or more candidates, such a situation must arise. This is in spite of how clever we are in designing the voting scheme[22].

However, this rule only applies to the methods where we choose a single output. Instead, I propose systems that return a relation, and perhaps a relation with the range containing real or floating point numbers.

Bartholdi and Conitzer found that this manipulation problem does not exist when there are a large number of candidates. Finding the manipulation is NP-complete as the number of candidates increases[23]. Unfortunately, Conitzer[24] showed that a single voter or voting block can manipulate a social choice system with few (finite) number of candidates, if the social choice method is polynomial time complexity. They simply try every possible preference order, plug in the other voter's votes and compute the result. They take the preference that gives them the best result.

---

21 GAMUT. Also see Sandholm, Tuomas W, Gilpin, A., Conitzer, V., Mixed-Integer Nash Equilibrium, Carnegie Mellon University Research Showcase 1-1-2005.

22 Conitzer, op. Cit., citing Gibbard, A. 1973, "Manipulation of Voting Schemes" *Econometrica* 41, 587 – 602.

Note, a voter would have no incentive to be dishonest in a dictatorship, where the choice of a single person was the person elected. Nor would they have an incentive to be dishonest if the voting system ignored the preferences entirely and chose someone at random to be "elected". However, those are not reasonable voting schemes, and the theorem says that any voting scheme that is neither a dictatorship nor a totally random choice is manipulable.

23 Conitzer op cit and J. J. Bartholdi III, C. A. Tovey and M. A. Trick, "The Computational Difficulty of Manipulating an Election" *Social Choice and Welfare*, 1989, 6:227-241.

24 Conitzer op Cit

However, this paper's seminal result found that many voting schemes are NP-complete to manipulate if we have a large number of voting blocks with different preferences. In a democracy, a leader for a group would calculate the manipulation and have the ditto-heads vote a certain way, even though it would not be obvious to a layperson.[25] This could also be done by a firm or an environmental group trying to get a candidate elected that favors their interests. Imagine that there are n of these, each controlling a different number of ditto-heads. And assume there are m candidates. Then, for what number of candidates, m, is the manipulation problem NP-complete? The authors provide a table for the most prominent voting choice systems. For "constructive manipulation," finding a manipulation that will have a particular candidate win, the single-transferable-voting system is NP-complete for three or more candidates, plurality is never NP-complete, and in a tournament whose choice is randomized, it is NP-complete for seven or more candidates.

In contrast to an election for a person to fill a position, e.g. President of the United States, I propose that voters develop and choose a budget or a penal code. These have large numbers of alternatives. Imagine an ID3 system with the n binary Titles. The penal code simply is permitted or not permitted. Then, there are 2^n possible relations between the values of the Titles and the true or false Result. Of course, a penal code divides crimes into categories such as class B felony or class A Misdemeanor, and most Titles such as type of gun would have several Attributes. Thus, there is hope that Bartholdi's result would help us design a practical computationally manipulation-resistant voting scheme.

And of course, in choosing a tax plan with coefficients, such as 37.6% tax rate for income in a certain class, there truly is an infinite number of tax plans. In Genetic Algorithms, Voters choose a rating for each "Chromosome," or possible tax system or budget. If the system permits only the weights zero or one, one has an integer programming problem. The general integer programming problem is NP-complete.

The probability that a manipulation is profitable also goes down both as the number of Voters, and presumably voting blocks, goes up, and more importantly as the number of alternatives goes up[26]. Researchers found a lower bound for the probability of a given voter having a profitable manipulation. It is $\frac{e^2}{2n^3q^6(q!)^2}$ where:

1. q is the number of candidates
2. n is the number of voters
3. e is the percentage of times the social choice function differs from a dictatorship

My techniques generate a number of candidates exponential in the number of Titles; note there is a factorial in the denominator of this probability expression; this is very favorable to voting among decision-trees.

On the other hand, probability distributions are important in using NP-complete. An NP-complete proof for a problem says that there are some problems that cannot be solved—but there may be infinitely many or a high percentage of problems that are resolvable. As a trivial

25 The term "ditto-head" is taken from a phrase for the followers of Rush Limbaugh, a Conservative talk-radio show host—see the Wikipedia article on this show.

