E-CONSTITUTIONS: CONCEPTUALIZATION, THEORY, DESIGN MODEL AND EXPERIMENTAL EVALUATIONS

By

Hamed Khaledi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfilment of the requirements
for the degree of

Business Administration – Business Information Systems – Doctor of Philosophy

2018

ABSTRACT

E-CONSTITUTIONS: CONCEPTUALIZATION, THEORY, DESIGN MODEL AND EXPERIMENTAL EVALUATIONS

By

Hamed Khaledi

This project addresses the problem of collective design in cyberspace using computerized governance rules. Despite many applications of computerized rules, there is no systematic method or model to design them. I conceptualized an e-constitution as a set of rules in computer code that allocate decision rights and incentives to govern decision making process. This dissertation develops a design model consisting of a structured representation that breaks down a constitution into 14 components including a state transition function and a weighting function. As a meta-artifact, this model provides a unified architecture for governance structures in a wide range of situations including crowdsourcing, blockchains and corporate governance. This model enables use of quantifiable performance measures to evaluate constitutions objectively, liberated from the fairness criteria used in impossibility theorems. A systematic methodology is also presented to improve the performance of constitutions efficiently. As a proof of concept, I implemented a generic e-constitution in a web application and measured the effects of different factors on the constitutional performance metrics through online experiments. One finding is that approval voting is significantly superior to the plurality voting, even under a prediction voting incentive scheme.

Keywords: Design Science, Constitution, Governance, Mechanism Theory, Game Theory, Crowdsourcing, Collective Intelligence, Blockchain, Distributed Autonomous Organizations.

ACKNOWLEDGEMENTS

Hereby, I want to thank my advisor, and the chair of the dissertation committee, Professor Severin Grabski for his tremendous support and help. Moreover, I would like to express my deepest appreciation and gratitude to Professor Joshua Introne for his precious guidance and direction. I also thank Professor Bill McCarthy and Professor Frank Ravitch for their support and believing in me. Additionally, I am grateful to the system administrator, Jeremy Isaac for helping me to develop and debug the web application for the experiments.

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES viii

1. Introduction 1

2. Literature Review 6

3. Problem and Motivation 11
3.1. Collective Design *11*
3.2. Enforcement versus Implementation *14*

4. Constitutions for Collective Design 18
4.1. Iterative Design *18*
4.2. Parallel Design *21*
4.3. Range Voting *23*

5. Incentives 25

6. Structured Representation 31

7. Formalization 37

8. Weighting 40

9. Constitutional Design 48

10. Analytical Model 50

11. Propositions and Relationships 53

12. Research Method 60
12.1. Experimentation Procedure *60*
12.2. Factor Selection Guidelines *65*
1- Accuracy of selection (p): 66
2- Number of suggestions (m): 70
3- Average quality of suggestions (μ): 70
4- Variance of suggestions (σ): 71
5- Number of rounds (z): 71
Parameter-wise Summary: 72
12.3. Evaluation of Constitutions *73*

13. Proof of Concept 76
13.1. Implementation *76*
13.2. Pilot Experiments *80*

14. Results and Analysis **84**
14.1. Participants *84*
14.2. Dependent Variables *85*
14.3. Treatment (1) *86*
14.4. Treatments a, b and ab *88*
14.5. Treatments ac, ad and acd *92*
14.6. Treatments ac'd, acde and ac'de *96*
14.7. Including Control Variables *97*
14.8. Subject Level Analysis *101*

15. Discussion and Limitations **109**

16. Conclusions **114**

APPENDICES **116**
APPENDIX A: IRB Application Determination Letter *117*
APPENDIX B: Registration Page Including the Consent Form *118*
APPENDIX C: Screenshots of the Webpages for the Experiment *119*
APPENDIX D: Final Survey Webpage *123*
APPENDIX E: Control Panel for the Experimenter *125*
APPENDIX F: Computer Code of the Generic E-Constitution *126*
APPENDIX G: Database of the Website *132*
APPENDIX H: HIT Description in MTurk *133*
APPENDIX I: Constitution for Treatment (1) *134*
APPENDIX J: Price-Based Constitution *135*

BIBLIOGRAPHY **142**

LIST OF TABLES

Table 2-1: Eight components of an Information Systems Design Theory (Gregor & Jones, 2007)…….....8

Table 3-1: Situations versus Collective Action Schemes……12

Table 3-2: Different Mechanisms to Carry out Rules……..……..……...……15

Table 8-1: Distribution of Voting Rights using Linear and Concave Weighting Functions ……..…45

Table 12-1: Three Levels of Design Artifacts and Methods……60

Table 12-2: Elements of the Improvement Direction: *∇E[Q]*(δp,δm,δμ,δσ,δz)*……..…65

Table 12-3: Pairwise Comparisons between the Effects of the Mediators……..65

Table 12-4: Summary of Guidelines for each Constitutional Parameter……72

Table 14-1: Summary of the Outcomes for Treatment *(1)*……..86

Table 14-2: Summary of the Outcomes for Treatments *a*, *b* and *ab*……....88

Table 14-3: Levels of the Dependent and Independent Variables in the First Four Treatments……..…90

Table 14-4: Regression Results for the First Four Treatments……90

Table 14-5: Regression Results for the First Four Treatments Excluding Variable *n*……90

Table 14-6: Summary of the Outcomes for Treatments *ac*, *ad* and *acd*……..92

Table 14-7: Levels of the Dependent and Independent Variables in the First Seven Treatments……...93

Table 14-8: Regression Results for the First Seven Treatments……..94

Table 14-9: Stepwise Regression Results for the First Seven Treatments……..94

Table 14-10: Summary of the Outcomes for Treatments *ac'd*, *acde* and *ac'de*……...96

Table 14-11: Levels of the Dependent and Independent Variables in All Ten Treatments…….....97

Table 14-12: Descriptive Statistics and Correlations among Group-Level Variables ..……..98

Table 14-13: Regression Results for the Ten Treatments……99

Table 14-14: Stepwise Regression Results for the Ten Treatments……99

Table 14-15: Regression Results for the Ten Treatments with all Control Variables ...…...……100

Table 14-16: Subject-Level Descriptive Statistics on the 185 Final Participants from All Treatments....101

Table 14-17: Path Coefficients for Bonus as the Dependent Variable……..104

Table 14-18: Path Coefficients for Comprehension as the Dependent Variable……...104

Table 14-19: Path Coefficients for Expertise as the Dependent Variable…….104

Table 14-20: Path Coefficients for Bonus as the DV in the Reduced Model……..…..106

Table 14-21: Path Coefficients for Comprehension as the DV in the Reduced Model…….106

Table 14-22: Path Coefficients for Expertise as the DV in the Reduced Model……...106

LIST OF FIGURES

Figure 2-1: Interrelationships among Theory Types (Gregor, 2006)…….....8

Figure 2-2: Integrated Roadmap for Information Design Science Research (Deng & Ji, 2018)……..…...10

Figure 3-1: Part of the Versioning (Forking) History of the Linux Operating System (from debian.org)..11

Figure 4-1: Flowchart of the State Transition Function for Periods and Stages in Iterative Design……...19

Figure 4-2: Design Process with Multiple Parallel Suggestions per Period……...…….22

Figure 4-3: Range Voting with an Open Range of Scores Starting from 100…….……24

Figure 6-1: Data Flow Diagram of the Generic Design Model for e-Constitutions……..……..33

Figure 8-1: Distribution of Power with Respect to the Share of the Strategic Resource…….....46

Figure 8-2: Performance of Constitutions with respect to Distribution of Power……...……46

Figure 8-3: Taxonomy of Governance Structures based on the Design Model for Constitutions ……..47

Figure 10-1: Numerical Approximation of Function g(m) for m = 1 to 100 using MATLAB®……....51

Figure 11-1: Antecedents of Constitution Performance (Quality) and their Relationships……....……54

Figure 13-1: Flow of Control among the Webpages in the Website and MTurk..... 77

Figure 13-2: Timeline for one Treatment from the HIT in MTurk to the Final Survey……..77

Figure 13-3: Sequence of Events and Actions for one Participant in one Treatment…….....78

Figure 13-4: Flow of Versions in an Incremental Design Process (Example)…….....79

Figure 14-1: Path Diagram for the Full Model with Labeled Associations……..…..…102

Figure 14-2: Standardized Results of Path Analysis from AMOS for the Full Model……..……....105

Figure 14-3: Standardized Results of Path Analysis from AMOS for the Reduced Model……..107

Figure 14-4: Unstandardized Results of Path Analysis from AMOS for the Reduced Model……..107

Figure 14-5: Path Diagram with the Unstandardized Path Coefficients for the Reduced Model….…….107

Figure A-1: IRB Approval and Application Determination Letter……..…..117

Figure B-1: Screenshot of the Main Page in the Web Application……..……..118

Figure C-1: Constitution Page for Participants to Read the Constitution as Instructions……..…119

Figure C-2: Suggestion Page for Participants to Submit their Suggestion……120

Figure C-3: Voting Page for Participants to See Versions and Submit their Selection……..…121

Figure C-4: History Page for Participants to See the Versions from Previous Periods…….……122

Figure D-1: Survey Page for Participants to Answer Seven Questions after Experiment….……123

Figure E-1: Control Panel for Experimenter to Instantiate E-constitutions and Treatment Groups……..125

Figure G-1: Data Structure Diagram for the Database Used in the Web Application……...132

Figure H-1: HIT Description for the Workers in Mechanical Turk……..…….133

Figure J-1: Prices of Parallel Versions Using a Price-based Constitution (Example)……...138

1. Introduction

The open sourcing and crowdsourcing paradigms not only question the traditional hierarchical structures for collaboration, but also blur the organizational "boundaries" that separate the contributors "inside" an organization from "outsiders" (Thuan, et al., 2017). This conception of an organization is *inclusive* in the sense that it can include everyone who is *able* and *willing* to participate in the decision-making or design process. In many circumstances, groups are smarter than their smartest members, as humans have evolved to be smarter collectively (Surowiecki, 2004). In this regard, the Internet provides a seamless technology to aggregate millions of dissimilar independent ideas.

Brabham (2008) differentiated between crowdsourcing and open sourcing in that open sourcing allows anyone to contribute and modify a product, and freely distribute it, whereas crowdsourcing usually includes a policy for compensating the contributors and allows the originator to profit from the outcomes. In the end, the originator calling for solutions to its problems owns the outcomes. Brabham stressed that crowdsourcing, while cost effective, can deliver faster and even better results than top experts in most cases. Sakamoto and Bao (2011) used crowdsourcing to produce creative text solutions for a social problem and found that the best ideas from crowd were as novel and useful as the best ones from experts.

Collective intelligence is a subclass of crowdsourcing that decentralizes some decisions to the crowd. Malone et al. (2009) defined collective intelligence as "*groups of individuals doing things collectively that seem intelligent.*" They focused only on the *web-enabled collective intelligence*. Collective intelligence occurs when an information technology helps a group to reach superior results compared to results obtained by individuals (Kornrumpf & Baumol, 2014). Based on design science, Kornrumpf and Baumol developed a conceptual model to design *collective intelligence systems* for a business challenge. They framed this design process as the inverse problem of predicting outcomes of collective intelligence.

Many open sourcing methods let users modify a solution (e.g., software code) locally and share their own versions to the public, thereby forming a tree of versions. Such divergence can result in contradictions and inefficiencies when there are large externalities and interdependencies such as a shared resource or a

common objective. Generally, a resource can be allocated one way or another and there can only be one resource allocation scheme at a time. For example, a website can have only one code at a time because a server cannot execute multiple versions of a code simultaneously. Similarly, Wikipedia needs to choose one version of an article at a time. Otherwise, we would have multiple versions of the truth or "alternative facts". In a corporation, only one version of a policy can be effective at any time even though it can change over time. To have a blockchain rather than a block tree, we can only add one block of information as the next block at each point, so to have one version of the distributed ledger at any time.

In such situations, participants need to reach joint decisions. One way to reach unity of decisions is unity of decision maker, as in centralized organizations. However, central authorities are susceptible to moral hazard and abuse of power. It would be better if multiple stakeholders could make joint decisions collectively, because a group of people who do not trust each other can become more trustable as a whole. This requires some rules to collect and aggregate individual choices into group decisions. We refer to those rules as a *constitution*. A constitution distributes power and decision rights among members and controls their power by constraining their set of choices. It should specify how the possible choices and ideas are generated and how one out of many is selected. Moreover, a constitution outlines how to allocate resources and incentivize members. In a firm or a private institution, such rules are usually found in corporate bylaws or operating agreements, but sometimes they are in management control systems, partnership agreements, articles of incorporation, articles of organization or articles of association.

A constitution can be regarded as a social contract. Generally, contracting parties rely on courts, and brick-and-mortar institutions for interpretation and enforcement of rules. Traditional enforcement methods are slow and costly and depend on human interpretation, subjectivity and jurisdiction, which is ambiguous for multinational organizations and in the Internet. This can lead to uncertainties in implementation and inconsistencies among intended, actual, perceived, and anticipated enforcement, thereby increasing risk and transaction costs. Most transaction costs are due to enforcement issues and in the Internet it is impossible to use traditional enforcement that relies on physical force (Szabo, 1997).

Enforcement of a constitution is even more challenging because it usually involves multiple unspecified parties and is prone to free riding. However, if we could convert constitutional rules into computer code, computers could execute and enforce them. Computers are faster, cheaper and more trustworthy than humans are, thereby reducing transaction costs. Computers cannot misinterpret codes and do not care about jurisdiction. They execute instructions without caprice, whereas humans may refuse to obey (Yu & Nickerson, 2011). Moreover, we can test, evaluate and compare such digital rules through online experiments, which would be too costly otherwise. Fortunately, many important rules in business settings are precise enough to be convertible into computer code. ERP and EDI systems already execute many rules in corporate governance structures, managerial control systems and supply chain contracts.

Generally, computerized or digital rules are *constraints*, *outcome functions*, and *conditional executions*. Constraints include business integrity constraints, automated controls, and limits on the access levels. Outcome Functions include payoff functions, state transition functions, social choice functions and weighting functions. Conditional Executions include scheduled and conditional transactions. Combining these three kinds of rules, one can produce various processes such as conditional access level authorization, digital escrows and different voting schemes. If a constitution only consists of such computerized rules, machines can execute and enforce it. I define an *e-constitution* as the set of rules or bylaws in computer code that determine the incentive schemes, information access levels, decision rights and possible choices for collective design and decision-making. It can only include instructions for machines and not humans. For example, it cannot prohibit collusion among members or force anyone to do something. An e-constitution may have several different expressions in human language and vice versa. This project focuses on constitutions that can be implemented in computer code as e-constitutions, but it does not make any claim about formal verification, even though the proposed meta-model facilitates effective verification.

This research also presents a meta-model for constitutions defining a family of constitutions. A meta-model enables reuse and customization so that designers manipulate some parameters to generate instances that fulfil their needs (Kyriakou, et al., 2017). It defines the dimensions for constitutions and enables systematic search through the domain space of possible constitutions. In fact, innovation is a search through

a multidimensional design space (Brooks, 2010). Kyriakou et al. (2017) found that meta-models were reused more than other models especially when built by experts. In their case (Thingiverse), meta-models were used to design customized 3D objects, but they suggested generalizing this concept to software design.

The meta-model for constitutions facilitates their design and formal verification. Instead of coding each constitution, the designers can modify a generic e-constitution (meta-code) to implement different constitutions. This is especially important when there is possibility of errors and bugs in coding and formal verification is costly and time consuming. Once a generic e-constitution is formally verified, it can be customized and reused thousands of times. If a loophole is discovered, it can be corrected in the generic e-constitution and then reflected in every instantiation. It is like having a generic standard contract with default parameter values, so that for each case, the parties only specify the deviations from the default values and it is not necessary to analyse all the contract terms every time.

This concept is applicable to a wide range of collective action situations. Crowdsourcing protocols and Collective Intelligence schemes are almost entirely composed of digital rules. A *Distributed Autonomous Organization* (DAO) is essentially an e-constitution implemented on top of the application layer of a generalized blockchain like Ethereum (Buterin, 2013) or EOS (Grigg, 2017). Blockchain protocols are special kind of e-constitutions that, like constitutional laws and social contracts, rely on inducing a *Subgame Perfect Nash Equilibrium* (SPNE) for implementation. SPNE is a Nash equilibrium with no incredible threat or promise. Despite numerous applications of e-constitutions, no model or method exists to systematically design them. As a result, many crowdsourcing campaigns fail as did Wikinovel (Pullinger, 2007). The first DAO failed in 2016 precisely due to lack of a systemic design and formal verification (Atzei, et al., 2017).

Many of the top financial institutions in the world have joined a consortium to develop and utilize blockchains in the financial sector (Boreham & Rutter, 2018). These institutions are interested in designing effective smart contracts to improve efficiency and reduce transaction costs. However, there has been minimal theoretical development to systematically model and design blockchains and their applications. As Norta (2017) explained, *"[It] is evident that the lack of academic involvement is a reason for suboptimal solutions pertaining to achieving a scalable management of blockchains ...." (Page 9)*

Szabo (1997) introduced the concept of smart contract. A smart contract is an autonomously executing piece of code whose inputs and outputs can include money and other digital rights, thereby eliminating the need for trusted intermediaries or reputation systems (Juels, et al., 2016). In more recent usage, a smart contract is a program written in a Turing-complete scripting language such as Serpent or Solidity to be executed by a generalized blockchain like Ethereum (Wood, 2015). A DAO is a multi-party smart contract that governs an organization in the Internet. Norta (2015) investigated the collaboration setup-lifecycle for DAOs and explained how DAOs can make it possible to have Governance-as-a-Service (GaaS) in cloud. Norta (2017) described a conceptual setup lifecycle for the establishment of smart contracts and Distributed Governance Infrastructure (DGI), which relates to the concept of e-constitution.

The next chapter reviews the literature and describes various design science models. Chapter three further explains the problem of collective action in the Internet. Chapter four presents three simple constitutions along with the key concepts. Chapter five explores the effects of rewards and incentives in constitutions. Chapter six presents a structured representation for e-constitutions, providing a unified architecture for governance structures. Chapter seven formally defines an e-constitution as a 14-tuple, specifying a constitution as a limited set of design parameters and functions. Chapter eight expounds the weighting function in the constitution model and classifies constitutions based on that. Chapter nine is about designing constitutions and setting constitutional parameters. Chapter ten provides a mathematical model for quality of constitutions and chapter eleven presents a structural model for predicting the performance of constitutions, paving the way for experimental evaluations. Chapter twelve proposes a research method to conduct experiments and to improve and design constitutions. It includes an evaluation strategy, factor selection guidelines and a procedure for experimental design. Chapter thirteen explains the implementation of a generic constitution (meta-artifact) as a web application demonstrated through some pilot experiments. Chapter fourteen presents the results of the experiments with group level and subject level analyses. Chapter fifteen interprets the results and explains their theoretical implications and practical applications. It also describes the limitations of the study and suggests avenues for future research. Chapter sixteen summarizes the contributions of this study.

2. Literature Review

This project models a constitution as a *Distributed Collective Design Process*. Constitutions are more *distributed* than *decentralized* in the sense that Bonabeau (2009) differentiated between them. A constitution distributes the sources of power, but in decentralization, a central authority or principal delegates decisions to agents (Melkonyan, 2013). Hence, a constitution is closer to a multi-principal system than a multi-agent one. Here, a constitution governs collaborative design of an object or solution. It is more about *collaboration* rather than *collection* based on how Malone et al. (2009) delineated them, but also it relates to what Leimeister (2010) described as *collective*: "Groups of individuals who can have different attitudes or viewpoints." A constitution includes a *Group Decision gene* that binds everyone in the group to the same decision (Malone, et al., 2010). *Design* can be regarded as a series of interrelated decisions.

Modelling a constitution as a design process establishes quantifiable performance measures. I outline quality, cost and time as such measures, consistent with engineering studies (Gardiner & Stewart, 2000) and some crowdsourcing studies (Chilton, et al., 2013). I define the performance of a constitution as the value or quality of the outcome or final edition of the solution. Essentially, the quality of the outcome reflects the quality of the process (constitution). Cost is the expected total cost of reaching the outcome. Evaluating a constitution is a multi-criteria problem, but one can fix the speed (time limit) or cost (budget) and make a trade-off between the other two. Unlike the impossibility theorems (Nisan, et al., 2007), these performance measures do not depend on individual preferences and fairness criteria.

This project usually uses the word "design" as the process (set of activities) of designing, consistent with Hevner et al. (2004). They asserted that design science is a problem-solving paradigm and "Design is essentially a search process to discover an effective solution to a problem" (page 88). I use the word "Solution" to refer to the outcome of a constitution or a design process. The words "Problem" and "Task" define the characteristics of acceptable solutions and the purpose or objective of a constitution in a specific situation. A solution can be a product, a policy, a financial portfolio, a trading strategy, a resource allocation scheme or a simple decision.

Following Hevner et al. (2004), this project contributes to the prescriptive knowledge base (Λ) of the *Design Science Research* (DSR) foundations by introducing the concept of e-constitution as a new construct, and by providing a structured representation for constitutions as a new model. An e-constitution is a technology-based (digital implementation), organization-based (structure) and people-based (consensus building) artifact (meta-artifact), thereby addressing the relevance aspect of DSR (Hevner, et al., 2004). The design evaluation method is experimental, such that one can evaluate the utility, quality and efficacy of a constitution via online experimentation. However, this project only explores a small subset of the search space of possible e-constitutions through few experiments.

As Prestopnik (2010) acknowledged, a comprehensive coverage of all the three aspects (theory, design and evaluation) of a design science research can be too large for one project. He recommended addressing different aspects of such research in different papers. A research study does not need to include every issue of value, but rather the design and development of constructs, models and methods that address important social or organizational problems is a significant contribution by itself (Niederman & March, 2012).

This project belongs to the exaptation and improvement quadrants in the DSR knowledge framework (Gregor & Hevner, 2013) because it extends the smart contract paradigm to constitutions and governance structures, and also enables systemic design and enhancement of crowdsourcing protocols, blockchain protocols and DAOs as they are forms of e-constitutions. It has already been established that crowdsourcing protocols are algorithms undertaking collective action (Yu & Nickerson, 2013).

Relating to Gregor (2006), the main contribution of this project is to establish a theory for design and action or theory type *V*. Gregor identified five theories relevant to IS: (1) theory of analysing, (2) theory of examining, (3) theory for predicting, (4) theory for explaining and predicting, and (5) theory for design and action. She discussed that the theory for design can be informed by all the other classes of theory especially theory for explanation and prediction as figure 2-1 illustrates.

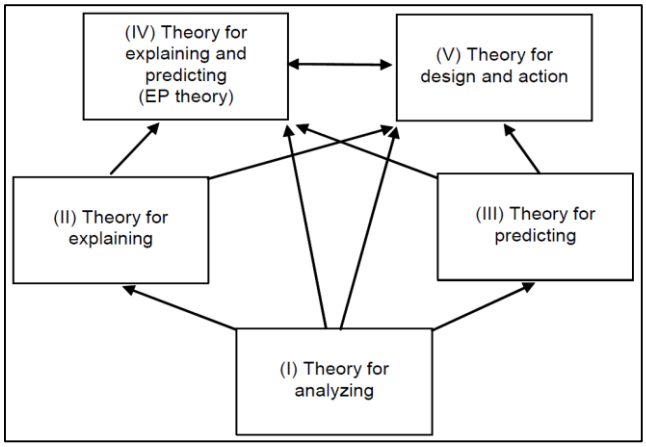


Figure 2-1: Interrelationships among Theory Types (Gregor, 2006)

Component	Description
Core components	
1) Purpose and scope (the *causa finalis*)	"What the system is for," the set of meta-requirements or goals that specifies the type of artifact to which the theory applies and in conjunction also defines the scope, or boundaries, of the theory.
2) Constructs (the *causa materialis*)	Representations of the entities of interest in the theory.
3) Principle of form and function (the *causa formalis*)	The abstract "blueprint" or architecture that describes an IS artifact, either product or method/intervention.
4) Artifact mutability	The changes in state of the artifact anticipated in the theory, that is, what degree of artifact change is encompassed by the theory.
5) Testable propositions	Truth statements about the design theory.
6) Justificatory knowledge	The underlying knowledge or theory from the natural or social or design sciences that gives a basis and explanation for the design (kernel theories).
Additional components	
7) Principles of implementation (the *causa efficiens*)	A description of processes for implementing the theory (either product or method) in specific contexts.
8) Expository instantiation	A physical implementation of the artifact that can assist in representing the theory both as an expository device and for purposes of testing.

Table 2-1: Eight components of an Information Systems Design Theory (Gregor & Jones, 2007)

Gregor and Jones (2007) proposed a structure with eight components for an *Information Systems Design Theory* (ISDT): (1) purpose and scope, (2) constructs, (3) principles of form and function, (4) artifact mutability, (5) testable propositions, (6) justificatory knowledge, (7) principles of implementation and (8)

an expository instantiation. They described each component briefly in a table like table 2-1 here. This project addresses all the eight components in the proper chapters.

Relating to Gregor and Jones (2007), the primary design goal of this research is to develop a model (as an abstract artifact) for constitutions (as product) and a method for designing them. Particularly, the scope and purpose of this model is automatic governance of design processes using machine code. This addresses the first component of the ISDT. This design goal and purpose also address the second activity (objective) of the *Design Science Research Methodology* (DSRM) proposed by Peffers et al. (2007). DSRM is a process model for conducting design science research. It has six activities or steps: (1) problem identification and motivation, (2) definition of objectives for solution, (3) design and development, (4) demonstration, (5) evaluation and (6) communication. The next chapter identifies the problem of governing collective design in cyberspace. Chapters 4 to 11 develop and design a model for e-constitutions and chapter 12 develops and designs a method to improve and design e-constitutions. Chapters 13 and 14 and appendices B to E demonstrate how e-constitutions operate and how the proposed method can improve e-constitutions efficiently. Chapter 14 presents evaluation of the performance of e-constitutions and the proposed method to design them. Chapter 15 communicates the findings and implications of this research.

Deng and Ji (2018) did a comprehensive literature review on design science papers and identified four aspects for *Information Systems Design Science Research* (ISDSR): (1) concept, (2) process, (3) outcome, and (4) evaluation. Then they categorized design science papers according to their main topic in an integrated roadmap as shown in figure 2-2.

According to their classification, this project falls under the process aspect, because it models a constitution as a design process and proposes a process or method to design constitutions efficiently. I also propose an evaluation strategy to evaluate the model and method through online experiments. The outcome of this project is a model (abstract artifact) and method for designing e-constitutions, leading to a nascent design theory for e-constitutions.