26 Marchus Isaksson, Guy Kindler and Elchanan Mosel, "The Geometry of Manipulation—A Quantitative Proof of the Gibbard Satterthwaite Theorem" *Combinatorica* Volume 32 Number 2 (2012) 221-250 DOI: 10.1007/s00493-012-2704-1

example, in the Republican primaries for the 2012 Presidential election, an Obama supporter knows that an extreme Republican such as Rick Santorum or Ron Paul would be easier for Obama to beat. Thus, they could "manipulate" the election by entering the Republican primary and voting for such a candidate. Unfortunately, most meaningful voting or social choice rules have a reasonable probability of being manipulable, finding a manipulation in polynomial time, even though the manipulation algorithms are in NP.[27] Some of the calculations assume that each voter's ranking is randomly chosen from a uniform distribution—there isn't a tendency for groups of voters to prefer a certain candidate.[28]

More favorably, one can engineer voting systems that are very difficult to manipulate. One combines different candidate mechanisms, by having several rounds. In some hybridization possibilities, the first rounds might use a different mechanism, e.g. Borda, to eliminate some candidates. The remaining candidates are eliminated by single-transferable voting.[29] Also, one can combine several steps, e.g. Borda Voting for the first rounds and for the final rounds, and get an algorithm whose manipulation problem is NP-complete.[30]

The ID3 mechanism I propose involves many instances of a voting scheme, one on each Node in the Decision Tree. The Genetic Algorithm for budgeting uses a voting scheme for each Generation of Chromosomes, trial budgets.

A Decision Tree for a tax code would include Formulae at the leaves. For example, a person with a particular citizenship status, marital status, income bracket and sources of income would pay a $\mathbf{tax} = 0.37 \times \mathbf{I} - 3000 \times \mathbf{C}$ where I is the taxpayer's income and C is the number of children. The numbers "0.37" and "3000" could be Median Parameters. Independent of approving the tax code, each year, the taxpayers can give the number they wanted for the income coefficient and deduction for number of children. The number used for computing the taxes would be the median of the expressed preferences. The Voters would go to the election office on any day to change their numbers. As citizens leave the jurisdiction, the median would be readjusted.

Chevaleyre et al.[31] developed techniques and analyzed the space complexity of compilation so that if most of the votes are known, the system can calculate the effect of the remaining Voters[32]. That is, if x% of the votes were known, it could pre-process these, so that as

27 Vincent Conitzer, Tuomas Sandholm and Jerome Lang, "When are Elections with Few Candidates Hard to Manipulate?" J*ournal of the ACM,* Vol 24, No. 3 citing Conitzer and Sandholm, "Nonexistence of Voting Rules that are Usually Hard to Manipulate" *Proceedings of the National Conference on Artificial Intelligence*2006 Boston Massachusetts.

28 Faliszewski, Piotr, and Ariel D. Procaccia. "AI's War on Manipulation: Are We Winning?." *AI Magazine* 31.4 (2010): 53.

29 Elkind, Edith, and Helger Lipmaa. "Hybrid voting protocols and hardness of manipulation." *Algorithms and Computation* (2005): 206-215

30 Also, see page 78 and 81 of *Communications of the ACM*, Vol 53 No 11 November 2010.

31 Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, Guillaume Ravilly-Abadie, "Compiling the Votes of a Sub-electorate" in IJCAI 2009, pages 97-102.

32 Chevalreyre, ibid., asked the question of how these results can be improved if one knows there are only *u* voters left to vote, for example, given a simple plurality voting rule with candidate **X** 1000 ahead of **Y**, the second highest score. If there are only 50 voters left to vote, one can predict the election. If there were 1001 voters left to vote, one would only need to know that all of them had to vote for the other candidate and, for the purposes of determining the winner, one can throw away all the other information. Lirong Xia, Vincent Conitzer, "Compilation Complexity of Common Voting Rules", American Association for the Advancement of Artificial

the last y% of the data come in, the system can calculate the final winner. I hope to develop analogous techniques to allow us to calculate these effects quickly in these systems[33].