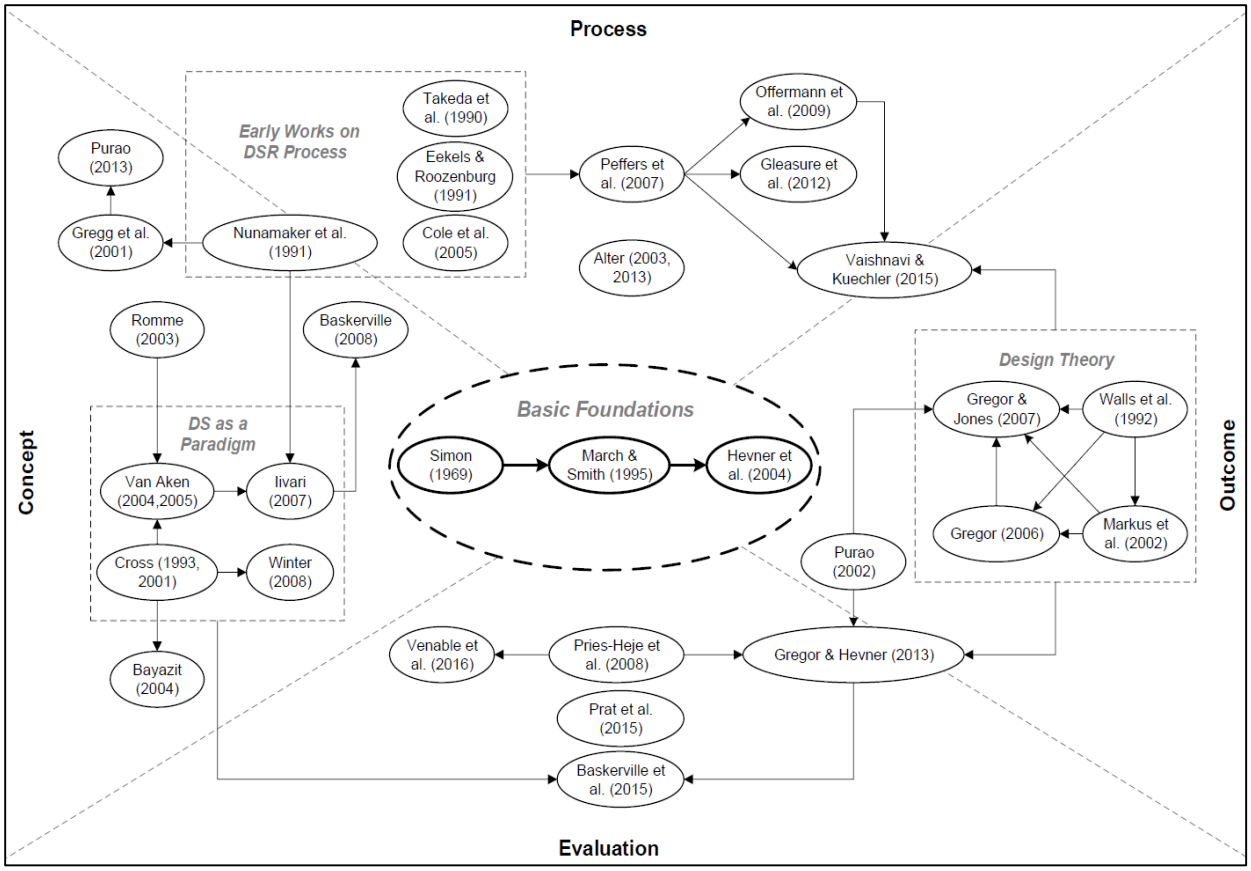


Figure 2-2: Integrated Roadmap for Information Design Science Research (Deng & Ji, 2018)

3. Problem and Motivation

3.1. Collective Design

This chapter explains the problem of governing collective design. Many open sourcing schemes let people use or modify the artifact/solution locally, resulting in a divergent tree of versions as in Figure 3-1.

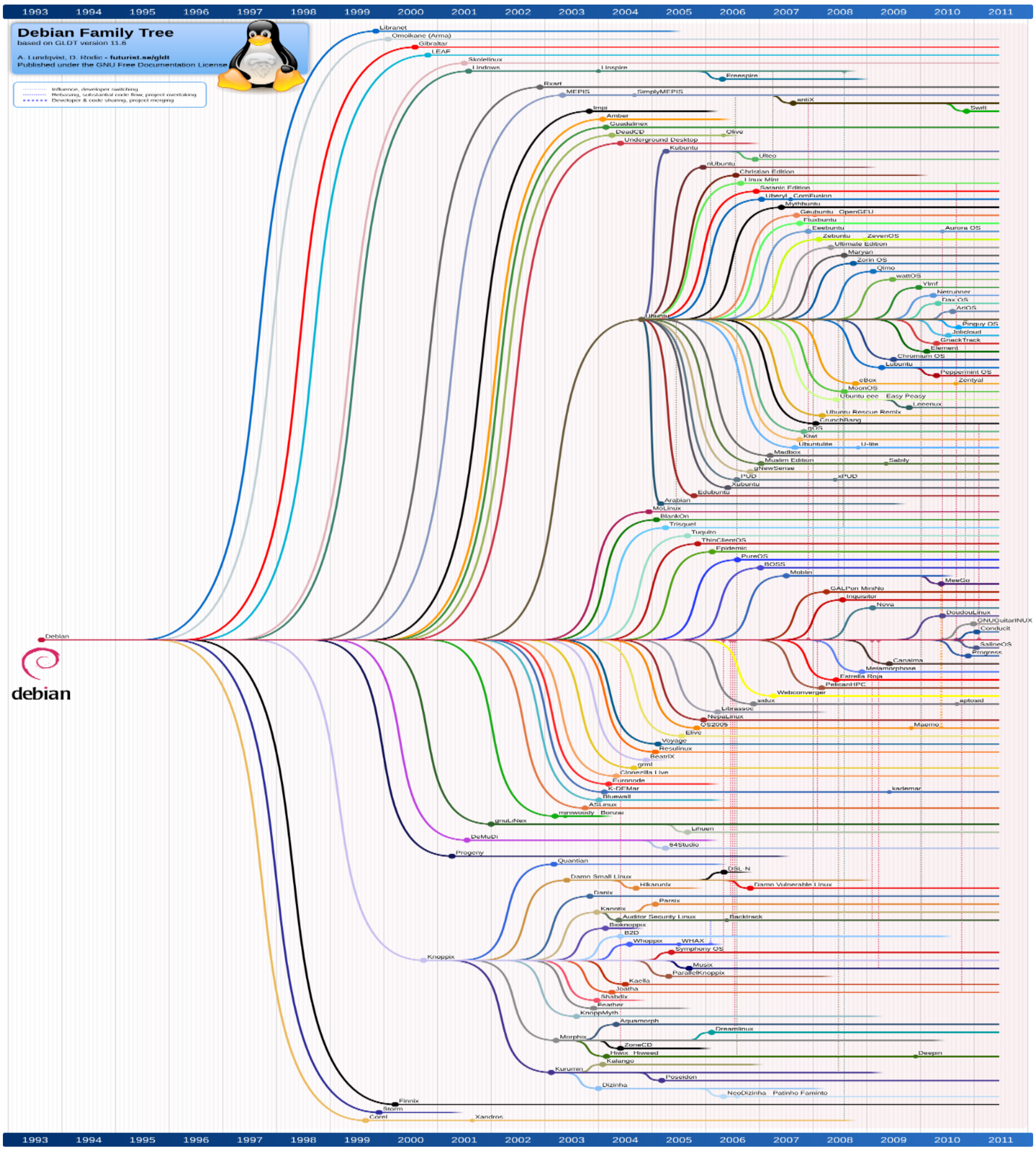


Figure 3-1: Part of the Versioning (Forking) History of the Linux Operating System (from debian.org)

This method works best when the interdependencies are low, and the costs of externalities are less than the costs of governance, so individuals can have independence and freedom. We might consider such cases as private actions. Essentially, we first need to establish the border between collective and private actions, and then we need to form the collective decision-making rules (Buchanan & Tullock, 1961). Table 3-1 maps different rules to different situations.

	Autonomy Multiple Outcomes Local Authority	Extractive Governance Unity of Outcome Central Authority	Inclusive Governance Unity of Outcome Distributed Authority
Low Externalities Private Resources	*Liberty* *(Open Source)*	*Totalitarianism* *(Error Type I)*	*Tyranny of Majority* *(Error Type I)*
Large Externalities Shared Resources	*Anarchy* *(Error Type II)*	*Autocracy* *(Error Type III)*	*Democracy, Plutocracy, Meritocracy…*

Table 3-1: Situations versus Collective Action Schemes

On one hand, when the externalities are low and cost of governance is higher than that, liberty and local decision-making are efficient, but governance (totalitarianism or tyranny of majority) would constrain individual freedom and impose one decision upon everyone even though they are not interdependent. On the other hand, when externalities are larger than the costs of governance (e.g. sharing a finite resource), we want to reach common decisions; otherwise there is contradiction or anarchy. Generally, the externalities can make decisions interdependent requiring a *group decision gene* that binds everyone in the group to the same decision (Malone, et al., 2010). A small and homogeneous group of decision makers like a family business may reach unanimity and combine compatible features and possibilities, but in general, unanimity is rarely achievable. In crowdsourcing, individual choices may not converge to one group choice even after many iterations (Ba, et al., 2001a). Disagreement among the writers in a globally writeable wiki that allows changes coming from anyone results in chaos and edit wars (Valentine, et al., 2017).

In order to reach a common group decision on each case, we need a set of rules or a constitution that aggregates individual inputs into a group output. Some define a constitution as a collection of common goals, norms, social relations and the responsibilities and rights of the participants (Kline, et al., 2017), but this dissertation focuses on the distribution of decision rights and power, determining the structure of

governance. The governance structure can be centralized (autocratic) or distributed (democratic, plutocratic, meritocratic…). When externalities are large, autocracy is categorized as error type III because while solving one problem, creates another one (moral hazard) due to the potential abuse of power. Moreover, centralized control can silence good ideas from the crowd (Valentine, et al., 2017). Acemoglu and Robinson (2012) labeled the centralized governance structures as extractive and the distributed ones as inclusive. A central authority may delegate some decisions, so that the structure becomes hierarchical (decentralized), but it is still extractive because the source of power is centralized. It is worth noting that many hierarchical structures that appear inclusive are actually extractive. They depend on some kind of pyramid scheme. Pyramid marketing is a more obvious Ponzi scheme, in which the seniors exploit the newcomers who hope they can exploit the future newcomers when they become seniors.

Malone and Smith (1988) asserted that centralization provides some economies of scale with respect to the coordination costs. They analytically showed how high coordination costs make hierarchical structures more desirable. However, concentration of power results in a *Single Point of Failure* for both ability and willingness, so that an intentional or unintentional error at the top propagates thru the organization. In many cases, this offsets the benefits of economies of scale that concentration provides (Chan, et al., 2016). Moreover, Chan et al. (2016) showed centralized evaluation systems scale poorly for crowdsourcing. Particularly, using hierarchy to evaluate thousands of submissions is both costly and time-consuming and diminishes the benefits from using crowd.

On the other hand, when a central authority allocates resources, hierarchy can cope with the free riding problem (Ba, et al., 2001a). A central authority can play the role of a principal who, as Holmstrom (1982) described, can make group penalties credible and prevent free riding. Some studies contrast anarchy to hierarchy (Fidler, 2008) or argue that a benevolent dictator is better than anarchy (Jain, 2010). Here I focus on a third option: *distributed authority*, where people can participate in making decisions without hierarchy or superiority. Edge and Remus (1984) conducted several experiments in simulated business settings, and found that the egalitarian groups demonstrate higher performance than hierarchical groups, and participants

in the egalitarian groups are more satisfied with their tasks. They found that groups with supervisors performed less resourcefully than the groups without a supervisor.

Meanwhile, a distributed system is reminiscent of the structure of neurons in the brain. After millions of years of evolution, the neurons have not developed any hierarchy among themselves and make decisions collectively without explicit superiority. Particularly, they always converge to one action towards the outside environment, despite possible hesitations or disagreements inside themselves.

3.2. Enforcement versus Implementation

As mentioned before, a constitution or governance structure consists of a set of rules. There are different mechanisms to make people follow those rules. Here, I classify those mechanisms into two categories:

1- *Enforcement* imposes ex-post costs on violation through incentives and penalties. It relies on economic mechanisms or behavioural theories to deter violation or incentivize adherence. From accounting and auditing perspective, *detective control* mechanisms are in this category.

2- *Implementation* imposes ex-ante costs on violation or makes it impossible (infinite cost) via laws of nature and physics. It relies on physical mechanisms or mathematical principles to prevent violation. From accounting and auditing perspective, *preventative control* mechanisms are in this category.

Some rules can be implemented and/or enforced and some rules can only be implemented or enforced. Usually we use *must* or *must not* to express the rules that are to be enforced and use *can* or *cannot* to express the implemented rules. For example, one may enforce the rule "*Unauthorized people must not enter this area*" by enacting a fine of $200 on violators. Alternatively, one may implement the rule "*Unauthorized people cannot enter this area*" by a lock on the door that is costly to break.

Social norms and criminal laws fall under the enforcement category. The incentives or penalties that enforce a rule can be extrinsic, intrinsic, financial, reputational, etc. Friedman (2000) explained how reputational enforcement can replace institutional enforcement in cyberspace. Friedman asserts that it is

hard to enforce laws and contracts in the virtual world upon parties we might not know where they live. In such situations, reputation becomes important and can convince parties to comply with the contractual terms in order to preserve and improve their reputations. However, it can only work for positive reputation because negative reputation can be abandoned with a new digital identity. Generally, in the cyberspace, there is no way to penalize other than by taking previously accumulated positive reputations or balances.

Digital rules and information security measures fall under implementation category because computers follow the laws of physics and mathematics (nature) and implementing rules in a machine applies those laws. *Digital Rights Management System* (DRMS) is an example of digital rules. A DRMS is a computer program that restricts the usage and distribution of a digital product and implements the terms of transferring digital content (Radin, 2000). A DRMS can prevent the copying of a content, or erase it after an agreed upon time or associate it with a specific machine and make it unusable anywhere else. It may eventually replace the intellectual property laws to govern the distribution of rights. This system is more trustworthy than humans are because it is incapable of deviating from the terms (Radin, 2004).

Enforcement can depend either on a central authority (an enforcement agency) or on inducing a SPNE (Subgame Perfect Nash Equilibrium) among the enforcers. Similarly, Implementation can depend wither on a central authority (implementer) or on inducing a SPNE among the implementers. Table 3-2 provides examples for the four possibilities.

	Central Authority	SPNE
Enforcement	Corporate Bylaws	Constitutional Law
Implementation	ERP rules, DAO	Blockchain, DAG

Table 3-2: Different Mechanisms to Carry out Rules

Contracts and corporate bylaws usually rely on courts and the judicial system as central authority for enforcement. ERP rules, collective intelligence systems and crowdsourcing protocols normally rely on a server administrator or an IT provider as the central implementer.

Unlike usual contracts, a social contract cannot rely on pre-existing laws for enforcement, and thus should be self-enforcing (i.e. induce SPNE) so that the participants can opt out, but the continuation of punishment (or lack of benefits) should be enough to deter that (Kim, 2016). Similarly, a constitutional law relies on its own provisions and institutions for enforcement. More precisely, a constitution should enforce itself by inducing a SPNE. Most of participants follow the rules because they expect there are enough people who follow and enforce the rules and have enough power to penalize those who violate a rule. However, Nash equilibrium, even under dominant strategies, is susceptible to collusion unless it is a strong equilibrium (Nisan, et al., 2007). While a collusion proof constitution with a (100%) strong equilibrium is almost unattainable, a relatively strong equilibrium can safeguard against plausible collusions and result in a collusion resistant constitution. However, even that is challenging at the beginning when no precedent or norm has formed for interpretations yet.

A constitutional law should distribute resources strategically to incentivize enough participants to follow the rules. In this regard, budgeting and monetary policies play important roles. There is a reason that every country in the world has some institution to control money supply. A currency gives power to the constitution. Conversely, a constitution gives value to its currency by making it scarce and transferable. Usually a constitution uses regulations and laws to make its fiat currency artificially scarce and valuable. Even when precious metals (e.g., gold) back a fiat currency, the currency still depends on the institutions and laws established by the constitution to honor that promise and deliver the equivalent precious commodity if demanded. Accordingly, the value of a currency reflects the effectiveness of its constitution in protecting the property rights and the scarcity of the currency.

Parallel to constitutional laws, blockchains and comparable technologies like *Directed Acyclic Graphs* (DAG) bring implementers into an equilibrium (SPNE). In fact, blockchain protocols are e-constitutions that compete for the governance in cyberspace. Instead of geographical boundaries, devices and passwords form the borders and jurisdictions in cyberspace. Therefore, this newly founded “land” may cultivate its own legal institutions (Johnson & Post, 1996). While cyberspace poses a threat to the local institutions and constitutions, it paves the way for global institutions. A few years ago, Musiani (2013) predicted that the

relationship between algorithms and rules addresses the problem of governance in the Internet. I envisage that it will address the problem of governance.

This project focuses on the bottom left quadrant of table 3-2 and proposes a design model for the family of e-constitutions that rely on a central authority for implementation, such as crowdsourcing protocols, DAOs and ERP driven corporate bylaws. Meanwhile, while the blockchain protocols need to induce SPNE among the implementers (miners), DAOs do not need to induce SPNE because they are implemented on top of a blockchain like the *Ethereum Virtual Machine* (EVM) as if it is a central authority. This is analogous to laws and contracts that depend on constitutional institutions (central authority) for enforcement. However, digital rules have only one interpretation and distributed computers can reach consensus on the outcomes and operate as one giant virtual machine. Whereas, rules in human language can yield multiple "almost correct" results, making consensus unlikely.

4. Constitutions for Collective Design

This chapter presents three generic constitutions for governing collective design: *iterative design, parallel design, and range voting*. In all of these constitutions, the *Initial Edition* refers to the first design or edition of the solution at the start of the design process. The *Updated Edition* is the winner at the end of each selection period before further modification. The modifications to the Initial Edition or an Updated Edition are referred to as *Suggestions*. Submitting a suggestion is the same as suggesting a modification. These constitutions are democratic and the *selection* is based on voting with equal weights for all voters.

4.1. Iterative Design

General Process: *The process begins with an initial solution to the problem. Then solution evolves through several editing rounds. Each round consists of a suggestion period followed by a selection period that results in an Updated Edition for the next round. This iterates until* $\boldsymbol{T_Z}$ ***time*** *passes. Anybody can participate in suggestion, voting, or both.*

Suggestion Period: *Each suggestion period is open until a participant submits a suggestion, then it ends and a selection period begins with two choices: accepting or rejecting the suggested modification.*

Selection Period: *Each selection period lasts* $\boldsymbol{T_V}$ ***time****, during which participants can vote for or against the suggestion, but the participant who submitted the suggestion cannot vote.*

Winning Version: *After each selection period, if the suggestion has the majority votes (more than 50%), it becomes the Updated Edition for the next period and the proposer's name is announced. Other suggestions and all votes remain anonymous.*

Then the next suggestion period begins for further modifications if the design process has not ended.

Generally, a constitution has four states or stages: Registration, Suggestion, Voting and Concluded. Figure 4-1 illustrates the flowchart of the state transition function for periods and stages. With some modifications, it can be extended to other constitutional design processes.

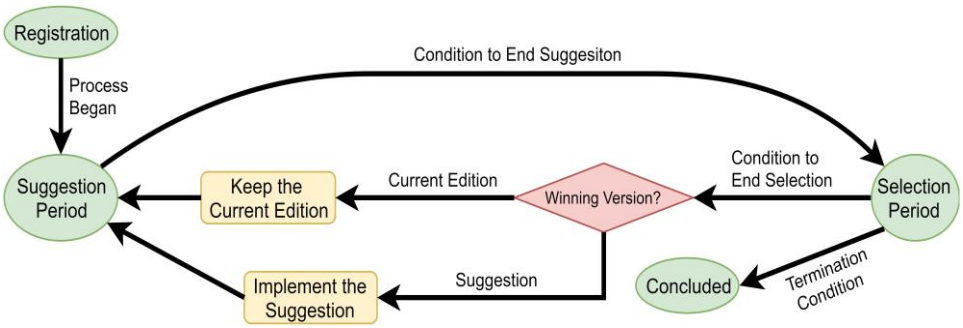


Figure 4-1: Flowchart of the State Transition Function for Periods and Stages in Iterative Design

This constitution is similar to Wikipedia except it gives the moderation power to an open crowd so that participants cannot be trusted (Moral Hazard) or tested/vetted for having expertise (Adverse Selection). In order to prevent double voting, we may need a physical registration and non-repudiation method. Any person can suggest a modification. Once someone submits a suggestion, a selection period begins and people can vote for or against the suggestion. A selection period lasts for T_V time, at the end of which, if and only if the "accept" (yes) votes outnumber the "reject" (no) votes; the suggested edition is approved and announced as the "Updated Edition". This is a binary voting scheme and does not fall under the Arrow's conditions or more precisely, Muller-Satterthwaite theorem conditions (Shoham & Leyton-Brown, 2010).

The state transits into "concluded" when a termination condition is met. Here, the condition is time T_Z, but it could be based on budget, number of iterations, or lack of successful modification for a period. It could also be limited to one iteration, after which the selected choice is final without further modification. On the other hand, the criterion can be "never" so the object or solution evolves indefinitely as is utilized.

Votes are anonymous and confidential and are cast with full privacy. Technically, distributed systems can implement such voting using group signatures (Szabo, 1997). Accordingly, the group encompasses all voters and digital signature authenticated memberships (i.e. eligibility to vote). Moreover, votes for the choices are hidden until the end of each selection period, because information cascading can influence

crowd's opinion and make it unreliable (Johnson, 2007). Revealing other participants' choices can result in bias due to information cascades (Easley & Kleinberg, 2010). Conversely, lack of interaction, isolated learning and diversity improve the accuracy of collective prediction or decision (Hong, et al., 2012).

The efficiency of collective decision-making depends on the number of individuals whose consent is required to approve a decision (Buchanan & Tullock, 1961). Buchanan and Tullock explain that smaller numbers make external costs larger, because a few individuals can impose costs on others. If this number becomes larger the external costs decrease, but the decision-making costs increase because it is harder to reach an agreement among more people. Adding the two costs, the social interdependence cost first decreases and then increases as the criterion becomes more inclusive and thus more restrictive. Therefore, the optimal number of individuals required to consent is in the middle, but it can be different for different categories of actions, and in many cases, the majority rule (50%) may not minimize the social interdependence costs (Buchanan & Tullock, 1961).

However, the majority rule can be justified if the criterion is the proportion of people who find the solution desirable or acceptable. The majority rule approves only changes that increase the acceptability of the solution assuming that participants have stable and transitive preferences. A more conservative condition (e.g. super majority), reduces the probability of accepting inferior suggestions (less error type I), but inevitably increases the probability of rejecting superior suggestions (more error type II). This leads to losing opportunities and keeping inferior editions. Nevertheless, when there exists an objective precise measure to evaluate and compare versions (e.g. ground truth), we expect high agreement among the voters and thus conservatism should not hinder progress and may improve security.

4.2. Parallel Design

When suggestions come frequently, having a selection period for each suggestion results in the need to reject every poor suggestion and thus a slow and tedious process. Moreover, exposure to one or few ideas may prime or induce similar and uniform ideas while exposure to large number of ideas can stimulate the generation of better ideas (Paulus, et al., 2013). Rather than allowing only one suggestion per period we can allow multiple suggestions each round. That will likely speed up the process and can improve novelty because it results in the generation of more suggestions independently (Little, et al., 2010). To this end, the winning criterion can be based on the plurality rule instead of majority and the clause for ***Suggestion Period*** in the constitution changes to the following:

Each suggestion period ends after $\boldsymbol{T_P}$ ***time*** *if AT LEAST one suggestion is submitted. Then the selection period begins with a minimum of two choices including the submitted suggestion(s) and the "Updated Edition." If there is no suggestion in the* $\boldsymbol{T_P}$ ***time****, the program waits until one suggestion is submitted and then the selection period begins immediately. During each suggestion period, a participant can submit only one suggestion if s/he wants to.*

Limiting the number of solutions that an agent can submit is common in crowdsourcing contests (Archak & Sundararajan, 2009). With a large group (crowd), the parallel suggestions process may result in too many suggestions per period, which may result in cognitive overload and ignoring choices in the selection period (Paulus, et al., 2013). To control that, we can shorten the length of suggestion period (T_P) and also limit the suggestions to a maximum of M. Counting the *Updated Edition*, there can be $M+1$ possible versions at most. Therefore, the first part of the above clause changes to:

Each suggestion period ends after $\boldsymbol{M}$ *suggestions are submitted OR after* $\boldsymbol{T_P}$ ***time*** *if AT LEAST one suggestion is submitted. Then, the selection period begins with a minimum of two and a maximum of* $\boldsymbol{(M+1)}$ *choices, including the submitted suggestions and the "Updated Edition."*

Figure 4-2: Design Process with Multiple Parallel Suggestions per Period

Figure 4-2 shows a process with parallel suggestions. Appendix I presents an example of a constitution with parallel suggestions. The number of suggestions per period is analogous to what Little et al. (2010) described as the number of parallel ideas. They classified crowdsourcing schemes into parallel and iterative processes and tested them via online experiments on *Amazon Mechanical Turk* (MTurk). They showed that iterative processes work better on average but parallel processes can result in higher "best quality" ideas due to their larger variance and despite their lower average. Here, a constitution conceals parallel suggestions in each suggestion period. In iterative processes, showing previous works of others negatively affects creativity and lowers diversity (Little, et al., 2010).

Generally, simultaneous exploration of multiple options results in more innovation than exploring one alternative at a time (Malone, et al., 2017). A hybrid system of both parallel and iterative processes could be more effective; thus, the average quality (iterative responses) and variance (parallel responses) should be in balance (Little, et al., 2010). In other words, there should be a balance between creative freedom and structure (Chilton, et al., 2016). The constitutional model can cover a wide range of iterative and parallel processes, and many hybrids in between via adjusting parameters like M and T_P. When $M=1$ or T_P is very small, the constitution becomes an iterative process. When M and T_P are large, it becomes a parallel process or even a *Greenfield process*, in which participants generate ideas from scratch (Yu & Nickerson, 2013). A Greenfield process can result in many creative and diverse ideas, but highly depends on participants' expertise (Ren, et al., 2014).

The selection process can be modelled as a social choice function that determines the winning choice and repeats every round. With more than two choices, it falls under the Muller-Satterthwaite conditions

(Shoham & Leyton-Brown, 2010). Precisely, the plurality rule violates the *Independence of Irrelevant Alternatives*. Consequently, two superior similar versions may share votes (steal from each other) so that an inferior version wins over them. *Approval voting* copes with this problem by letting participants select multiple choices at each period. It does not fall under the Muller-Satterthwaite conditions, because each participant classifies all choices into two sets of *Approved* and *Disapproved* versions.