As stated above, assume that a voting block or voter is willing to vote for something other than their own preferences in order to get a preferred candidate into office. It is sometimes NP-complete to determine if there is a set of votes they can cast that is more likely to get their candidate into office than by just voting their true preferences. But these results assume that they know the preferences of those not planning their votes. For a single manipulator, most of the voting rules specified are manipulable and the manipulator can compute what they should do by polynomial time when there are a fixed number of alternatives. However, in voting, as we know, there are many voters whose behavior cannot be predicted; they don't respond to pollsters or they say they are undecided. If there are four such groups, that makes many manipulations NP-complete to compute[36].

Intelligence, 2010 covers these ideas in detail.

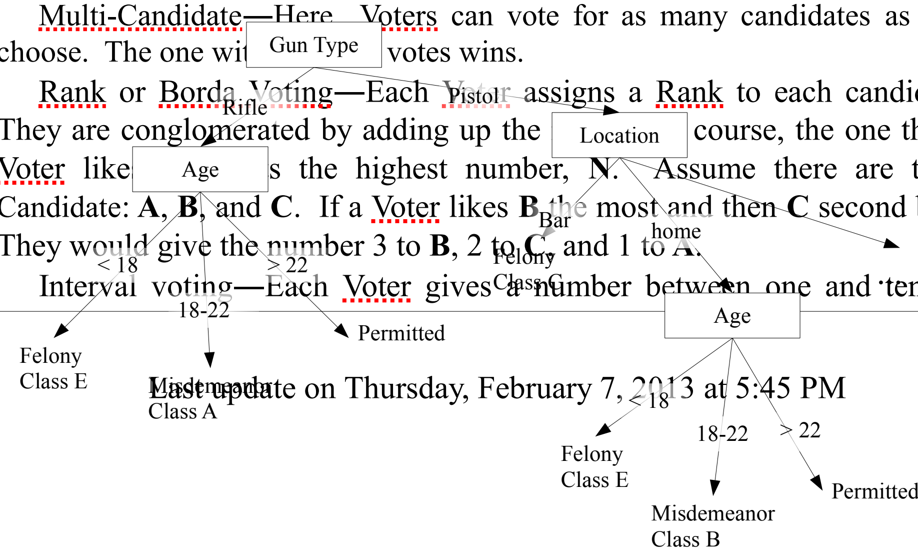
33 Last century, I developed symbolic math methods for pre-compiling mechanical engineering objects for design and engineering analysis. Assume one had a plate with a hole in it, where h was the height of the plate, w was the width of the plate, r was the radius of the hole, x and y gave the position of the center of the hole with respect to the lower-left hand corner of the plate. Develop expressions that would tell when the hole exceeded the perimeter of the plate—in other words, the object was no longer geometrically similar. Also, I developed techniques for precompiling the finite element analysis matrix that computed the stress and strains in the plate so they could be quickly recalculated when the values of x, y, h, w, or r changed. See my papers:

1. Compiler for Generating the Global Stiffness Matrix for Symbolically-Defined Regular Finite Element Analysis Grids" *Computers and Structures*, Vol 76, 461-469, 2000 (with Kyaw, M., Caplan, R, and Mogdans, D. A.)
2. "Symbolic Math Applications to Constructive Solid Geometry and Finite Element Analysis" *Computers and Structures,* Volume 59, Number 3, 561-582, 1994 (with Yun, D. Y. Y.)
3. "The Symbolic Finite Element Analysis System" *Computers and Structures*, Volume 34, No 4, 1990 (with Yun, D. Y. Y)

36 Conitzer, Vincent, Walsh, Toby and Xia, Lirong, "Dominating Manipulations in Voting with Partial Information" Association for the Advancement of Artificial Intelligence, 2011.

Appendix I: Glossary

Term	Definition
Approval Voting	This is a COAF voting system in which voters can vote for as many candidates as they wish. Each vote is added as one. Thus, the vector consists of zeros and ones. The candidate with the most total votes wins.
Approve	This means that the voters or legislators have to vote affirmative in a certain percentage, e. g. 50%, in order for the bill to become law.
Approve Budgeting	This is a method by which all the Voters or the legislators can participate in what Items are funded. All the possible Items are available for vote. The one with the most votes gets funded first. If money is still available from the revenue source, then the next one with the most votes gets funds, and so on until money runs out.
Attribute	One of the possible categories for a Title (q.v.) in an ID3 Decision Tree Process (q.v.).
Borda Voting	Borda voting is a COAF voting system in which the highest-rank vote gets the largest number and each successive vote gets one less. For example, if each Voter gives a first, second, and third choice, these are summed as 3, 2, and 1 respectively.
Chromosome	A collection of values in Genetic Algorithm budgeting. The Voters will have to to rate Chromosomes in each Generation.
Chromosome Entry	One of the values in a Chromosome. It might represent the percentage of income threshold for deducting medical expenses.
COAF	"Compact set based, One-vote, Additive, Fair"—One of several voting processes for Multi-way elections. Each person voting enters a series of numbers, one for each candidate. The numbers that each User enters are added up for each candidate and the one with the most votes wins. This can be configured into one of several types: • Plurality voting—this is conventional majority voting. Each Voter says yes to exactly one Candidate. The one with the most votes wins. The more mathematical would state this as each Voter can enter a vector with a single one. The vectors are added and the one with the highest number wins. • Multi-Candidate—Here, Voters can vote for as many candidates as they choose. The one wi[illegible] votes wins. • Rank or Borda Voting—Each Voter assigns a Rank to each candidate. They are conglomerated by adding up the [illegible] course, the one that a Voter like[illegible]s the highest number, **N**. Assume there are three Candidate: **A**, **B**, and **C**. If a Voter likes **B** the most and then **C** second best, They would give the number 3 to **B**, 2 to **C**, and 1 to **A**. • Interval voting—Each Voter gives a number between one and ten for



Term	Definition
	each Candidate. These are added up and the one with the highest total wins. This is how Olympic Skaters are scored and a winner determined.
Constitution	In the Constitution Construction Kit, the system will present several Constitutions. The Users can simulate them.
Constitution Construction Kit	This is the name for a system being proposed. Information is available in the *Constitution Construction Kit System Requirements Specification* document.
Decision Tree	This is the output of the ID3 Decision Tree mechanism available to the Voter to use to develop complicated laws such as a penal code, tax code and a benefit code (such as for Social Security). An example Decision Tree for the gun control section of a penal code (somewhat oversimplified) might be as shown below. *Figure 13: ID3 Decision Tree*
Demos	The word is derived from the Greek word for people and here represents the people of a country.
Ethnic Group	In some countries, it may be desired to have the political system protect some ethnic groups. Thus the Constitution Construction Kit may be configured to require voters or bills to receive support distributed over several ethnic groups. See section 2.4.
Fitness Function	Genetic Algorithms will see that the Chromosomes in a Generation that have a high Fitness Function are more likely to be used to generate the Chromosomes in the next Generation. In this work, each Voter rates each of the Chromosomes. From these, the programs will compute the Fitness Function of each Chromosome.