4.3. Range Voting

In range voting, selectors rate each choice using a range of scores. They may range from one to seven, as in a Likert scale. The range could be only zero and one as in approval voting. Generally, range voting is a cardinal valuation and not preference ordering, thus it does not fall under the impossibility theorems (Shoham & Leyton-Brown, 2010). The aggregation of the scores can be based on averaging, summation, median or on other formulation like mid-mean (Heer & Bostock, 2010). Aggregate measures such as peer averages enhance efficiency by informing about common uncertainties (Holmstrom, 1982). For extensive score ranges, median is better than average, as it filters out outliers, and the median of individual guesstimates quickly converges to actual values when the number of people grows (Lorge, et al., 1958). Nonetheless, rating takes longer and is harder to do than voting; therefore, usually people prefer voting and do more voting than rating (Bao, et al., 2011). Moreover, Bao, et al. found that the rating results do not have as much resolution at the extremes. Hence, when only the best solutions matter, voting is more efficient and more effective than rating.

The score range can be open and infinite, in which case the score for the initial edition sets the standard and reference point for other versions. Each period takes the score of its updated edition from the previous period and participants rate the other versions (suggestions) relative to that updated edition. Figure 4-3 depicts an example of range voting assuming the range is open valuation starting from a value of 100 for the initial design. The scores of other versions are set to the median of their individual scores, so that at the end of each rating period, the version with the highest median score wins and becomes the updated edition for the next period. Each horizontal dotted line represents the updated edition in a period. Each solid line

represents a suggestion and its score improvement from the updated edition. The colored solid lines are the winning suggestions. The small red circles above and below each winning version are individual scores for that version. The median of those scores is the score of the winning version. This example is for five periods and three suggestions per period.

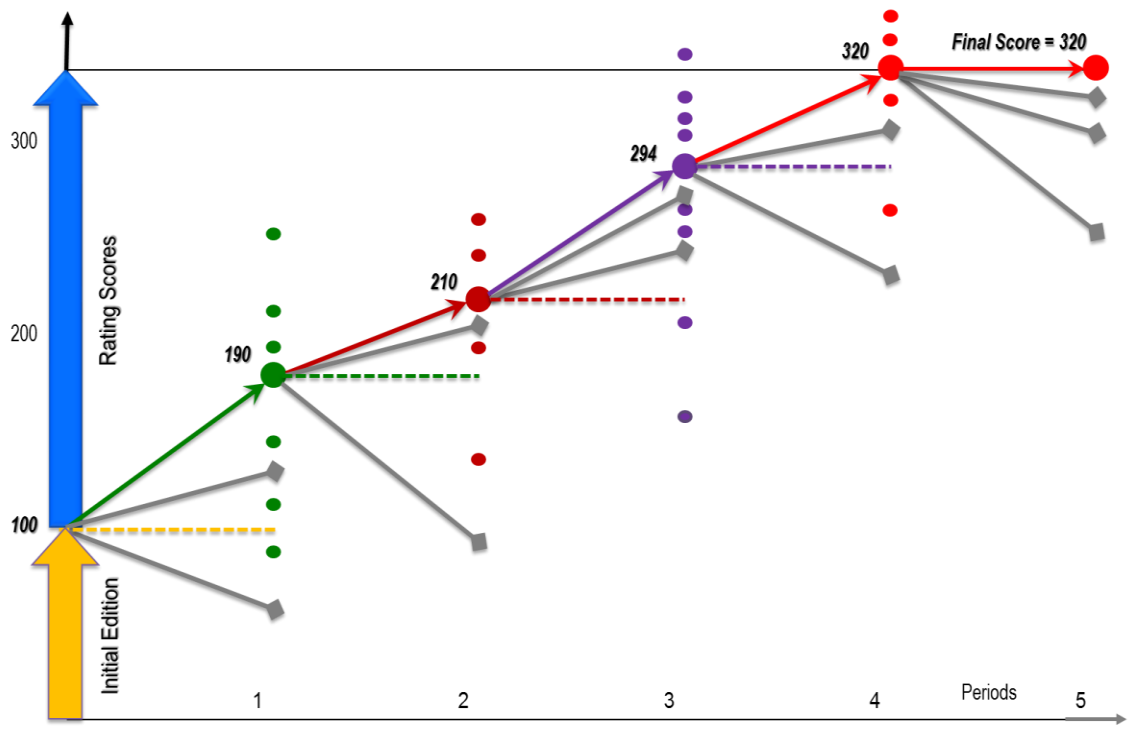


Figure 4-3: Range Voting with an Open Range of Scores Starting from 100

5. Incentives

A constitution should induce desirable activities and deter undesirable ones. The desirable activities can include more and better suggestions, more and better selection inputs, delegation to better selectors, payment of fees and investment. The undesirable activities can include low quality suggestions (spam), haphazard selection inputs (noise) and malicious selection inputs (manipulate). These undesirable actions can be deterred through ex-ante costs (e.g., fees and constraints) and ex-post costs (e.g., penalties).

Participation in an activity depends on people's *willingness* and *ability* to participate in that activity. Suggestion demands the ability to create valuable ideas. Selection requires the ability to evaluate quality. Paying fees and investing requires financial ability. A constitution can increase willingness via motivation and incentives. Incentives can also increase population's average ability by attracting more able participants, and enough motivation can also induce people to acquire higher skills (Cerasoli, et al., 2014).

Generally, there are intrinsic and extrinsic motivations. Extrinsic motivation includes money, recognition, and stakes in the outcome, which results in money for financial stakes. Malone et al. (2017) used a points system as a recognition method to motivate people. They explained that an incentive system should motivate people towards more valuable activities, and not motivate people to game (manipulate) the system or waste time. In addition, the incentive system should be fair and easy to understand.

If recognition is used to motivate individuals, participants will need to reveal their identities (or pseudo-identities) and thus it cannot motivate anonymous participation. Consequently, most of the anonymous contributions in Wikipedia are due to intrinsic motivation. Intrinsic motivation can include having fun, improving skills, love of community, etc. (Ren, et al., 2017). For example, in citizen science, gamification of scientific discoveries motivates contributions to science (Prestopnik & Crowston, 2012). Citizen science crowdsources scientific contributions to inexpert enthusiasts in public. *Mole Game*, *Eyewire*, and *Foldit* are other examples of gamification (Kornberger, 2016). However, intrinsic motivations mostly depend on problems and settings that constitutions cannot control. Hence, constitutional design is mainly based on extrinsic motivations.

Wightman (2010) classified crowdsourcing websites into four classes based on two dimensions: motivation, which can be direct or indirect, and competition, which can be competitive or non-competitive. He referred to the process and the set of rules as heuristic, which is actually a special kind of constitution. The constitutional incentives also have two dimensions, but different: reward for winning suggestions and reward for accurate selections. I first discuss the reward for winning suggestions. Many crowdsourcing websites such as *Innocentive*, *Taskcn*, *TopCoder* and *Threadless* provide monetary incentives to motivate good contributions. Adding the following clause to the constitution can implement a simple reward:

After each selection period, if the winning choice is a suggestion (not Updated Edition), its proposer receives a reward of amount $\boldsymbol{R_P}$ **.**

Horton and Chilton (2010) presented a model as a basis of a price theory for crowdsourcing. They conducted experiments on MTurk and found that workers behave rationally and work less for less payment, however, they are insensitive to an increase in task difficulty. Mason and Watts (2009) also conducted online experiments on MTurk and found that a higher payment increased the quantity of work but not the quality or accuracy. Actually, Heer and Bostock (2010) found that higher rewards slightly decreased accuracy while increasing the rate of task completion. In short, higher rewards result in more and faster but not necessarily better work. Mason and Watts attributed this to the "anchoring effect" (Ariely, et al., 2003), that is, the workers' perception of the value of their work depends on the payment for the work. They suggested that intrinsic motivation is a better driver to improve quality of work, but when such motivation is not viable, it is best to offer as little reward as possible to a large crowd who can provide enough quantity. Moreover, they found that paying a low amount to the workers made them perceive their job less valuable (thus higher performance) than not paying them at all. Contrariwise, other studies (Heyman & Ariely, 2004) found that paying nothing results in higher performance than paying a low wage. Mason and Watts (2009) ascribed this discrepancy to the workers' expectations. If participants expect monetary compensation, as in MTurk, low payment is better than no payment, but when there is not such expectation, intrinsic motivations dominate. A crowd's expectations also depend on whether a non-profit or a for-profit organization

sponsored the project (Hoffman, 2009). Monetary incentives can have no effect or even adverse effect on performance, when they diminish the participants' intrinsic motivations (Gneezy & Rustichini, 2000).

Liu et al. (2014) conducted randomized experiments on *Taskcn* and found that higher rewards result in higher quality submissions, more participation and higher quality users. *Taskcn* is a crowdsourcing website based on *all-pay auctions*, which have only one winner whereas many users may expend efforts and submit solutions. The users also gain reputation and credit for submitting good solutions and winning rewards. In addition, Wu et al. (2015) conducted several experiments in MTurk and found that with higher payments, the workers (i.e. *Turkers*) generate higher quality designs. They found that even an untrained and unskilled crowd could generate high quality designs and assess designs effectively.

With rewards, each suggestion period is like an all-pay auction or a tournament (Lazear & Rosen, 1981) with one prize and perhaps status-seeking subjects. Competitions and tournaments use information efficiently and improve risk sharing by using relative performance evaluation (Holmstrom, 1982). Moreover, experiments showed that in tournaments, rewards increase the participants' performances (Hossain, et al., 2014; Delfgauw, et al., 2013). This performance increase is larger for more able participants who are more likely to win (Freshtman & Gneezy, 2011; De Paola, et al., 2012; Bandiera, et al., 2013). In a constitution, only the highest performances matter, not the average.

Dechanaux et al. (2015) reviewed numerous experimental studies on Tullock contests, all-pay auctions and rank-order tournaments; and presented a general unified contest model. They described that the performance of contestants depends on the participants' effort, ability and luck. Contests differ from reverse auctions in that contests declare the winner(s) after the delivery of goods or services (Archak & Sundararajan, 2009). Archak and Sundararajan developed a game theoretic model to analyse the properties of crowdsourcing contests when the number of participants is large. They showed that when agents are sufficiently risk-averse, offering multiple prizes is more efficient than one grand prize, even if only one best solution is desired, whereas for the risk-neutral agents, it is optimal to reward only the best submission. Orrison et al. (2004) have two findings that I use in the analytical and design model of constitutions. They

found that in tournaments, one large prize is more effective than many small prizes, and the number of players does not affect the average effort level if the distribution of noise (i.e. unknown ability) is uniform.

Having stakes in the outcome may incentivize better participation including more accurate selections. People are more likely to vote thoughtfully and truthfully when they share a stake in the outcome. However, when the selectors do not have a stake in the outcome, we need a criterion for accuracy to measure the accuracy of selection inputs. The only endogenous criterion is the selection outcome itself. If there were any better criterion, we would use that instead of human selection inputs. Therefore, the selectors define the quality or correctness of versions (i.e., ontological relativism). The accuracy of individual evaluations is measured based on their alignment with the aggregated selection outcome.

Shaw et al. (2011) experimented online crowdsourcing in MTurk and tested different incentive schemes to motivate workers to give an accurate qualitative assessment of a content. They achieved the highest performances through financial incentives that tied workers' payoffs to the agreement of their responses with the majority responses. Their explanation is that thinking about others' responses causes confusion and higher cognitive demand, leading to more engagement with the question. An alternative explanation is that the workers perceived the open crowd more trustworthy and less corruptible than a central authority.

If selection is based on plurality voting, the criterion for the quality of choices can be the number of votes the choice received and the best voters are those who aligned with the majority. I assume that the voters cannot or do not want to coordinate among themselves and collude on voting for an inferior version. Therefore, the choice they think *will* win is the one they think *should* win. As a result, in (Nash) equilibrium everyone votes for the version they predict others will vote for, thus it can be labelled as *Prediction Voting*. Sakamoto and Bao (2011) compared prediction voting, Likert scale rating and other evaluation methods. They observed more participation in prediction voting. Moreover, prediction voting is more efficient because the evaluators focus on the best solutions instead of all of them, and in constitutions only the best solutions matter. To reward prediction voting, one can add the following clause to a constitution:

After each selection period, those who voted for the winning choice receive a reward of $\boldsymbol{R_V}$.

Having stakes in the outcome can also incentivize better suggestions. A hybrid method is to give shares as reward for the winning suggestions, so their proposers share the value they add. This dilutes the total shares but increases the total value. Particularly the reward shares can be proportional to the contributions that the proposers made to the solution. To this end, we need an evaluation method that indicates how much improvement the winning suggestion made. This approach can work best with rating scores. The constitution issues extra shares for the proposers of successful modifications, proportional to the increase in the score. This provides more incentives for superior suggestions because the winners will own shares of the solution proportional to their contributions. Meanwhile higher rating scores result in more new shares and more dilution of existing shares. Therefore, shareholders have a systematic bias to underrate suggestions. To control for this conflict of interests, we may exclude the shareholders from the rating process making it the opposite of plutocracy.

Like voting, the criterion for the quality of individual ratings can be the group results. Shaw et al. (2011) examined several incentive schemes and found that rewarding inexpert raters for giving scores close to the majority scores results in the highest collective rating performances. When selection is based on median of the rating scores, the best raters are the ones whose scores are closest to the median scores. The following clause rewards the score that are closest to the median:

*After each selection period, for each choice, the **rater** who gave a **rating score** closest to the median of the **scores** receives $\boldsymbol{R_V}$ reward. If there is a tie for a choice, the reward is divided equally among the **raters** who were closest to the median for that choice.*

In a tie, the reward is divided amongst the raters to deter collusion. Theoretically, a participant can rate multiple versions and receive multiple rewards in a period, but this makes it prone to haphazard ratings. To deter such rating, one can make rating each choice time-consuming or costly as described in chapter 11.

With monetary rewards, the selection process turns into an algorithmic mechanism. A mechanism is an implementation of a social choice function with asset transfers (payments and rewards) amongst the members (Jackson, 2003). Simply put, mechanism theory is concerned with manipulating rules of the game (i.e. selection process) and payments to agents to direct their decisions so as to realize specific desired

outcomes. In an incentive-compatible mechanism, the agents do not benefit from gaming the system, so truthful selection is the dominant strategy for every agent (Nisan, et al., 2007).

Horton and Chilton (2010) asserted that game theory and mechanism design are not very useful in crowdsourcing, because crowdsourcing is not about workers revealing private information but rather exerting effort and performing tasks. They explained that when output is observable and highly correlates with effort, there is not a moral hazard problem. However, even though output is observable, its evaluation can be costly and evaluation by crowd has complexities that only mechanism theory can address. Meanwhile, while humans are not perfectly rational, automated computerized agents can closely resemble a homo-economicus (Norta, 2017).

6. Structured Representation

This chapter develops a high-level design model for e-constitutions, defining their domain space. It provides the blueprint and principles of form and function for constitutions and the processes for implementing them, addressing the third and seventh components of ISDT (Gregor & Jones, 2007).

Pederson et al. (2013) did a review on crowdsourcing literature and provided a conceptual model with six components for crowdsourcing: problem, process, people, technology, outcome and governance. They asserted that governance is the key success factor in crowdsourcing, but minimal research has been done on it. Tuan et al. (2017) highlighted the high demand for, yet the lack of, a holistic model for crowdsourcing processes. They suggested *Business Process Crowdsourcing* as one such model with three stages, but it is more of a plan rather than a governance structure. Wu et al. (2015) proposed another plan as a methodology for crowdsourced design. It has four stages: specification, validation, execution and evaluation. Similarly, Ren (2011) decomposed a web-based crowdsourcing project into four stages: identify the crowd, request ideas from the crowd, evaluate the ideas and retain the crowd. Later, Ren et al. (2017) used this model to compare two cases and concluded that organizers should actively motivate crowd based on a top-down model rather than hoping the crowd commit to the campaign as in bottom-up models. As such, the model presented in this research for constitutions is a top-down holistic one with a limited number of parameters and functions that one may call the genome of e-constitutions.

Malone et al. (2009; 2010) classified the building blocks (genes) of collective intelligence into four types of "*genes*": task (What? create or decide), staffing (Who? crowd or hierarchy), incentives (Why? [extrinsic or intrinsic]) and structure (How? collection or collaboration). The create and decide tasks are akin to what Leimeister (2010) described as generating new solutions and evaluating them. Yu and Nickerson (2011) also classified crowd activities into creation and decision. They followed the principles of DSR and evolutionary computation to create a "crowd idea generation process" as an IS artifact, in which a solution evolves through random variations (mutations) and combinations of existing solutions. Their

experiment showed that a system with combination induces more creative (practical and original) ideas than a system without it (control). They recommended this approach for macro institutional innovation.

Human Based Genetic Algorithm (HBGA) is a system that outsources the innovation (create) and selection (decide) operations of genetic algorithm (GA) to human agents, while computers perform the organizational functions and control the flow of process (Kosorukoff, 2000). Some studies refer to HBGA as *interactive genetic algorithm* (Bao, et al., 2011). Kosorukoff (2001) stated that HBGA is a multi-agent system that combines intellectual power of human agents with the coordination power of computers. He described that some agents are convergent thinkers who tend to participate in the selection process while others are divergent thinkers who are more creative and propose solutions. HBGA takes advantage of both. It is robust because it does not depend on individual agents performing particular functions. The contributions of participants can be regarded as deliberate and directed modifications instead of random mutations, so HBGA is closer to Lamarckian evolution than Darwinian evolution.

Yu and Nickerson (2013) considered some "meta-parameters" in the HBGA algorithm to explore the "higher level" design space of different systems. They explained that information technology made a new class of unexplored organizational structures possible. Kosorukoff and Goldberg (2002) asserted that HBGA is a kind of organization that is more reliable and effective than conventional organizational forms. They analysed two social systems based on HBGA and explicated how "Human Computation" can organize workers like a system with humans as its parts. They explained that evolutionary human computation accommodates and utilizes human creativity in a constructive manner.

The constitution model developed in this research expands HBGA and human computation systems to cover a wide range of protocols and governance structures. The *Suggestion Interface* addresses the idea generation or creation task and the *Selection Process* includes the evaluation or decision task. This model also incorporates other forms of participation and human inputs such as delegation (proxy voting), betting, investing and spending. Moreover, the model includes other structural components such as *Filtration*, *Sorting* and *Weighting* as Figure 6-1 illustrates. Green arrows denote transfer of asset. They can have an incentive or deterrence effect on the participants and change their states (balance, share, eligibility…) or

the system states (treasury balance). Bold black arrows indicate the flow of versions (objects) and thin black arrows carry numerical data. Yellow boxes are stock variables. Red ovals are functions and conditions. Blue clouds represent external information entering the process. Blue arrows denote the flow of decisions and information to and from the process. Parallelograms represent the interfaces for participants.

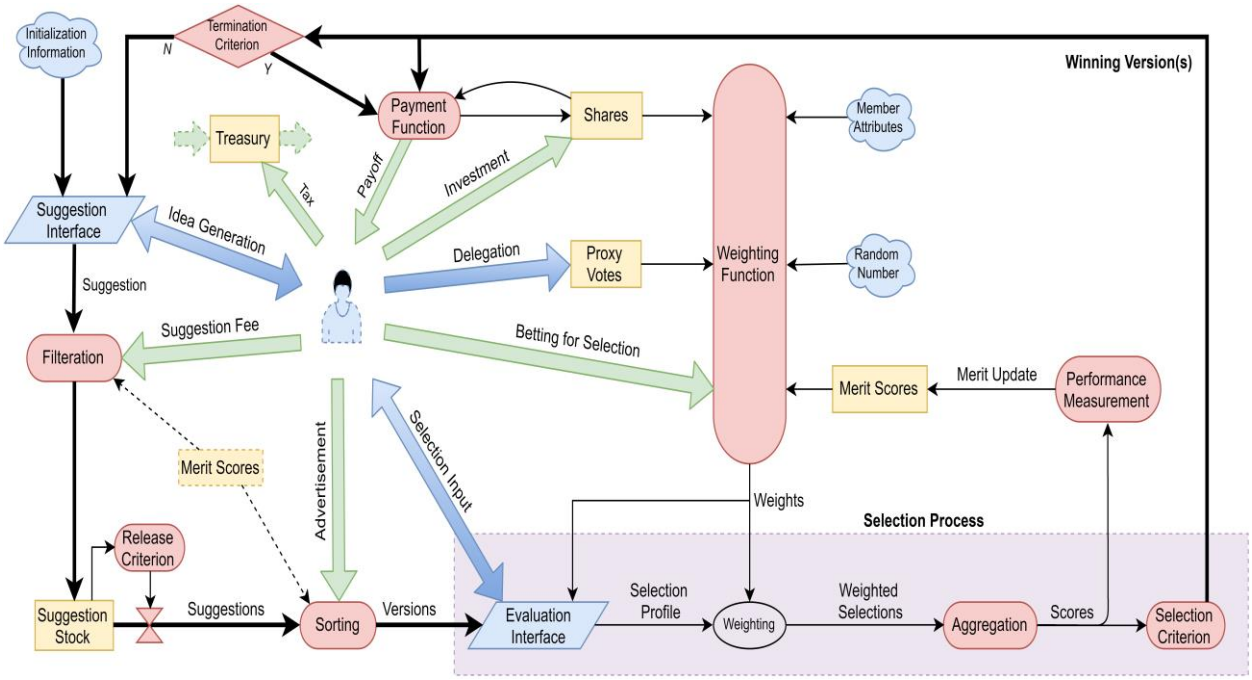


Figure 6-1: Data Flow Diagram of the Generic Design Model for e-Constitutions

An interface mediates the flow of information and structures the interactions between internal and external entities (Kornberger, 2016). The suggestion interface is essential to collect ideas and alternative solutions. Leimeister et al. (2009) emphasized the importance of managing the idea generation process. The suggestion interface controls how people propose different versions for the solution and determines the sources of innovation, creation and generation of ideas and the boundaries of the organization. It can include one person, a group of authorized experts, employees or a crowd in the Internet. This boundary is a trade-off between diversity and in-depth expertise (Leimeister, 2010), but open involvement regardless of expertise can bring about more novel ideas (Davis, 2015). Moreover, electronic participation can result in more creativity and idea generation compared to direct meeting and brainstorming (Leimeister, et al., 2009;

Paulus, et al., 2013). Some studies have shown that direct interaction could undermine creativity and dissuade contribution of novel ideas (Mullen, et al., 1991; Lorge, et al., 1958; Gallupe, et al., 1992). Confidential and independent contributions work better than open discussion forums, which deter uncommon suggestions (Kahneman, 2011). In fact, it is better to have independent participation (Yu & Nickerson, 2011) and lack of communication among participants (Bao, et al., 2011). However, the suggestion interface can coordinate participants' idea generation by sharing specific information (e.g. number of submissions), but mainly it mediates collaboration and coordination by showing previous designs as in HBGA crowdsourcing design processes (Yu & Nickerson, 2011). The suggestion interface can also impose formatting for submissions.

The suggestion interface provides the participants with a set of information from internal and external sources and enables them to submit their suggestions. One piece of external information is the specification of the problem to be solved. It determines the nature of the solutions and the objective of the process and relating to Ren, et al. (2014), it is the first force that affects the type and quality of the generated ideas. Another piece of external information is the initial edition(s) in the first period. Yu and Nickerson (2013) called the initial editions the "seeds". Seeds can come from the crowd, experimenters, experts or a computer program. It can be large or small or just one edition (e.g. status quo). It can be blank so that crowd generate the first set of versions from scratch as in the *greenfield* idea generation system as Yu and Nickerson define.

The suggestion interface incorporates internally generated information as well. It can present the best version(s) from the previous round to participants to build upon them. A more complicated interface may algorithmically combine the best versions for participants or enable the participants to combine them. Kosorukoff (2001) focused on solving problems by combining existing solutions. Yu and Sakamoto (2011) described a sequential combination process in which one crowd generated initial designs and another combined them. They found that combination improved both originality and practicality features in generations (i.e. rounds). In addition, Yu (2011) and Nickerson et al. (2011) used combination to aggregate ideas of multiple participants and found similar results regarding originality and practicality of designs through generations. Likewise, Yu and Nickerson (2013) found that a sequential combination system results

in significantly more creative designs than a greenfield idea generation system after three generations. On the other hand, Ren et al. (2014) showed that modification results in better outcomes than both combination and greenfield systems in all dimensions (divergence, relevance and effectiveness). In a constitution, the suggestion interface is essentially a modification platform, which can also support combination.

A constitution needs a selection process to decide and choose one alternative from a set of alternatives and eliminate other ones. Wu et al. (2015) stated that the most important part of every crowdsourcing system is effective selection of design choices. The selection process specifies the sources of selection and distributes power and decision rights among the participants. The sources may consist of one person (autocracy), a group (oligarchy), crowd (democracy, meritocracy, plutocracy) or a computer program (cryptocracy?!). A computer program can perform selection if the quality of solutions is computationally assessable. To incorporate human judgement, the selection process has an *Evaluation Interface*, which provides participants with relevant information and collects their assessments in proper format. A form of selection inputs is voting, which is a unity vector with *1* for the preferred choice and *0* for other choices. Approval voting is a vector of *1*'s and *0*'s. Rating is a vector of numbers in a range.

The weighting function can be a constant and weight all evaluations equally. Alternatively, it can depend on some measures such as participants' states (shares, merit scores, proxy votes), their periodical spending (betting amounts), external information (random number, attributes) or a combination of these factors. The weighting function can be nonlinear and multivariate as will be discussed in chapter eight. Each period, each participant can submit one selection input with one weight and cannot distribute the weight across multiple selection inputs. Voters have strong incentives to use all their voting power (weight) only on their most preferred option (Scott & Antonsson, 1999).

The selection process includes an *Aggregation Function* that combines individual selection inputs and gives a vector of *Aggregated Scores* for all versions in a period. Depending on the format of the selection inputs, the aggregated scores can be the number of votes, the median of the ratings, market prices, etc. Based on the aggregated scores, the *Selection Criterion* determines the winning choice(s) at the end of each round. It can be maximum (e.g. number of votes), minimum (e.g. evaluated costs) or meeting a condition

(e.g. more than 30% votes). The selection criterion also includes the condition to finalize the selection period and start the next round. This criterion can be based on time or selection inputs, so that the selection period ends when enough votes are cast, and the selection is statistically conclusive (Ertekin, et al., 2013). Filtration and sorting can facilitate better evaluation and improve the accuracy of selection. They can be functions of costs, previous performance and time. The suggestions stock variable is an object vector that accumulates valid suggestions. The *Release Criterion* incorporates the condition to end the suggestion period and release the accumulated suggestions for sorting and evaluation. It can be based on time or the number of suggestions or both.

A constitution needs a source to supply the money for the rewards and expenses. If an external authority supplies the money and controls the source, the constitution is *incomplete* because it does not include the source of money supply. If a constitution is financially self-sufficient and (weakly) balances the budget, I call it *autonomous* because it does not depend on an external authority. A constitution may issue money (currency) through the *payment function* as blockchains do. Other possible financial sources include *membership fees* (taxation), *participation fees*, *betting*, *advertisement auction* and *investment* by participants or crowdfunding when participants can buy shares. The shareholders may share the ownership of the solution. This not only supplies money but also can provide incentives for better participation. To this end, the solution should have some value. Hong and Page (2001) interpret the value of a solution as the equilibrium price of the outcome in a market. Generally, stakes in the outcome and the reward function can incentivize valuable participation, whereas the costs and fees can deter haphazard participation.