Term	Definition
Formula	In an ID3 Decision Tree of Type 3, one can label a Node with a computation to compute the Result. The Result will be the median of the Result of all at that Node. For example, in an ID3 Decision Tree of Type 3 for taxes, a Node might represent the individuals earning between $20,000 and $30,000 with two children and who are citizens. One Voter might suggest the Formula $500.00 plus ten percent of the income above $20,000 for the tax to be paid.
Gene	Synonym for Chromosome Entry
Generation	A collection of Chromosomes. They are awaiting the full completion of Rating by the Voters.
Genetic Algorithm	This applies the Genetic Algorithm paradigm to voting. Each voter will get to Rate several Chromosomes.
Genetic Algorithm Model	This takes a Chromosome and generates the revenue if for a tax structure, or the cost if for a benefit program.
ID3	See ID3 Decision Tree Process.
ID3 Decision Tree	See ID3 Decision Tree Process.
ID3 Decision Tree of Type 1	This simply returns `true` or `false`; for example, a person can or cannot have a gun in a certain situation.
ID3 Decision Tree of Type 2	The ID3 Decision Tree will have at each Leaf Node one of a set of values. For example, an ID3 Decision Tree representing part of a penal code will have Results that will be a crime classification such as Felony Type C or Misdemeanor B.
ID3 Decision Tree of Type 3	A Decision Tree that returns a Result which is a number. Each Leaf Node will generally have a Formula.
ID3 Decision Tree Process	Section 2.1 discusses how a penal code, tax code, etc. can be created by a series of votes on each Node. This creates a Decision Tree (q.v.). An ID3 Decision Tree Process has a set of Titles and Attributes (q.v.).
Item	A category to be budgeted by the Approve Budgeting. See section 2.3.
Item of Type One	An Item in Approve Budgeting to be assigned a fixed amount of money, say a bridge whose costs are estimated at one billion dollars. It will either be funded or not funded at all.
Item Of Type Two	An Item in Approve Budgeting that will allow Users to Approve at a specific level of funding. The system can fund any amount up to the maximum specified by a Voter, depending upon how much revenue is available.

Term	Definition
Item Of Type Three	An Item in Approve Budgeting that is per person. For example, it might be the amount that individuals who are 67 years old and who worked for 40 years should get for Social Security. Again, it can be funded at any amount up to the maximum specified by a Voter, depending upon how much revenue is available.
Leaf Node	A Node with no branches.
Lewis Dodgson	A method for converting a Ranking into the Person or Persons elected. This is defined in Wikipedia and also in J. Bartholdi III, C. A. Towey and M. A. Trick, "Voting Schemes for which it can be difficult to tell who won the election." *Social Choice and Welfare* Volume 6 no 2 1989 pages 157 to 165. Computing this is an NP-complete problem.
Manipulate	This means a voter will not vote their true preference. The canonical example is a simple plurality voting for three candidates: **B**, **N**, and **G**. A voter prefers **N** but would rather have **G** win than **B**. However, **N** does not have enough votes to win. If the population is manipulating, they will vote for **G** in order to avoid **B** winning. If they vote their true preferences, they would vote for **N**.
Median Parameter	A Voter can enter a number for it, at any time. The value used will be the median of all entries.
Merge	This is a term from the Approve Budgeting process (q.v.), Two or more Items are combined. Then the voters can vote on a new Item that includes both source Items. For example, the first group would vote to Merge an Item to support the West Cost with an Item to support the East Coast, hoping to get the support of Voters from both coasts.
Multi-Candidate	This means that the Voters will be selecting among more than two possibilities, E.Ge.g., several people running for office or several possible bills. There are three types: • a COAF system • a Ranking based system • Nth system
Multi-way	See Multi-Candidate.
Node	This is part of a Decision Tree constructed using an ID3 Decision Tree Process. It represents dividing the cases on the bases of one of the Titles (q.v.) into a group for each Attribute of that Title.
Non-Strategic Population	Used to refer to a population when simulating Genetic Algorithm or ID3 Decision Tree development. This population votes its true preferences. It does not consider the other populations and how to give a vote that would most likely create a more-preferred Result.