7. Formalization

Generally, a constitution is an automaton with a function that determines the permissible decisions for each person based on the state of the person and the system. For example, a person can only spend less than or equal to his balance or can submit a rating score if has paid the selection fee and has more than 10 unit shares. I formally define a constitution as a 14-tuple ($\boldsymbol{N, \eta_p, \eta_v, I_p, I_v, G_p, G_v, \Phi, \Psi, \theta, \theta_o, \delta, W, F}$) that includes 7 sets, two conditions (binary functions) and 5 vector functions. They cover all elements and units in constitutions, and their variations generate different instantiations of constitutions (i.e. mutability), addressing the second and fourth components of ISDT (Gregor & Jones, 2007). The components of a constitution are defined as follows:

$\boldsymbol{N}$ **:** *The set of possible participants or agents*. I define $n_v(t)$ as the number of agents who did participate in evaluation in period t, and $n_p(t)$ is the number of agents who proposed a suggestion in period t, and $n(t)$ is the total number of agents who participated in some way in period t.

$\boldsymbol{\eta_p}$ **:** *The set of information provided in the suggestion interface.* It includes external information such as the specification of the problem and the initial editions (seeds) which can be blank. It also includes the specification of the internal information - such as winning version - to be presented to the proposing agents.

$\boldsymbol{\eta_v}$ **:** *The set of information provided in the evaluation interface.* It can include the submitted suggestions, the updated edition(s), the weights and sometimes the aggregated scores. In some cases (e.g., reputation), it might include the identities of the proposers of the suggestions.

$\boldsymbol{I_p}$ **:** *The set of possible individual suggestion inputs.* It is the domain of the versions of the solution. It can impose formatting and restrictions on the suggestions.

$\boldsymbol{I_v}$ **:** *The set of possible individual selection inputs.* It determines the format for collecting individual evaluations. It can be votes, approval votes, scores, price biddings, etc.

$\boldsymbol{G_p}$ **:** *A condition for releasing submitted suggestions and ending a suggestion period.* It can include a maximum number for suggestions (M) and/or a time limit (T_P) in a logical statement.

$\boldsymbol{G_v}$ **:** *A function that determines the outcome of the selection process based on the scores and time*. It incorporates the condition for ending the selection period and releasing the winning version(s) and yields null when the condition is not met. The condition can be a simple time limit (T_v) or depend on the latest aggregated scores like number of votes.

$\boldsymbol{\Phi}$ **:** *A condition for filtering out suggestions.* It can depend on the characteristics of the suggestion or its proposer. It can impose a submission fee for suggestions by excluding members who did not pay the fee. It can enforce banning or suspension of members based on their past performances.

$\boldsymbol{\Psi}$ **:** *A vector function sorting the suggestions.* It can depend on submission time (chronological), the proposers' past performances, a property of the suggestions and the amount paid by the proposers for advertisement, which means auction for ranking places.

$\boldsymbol{\Theta}$ **:** *The set of states for the system and participants.* The states of the system include period number, period type (suggestion, voting, concluded…), balance in the treasury and the updated edition. The states of the participants include balance, shares, merit score, net proxy votes and their allowable activities (suggestion, evaluation, delegation, investment, all, none…).

$\boldsymbol{\Theta_o}$ **:** *The initial values for the states for the system and participants.*

***δ* :** *The state transition function for the states of the system and participants.* It is a vector function of their current states, the actions of the participants (suggestion, evaluation, delegation and investment) in each round and the outcomes of each round. It can depend on whose suggestion and selection won. This function incorporates the termination condition and switches the state of the system to "*Concluded"* when meeting that condition, which can depend on time, number of periods, idleness, lack of progress, etc.

Since the balances of the participants are states, the state transition function can reward the proposers of the winning suggestions or the voters for the winning suggestions or the raters whose scores where closest to the median. If possible, it can also assign negative rewards to penalize members or impose taxes.

***W* :** *A vector function that determines the weights of participants' selection inputs at each period.* The output of this function is one *n*-dimensional vector of weights for *n* participants each period. It can be a constant (democracy) or depend on some *n*-dimensional vectors about participants such as their attributes (oligarchy), proxy votes (liquid democracy), spending amounts (proof of work), past performances (meritocracy) and shares (plutocracy). The next chapter provides more detail.

***F* :** *A vector function that aggregates scores based on individual selection inputs and weights.* This function determines the estimated quality of the submitted suggestions. It can be the sum of the number of votes for each choice or the median of the rating scores for each choice, etc.

8. Weighting

The weighting function distributes power among the members by determining how much influence each person has on selection outcome. **Direct democracy** weights all votes equally and applies no information other than membership (i.e. citizenship). However, not all votes are created equally. Participants may have different levels of incentives and skills vis-à-vis the problem and therefore their evaluations may have different values. Hence, unequal weighing schemes may result in better outcomes.

The weighting scheme can be ones and zeros (binary) based on a criterion, which can depend on participants' attributes obtained from an external source. Attributes can include nationality, age or belonging to a specific class or group like **oligarchy** or elected members. Generally, there can be multiple classes with different voting weights. In autocracy, the dictator is the only member of a class with non-zero voting weight. A large negative weight can give a member veto power to reject any choice.

In proxy voting or **liquid democracy**, participants can delegate their voting rights to others (target voters), thereby transferring their voting weights. When participants do not have the expertise or time to evaluate choices, they may decide to delegate their votes to those whose judgement they trust. Proxy voting is mainly based on reputation and trust, and thus the target voters cannot be anonymous. They do not need to reveal their real identities but should have persistent identities. However, in many situations, to reduce the possibility of collusion or vote selling, target voters should not know who delegated their votes to them. Proxy voting can also make binary weights such that only the members with more than a specific number of proxy votes have a positive voting weight. In **representative democracy**, only a limited number of target voters have non-zero weights and the weights can change only at specific times.

A binary weighting function can distribute ones and zeros based on a random number, in order to select a **random sample of voters** each period. To enforce a selection fee, binary weights can depend on participants' periodic betting amounts, giving zero weight to those who pay less than the selection fee. As explained before, this deters haphazard selection inputs. Betting contests weight votes proportional to the

amount participants spend (bet) so that whoever spends more receives larger voting power. The betting amounts can reflect the participants degree of confidence about their choices.

Proof of work is a binary weighting function that randomly selects one participant based on the betting amounts and a random number each period, so that the participants who spend more are more likely to have the non-zero weight in a period. Technically, this function partitions a line between zero and one into *n* intervals (probabilities), which *n* is the number of participants. The lengths of the intervals are proportional to the participants' spending amounts ($\frac{s_i}{\sum_1^n s_i}$). A uniform random number falls into one interval each period and determines whose selection matters. The Bitcoin blockchain distributes the intervals according to a convex function (e.g. $\frac{{s_i}^2}{\sum_1^n {s_i}^2}$) rather than a linear one. That is because of the economies of scale in mining.

Meritocratic weighting schemes depend on selectors' qualities and merits. Raykar et al. (2010) suggested weighting the evaluations done by experts more heavily. We need to identify voters' expertise to decide how much to trust them and weight their votes (Ertekin, et al., 2013). However, this requires a criterion for expertise. Hill and Ready-Campbell (2011) suggested using past performances to detect and select group of experts in crowd and showed that such strategy increases the crowd's predictive power. Another dimension of merit, beside expertise, is credibility or trustworthiness. Davis and Lin (2011) used the majority rule to resolve conflicts, but they adjusted user's impact based on their trustworthiness, which was deduced from their past inputs. They acknowledged the need to develop better aggregation techniques to handle disagreements. The challenge is to find a criterion that determines expertise or credibility based on endogenous information, not on an external authority to label participants.

A constitution can accumulate information on participants' performances in a state vector, *merit scores*. The performance measurement function, which is part of the state transition function, can adjust the merit scores periodically based on the outcomes of the selection process. The performance measures can relate to the selection activities and their alignment with the majority as in Ertekin et al. (2013). Another approach is to use the participants' success rate in winning suggestions. The suggestions can reflect their understanding of the problem and thus the value of their evaluations. One such measure is the number of

votes each suggestion receives, based on which the proposers can obtain extra votes (merit scores) to cast in upcoming periods. The following clause reflects this approach:

At the end of each selection period, the proposers of all suggestions (not the "Updated Edition") receive extra votes equal to the number of votes given to their suggestions in that period.

An alternative approach is to give extra votes only to the winner, but the above one is more robust when considering close contests. One can make it even more robust by using relative votes instead of absolute votes. It can be relative to the *updated edition* or to the *least-voted choice* as in the following version:

At the end of each selection period, the proposers of all suggestions (not the "Updated Edition") receive extra votes equal to the number of votes given to their suggestions in that period minus the least number of votes given to any choice in that period. Hence, the proposer of the least-voted choice shall not receive any extra votes.

To keep the power diversified and prevent dominance of a few participants, the weighting function can limit the extra voting power in each period and spread it over multiple periods. Otherwise, some proposers can use their extra votes to make their own suggestions win again and win extra votes again. To obtain reliable signals and discover actual innovative potential we need to control for "rich get richer" and for the popularity effects (Chan, et al., 2016). The following clause controls the usage of extra votes:

*A participant's extra votes are automatically applied when he/she votes, increasing his/her voting power. However, only **W** extra votes is used in each selection period so the maximum vote weight is **(W+1)** because every participant has one base vote in each selection period. After voting, the number of extra votes decrease by the number of used votes, and the remaining are carried forward into the next rounds.*

Meritocratic weighting function can be linear, nonlinear or even binary, which gives voting rights to qualified participants. The extra votes resemble proxy voting. The winning proposers can use their extra votes to select better suggestions, which gives extra votes to the proposers of those suggestions. After several iterations (asymptotically), the voting powers may reflect the participants' perceived expertise. Therefore, meritocracy and proxy voting are related to the *Ranking Systems* with *Cumulative Voting* in mechanism theory. In the ranking systems settings, approval voting is the only ranking rule that satisfies ranked independence of alternatives, positive response, and anonymity (Shoham & Leyton-Brown, 2010). Hence, approval voting can be particularly beneficial in meritocracy and proxy voting.

Meritocracy and proxy voting work like a cumulative positive reputation system and makes the selection results less sensitive to registration and double voting. Liu et al. (2014) assumed that each user ID belongs to a unique user because the *Taskcn* reputation system incentivizes the users to use only one identity for all tasks. The reputation system disincentives users against creating new accounts and starting over as newcomers, especially when the expected future profits gained by the reputation exceeds the profit from cheating that ruins reputation (Szabo, 1997). Nevertheless, if reputation is accumulated based on human subjective judgment, registration is required to reliably aggregate user judgments. It does not need registration if reputation is based on a verifiably objective measure, as in the special case of the New York diamond industry, described by Friedman (2000).

While, meritocracy aims to improve the *ability* to find the best version, **plutocracy** can improve the *willingness* to select the best version. In plutocracy, the weighting function depends on the number of shares that selectors hold. Investment by participants is not only a financial source, but also a source of incentive for better selection when participants share ownership of the outcome or have a stake in the outcome. Plutocracy weights their decisions based on their shares. Usually the shareholders' voting power is proportional to their shares, as in public corporations and proof of stake in most blockchains. It can also be binary and give positive weight only to the participants who have enough stakes. In the *Dash* blockchain, a *master-node* with at least 1000 *Dash* can vote.

An advantage of such linear plutocracy is that it is not sensitive to the *non-repudiation* method, whereas democracy is. In linear plutocracy, the amounts of investments determine the voting power and it does not matter if someone controls multiple identities. This obviates the need for non-repudiation and thus plutocracy has investment stage/state instead of registration stage/state. Conversely, democracy requires registration and *proof of individuality* to prevent double voting and *Sybil* attack. In a Sybil attack, individuals make multiple fake identities to influence outcome. In democracy, participants' voting power increases with the number of identities or accounts they control. As a result, democracy is not very effective in the cyberspace with anonymous or pseudonymous digital identities and in *permission-less* blockchains. Some techniques try to prevent double registration by using IP addresses or persistent cookies, but one can easily circumvent them through client-side manipulations. Some techniques make the registration process costly or time-consuming. This turns it into a type of plutocracy, wherein registration is the investment and the number of identities is the share. Essentially, we need to link digital identities to physical entities to detect if different digital signatures belong to the same body. MTurk connects Turkers' IDs to their bank accounts to ensure each person is associated with one ID (Mason & Watts, 2009).

Conversely, democracy is more resistant to *51% attack* or *collusion attack* compared to plutocracy. Such attack in democracy requires the majority of members collude or vote maliciously, hence the term *tyranny of majority*. In plutocracy, a few shareholders can own 51% of the shares, control 100% of the resources, and exploit the other 49% of investments. This concentration of power undermines the impartiality and perceived impartiality of the selection process and disincentives effective participation by minority shareholders. In practice, participation is not perfect and a small group of major shareholders with as little as 30% of the shares might be able to influence all decisions most of the time. It may explain why some top executives receive astronomical salaries at the expense of minority stockholders. Boards of directors often claim that such salaries are necessary to hire high quality managers and are worth the benefits. However, it is usually hard to ascribe such salaries to improvement in firm performance. Most of successful managers do not perform better than average ones in the long term, and the success of firms is mostly due to luck and other factors rather than the effectiveness of the top executives (Kahneman, 2011).