Term	Definition
N^{th}	This is one of the ways one can define a Multi-Candidate vote. It consists of choosing the Item for which to vote by Province.
Person	Either a User or a Voter.
Province	A region of a nation-to-be. Of course it may have another name in the country, such as "state." Each Voter can be assigned a Province.
Range Voting	Each Voter gives a number between 1 and 10 for each candidate. These are added up and the one with the most votes wins. This is how Olympic Skaters are scored and a winner determined. Also called interval voting.
Rank	See Ranking.
Ranking	One of two types of Multi-way voting systems. Each User Ranks the Candidates in order. One must specify either Single Transferable Vote or a Lewis Dodgson System. (Note that the latter is NP-complete so the system may take a long time to find the winner.)
Rating Rate	In Genetic Algorithm budgeting, each Voter will give a number from one to to ten that is an evaluation of a particular Chromosome in the current Generation.
Ratification	The Voters choose which Constitution would be used in reality. This is a Multi-Candidate vote among possibilities. It is the final step of a session of the Constitution Construction Kit.
RDF	RDF stands for Reality-Detail-Filler-in, and has no relationship to the World-Wide Web XML standard Resource Description Framework. See section 2.4 for more information.
Reality-Detail-Filler-in	This is analogous to the game master, Dungeon Master, Game Operation Director, Referee and Storyteller in other games. The RDF sets up various situations, such as the finding of large deposits of a natural resource such as oil, disasters, wars, etc. See www.wiu.edu/users/mflll/CR.odt section 15 for more information.
Result	This applies to an ID3 Decision Tree of Type 2 Process. It is a category. Thus, for example, a penal code might be created by an ID3 Decision Tree Process. It might designate felonies as Type A, Type B, Type C, etc. At a Leaf Node, the Voters will be presented with a vote as to which of the above would be chosen. If there are more than two, this will be done as a Multi-Candidate vote.
Root Node	The Node at the start of an ID3 Decision Tree.
Simulate	The Constitution Construction Kit's purpose is to allow Users to interact with several possible Constitutions, voting and playing the role of the

Term	Definition
	elected legislature, elected and appointed officials and even judges.
Single Transferable Vote	One of two possible ways for choosing the winner or winners from a Ranking Multi-way voting system.
Split	In Approve Budgeting, one can start with an Item and add two or more new Items that are only a portion of the source Item. The original Item remains. For example, one might take a health care funding bill and Split it into one Item, medical funding without abortion, and another Item with abortion funding.
State	State corresponds to "states" in a computer science state diagram and are shown as a relatively-large black diagram on a StateFlow in the Visual Drawing Tool. They are connected by Transitions. Most States will be drawn in a box designated with the legislative body or the electorate who will be voting on the bills or other items flowing through the workflow.
StateFlow	The diagram used to represent a WorkFlow.
Strategic Population	Used to refer to a population when simulating Genetic Algorithm or ID3 Decision Tree development. This population may not vote its true preferences. It will set up a game tree to determine the optimal votes to make, considering the preferences of the other populations.
Terminal State	A State at which a bill going through a WorkFlow no longer move forward. That is, it only has an incoming Transition and no outgoing Transition. It can be marked Approved, in which case the bill becomes law. It can also be marked rejected, which means that the bill will no longer be considered.
Title	In an ID3 Decision Tree Process, Titles are how the people or things being classified can be divided. For example, in developing a gun law as part of the penal code, Titles might be "gun type," "location where gun is carried," and "person's mental health status." The Voters might vote to use the "`gun type`" for the Root Node. This creates a Node for each Attribute for that Title. Then the Voters would have the option to choose the Title for each of these Nodes. In the above example, suppose the Attributes for Gun Type are "`pistol`", "`antique`", and "`rifle`". The system would create Nodes for `guntype="pistol"`, `guntype="antique"`, and `guntype="rifle"`. (See Figure 13: ID3 Decision Tree, above, under Decision Tree.) The Voter would now have the opportunity to choose one of the remaining Titles for each Node. For example, the Voters may vote to classify those having an antique gun on the basis of the "`location`".
Transition	Transitions are connections between one State and an another. Voting

Term	Definition
Transitions	occurs on Transitions. Transitions correspond to arcs in the Visual Drawing Tool.
User	See Voter.
Visual Drawing Tool	The tool with which someone can create a StateFlow in the Constitution Construction Kit.
Voter	In the Constitution Construction Kit, a person participating, particularly a member of the electorate who would be considered registered to vote and thus would participate in elections for candidates and referendums.
WorkFlow	In the Constitution Construction Kit, a representation and definition of a process by which a bill becomes law. (It is also used for other governmental activities including administrative and judicial procedures. See www.wiu.edu/users/mflll/CR.odt).

Appendix II: Acknowledgment

I thank and acknowledge Mr. Will Pittenger for his persistence, diligence, and insight in polishing this document, in style, word-smithing and most importantly, technical content.