	Investments	Shares	Square Root with Linear Normalization		Root with Degree Normalization	
Investor	***1000 dollars***	***S =***	$W = \sqrt{S}$	**Normalized %**	$W = 0.5 * \sqrt[r]{S}$	Normalized Degree
1	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	**r =**
2	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	**1.62736744**
3	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	
4	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	1 / r =
5	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	0.614
6	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	
7	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	
8	$ 50	**5.0%**	0.224	**8.7%**	**7.9%**	
9	$ 600	**60.00%**	0.775	**30.2%**	**36.5%**	
Total:	**$ 1,000**	100.0%	2.563	100.0%	**100.00%**	**--> Set to 100%**

Table 8-1: Distribution of Voting Rights using Linear and Concave Weighting Functions

One may suggest requiring more than 50% of share-votes in the selection criterion. However, as explained in section 4.1, conservatism gives power to a smaller group to veto and reject others' suggestions. Here, I suggest concave weighting function to control power. One such function is square root so that the weighting function weights the votes proportional to the square root of the shares of the shareholders and then linearly normalizes them to add up to 100% as table 8-1 illustrates. This makes the 51% attack very difficult. To gain 50% of the voting power, an attacker should have about *n / (n+1)* of the shares, where *n* is the number of the other investors with approximately equal shares. Therefore, an investor needs to own more than 95% of the shares to have such power if there are only 19 other investors. However, when the weighting function becomes concave, it becomes sensitive to the registration and non-repudiation method.

To make 51% attack (by one person) impossible, we can use $\frac{1}{2} * \sqrt[r]{S_i}$ as the weighting function (where S_i is the agent *i's* proportion of share) and normalize it via adjusting the degree of the root (*r*) to make the total voting power sum up to 100% as table 8-1 shows. Another way to prevent such attack is to use a hybrid of democracy and linear plutocracy through averaging the percentages resulted from both functions. Notably, hybrid and concave plutocracy schemes need both registration and investment stages.

More egalitarian and inclusive constitutions distribute power more concavely (figure 8-1). Contrariwise, when the distribution of power is more convex, the governance becomes more centralized and extractive. However, democracy is not necessarily always the best, because first, it is sensitive to the non-repudiation method and susceptible to the Sybil attack. Second, it provides limited and low incentives for thoughtful participation, which makes it susceptible to the tyranny of majority. Generally, the

constitutions in the middle balance the powers of majority (low incentives) and minority (high incentives) and thus their selection process is less biased and more reliable.

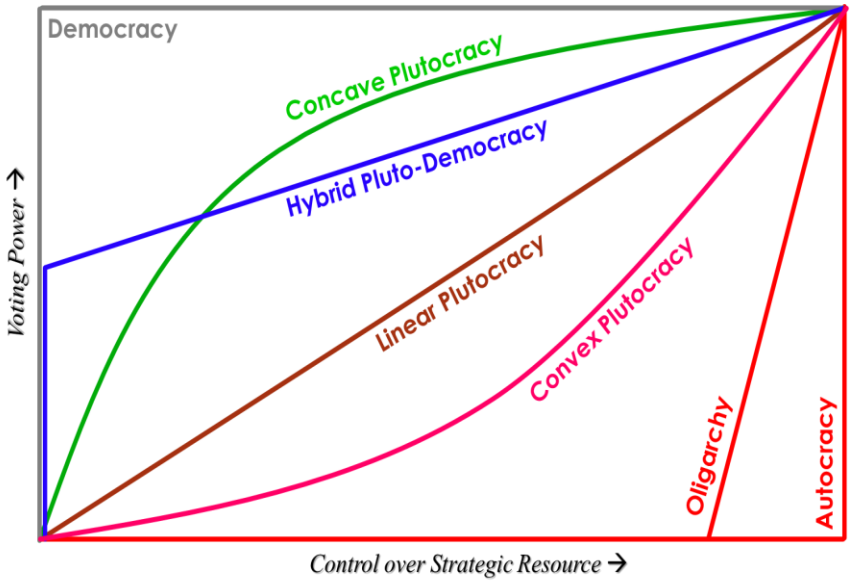


Figure 8-1: Distribution of Power with Respect to the Share of the Strategic Resource

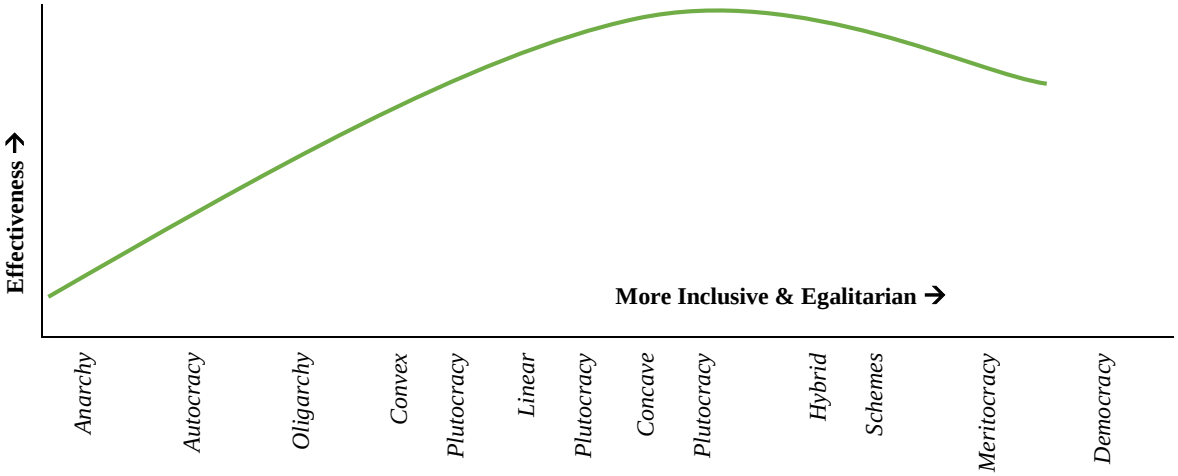


Figure 8-2: Performance of Constitutions with respect to Distribution of Power

Figure 8-2 illustrates the relative effectiveness of different degrees of inclusivity. Anarchy includes no person in collective action, because it results in no collective action. Autocracy is based on one person's decision and oligarchy extends it to a few. Convex plutocracy is vulnerable to collusion attack, which can concentrate power and turn into oligarchy. As convexity decreases, the selection process becomes more inclusive and egalitarian and thus more impartial and trustable. When the function becomes concave, it becomes sensitive to the registration and the non-repudiation method, and the sensitivity increases with concavity, due to more usefulness of multiple identities. Perhaps, linear plutocracy is popular because it is the least vulnerable scheme to collusion attack that is not vulnerable to Sybil attack at all.

To sum up, figure 8-3 provides a taxonomy of governance structures with respect to the distribution of power. As it shows, the market equilibrium price is the only collective decision that is distributed and insensitive to registration, while strategy proof (with rational agents), in the sense that it is resistant to *collusion attack* and *tyranny of the majority*. Appendix J presents a constitution that makes the selections based on market equilibrium prices.

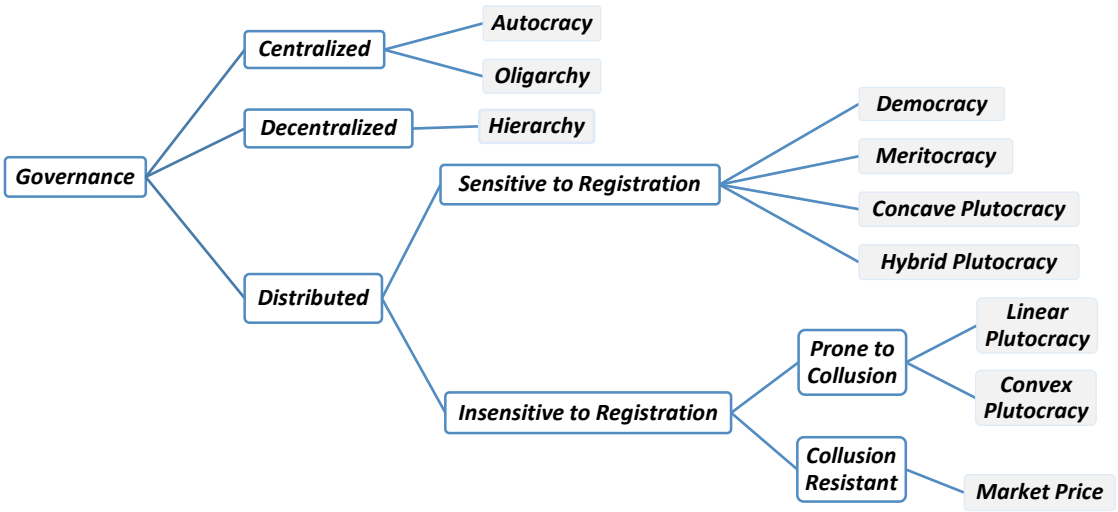


Figure 8-3: Taxonomy of Governance Structures based on the Design Model for Constitutions

9. Constitutional Design

Previous chapters focused on the rules for governing collective design of a solution to a problem. A constitution is also a solution to a problem (how to govern collective design). Therefore, we need some rules to design, modify and amend constitutions. Institutional economists refer to such rules as a second level constitution (Buchanan & Tullock, 1961). The amendment clause in a constitution is a second level constitution whose problem is to improve the primary constitution. The initial edition is the existing primary constitution. Blockchain communities often refer to such amendment rules as the constitution or governance structure of a blockchain. Bitcoin lacks an amendment protocol and is hard to upgrade, thus the developers cannot come to an agreement to resolve the Bitcoin's scalability issue without a fork in the Bitcoin chain.

In crowdsourcing context, Chilton et al. (2016) suggested having the crowd discover the micro tasks in design workflows. Nickerson et al. (2011) considered having the crowd modify the crowdsourcing workflow processes, and they called it *human based genetic programming*. Interestingly, if we regard a constitution as the DNA of an organization, designing it is analogous to genetic engineering.

Generally, to instantiate and execute a constitution for a class of problems, several parameters need to be determined. They include the amount of rewards for winning suggestions, the length of each selection period, the form of the weighting function, the specification of the problem, the initial edition, the population of participants, and other components in the design model (14-tuple). Constitution designers may exogenously (i.e. autocratically) determine those parameters based on theoretical or experimental analysis. The results may or may not be generalizable to other situations or problems. Moreover, constitution designers may set suboptimal values if they have conflict of interest with stakeholders. In general, any exogenous value is a potential source of moral hazard. Alternatively, participants may decide upon some parameter values endogenously through some rules or protocols in an initialization stage after the registration or investment stage. These rules form a second level constitution that specifies parts of the primary constitution. For example, the investors can determine a reward amount by proposing different amounts and then the median of the proposed values (weighted by shares) becomes the reward level. This

approach is endogenous but is still sensitive to incentives of the investors whose inputs specify the value. A better approach might be to use competition and equilibrium points that require fewer parameters. For example, selection fee has one parameter, whereas betting contest makes selection costly through competition. Similarly, suggestion fee has one parameter, whereas advertisement auction has none.

The initial edition of the solution also is a decision variable (included in η_p). It can be decided exogenously or in the initialization stage. It can be the existing solution if one already exists, or it may be set to blank so that suggestions start from nil and the outcome of the first round becomes the initial edition. In blockchains, usually the first block (genesis block) sets the initial balances to zero.

Another parameter is the problem definition and solution specification (included in η_p and I_P), which determine the nature of the solution and the objective of the constitution. Different individuals may represent a problem differently according to their own perspectives (Hong & Page, 2001). Problem specification is a decision that participants can make collectively in the initialization stage. Some problems are decomposable into multiple sub-problems and segments so that different parts of the solution can evolve independently, and then combined to yield a complete solution. It is beneficial when people with different skills can work on different parts and the integration cost is low (Kornberger, 2016). Malone et al. (2017) presented "Contest Webs" as a method for decomposing a problem into multiple parts and merging partial solutions into a big solution, but it has two challenges: identifying good partitioning and constraint management, which means subcomponents from different sources should be mutually compatible. Malone et al. explained that it can largely become automated using algorithmic and mathematical rules.

Representative democracy is different from proxy voting in that there can be specific positions with specific voting powers to be filled for specific periods. The electoral process for filling the representative positions is actually a second level constitution that determines how candidates win positions and what decision rights and voting power each position has. The outcome of this constitution is the set of winning candidates who take the positions and make the population of participants (N) in the constitution.

10. Analytical Model

Here I assume that we have a unidimensional measure for the quality of the solution. The final quality depends on the improvement in each period and the number of periods. The quality added per period depends on the number of suggestions in each period (m), the average quality (improvement) of the suggestions in each period (μ), the diversity of the suggestions in each period (σ^2) and the likelihood of selecting the highest quality in each period. (p). Quality of a suggestion is its contribution to improving the quality of the solution. Since the suggestions are purposeful, we expect $\mu > 0$, whereas in Darwinian evolutions, generally $\mu < 0$, because mutations more often result in defection rather than perfection.

Assuming m suggestions are submitted in a period and their qualities are random variables q_1 , q_2 , … q_m from the probability density function $f(q)$, and the cumulative density function $F(q)$, then the cumulative density function of the maximum quality: $q_{max} = Max\{q_1 , q_2 , \dots q_m \}$ is as follows:

$$F_{max}(q) = P(q > q_{max}) = P(q > q_1 , q_2 , \dots q_m) = P(q > q_1). P(q > q_2) \dots P(q > q_m) = F^{m}(q)$$

Therefore, the probability density function of the maximum quality amongst m qualities is:

$$f_{max}(q) = \frac{dF_{max}(q)}{dq} = \frac{dF^m(q)}{dq} = m.F^{m-1}(q).f(q)$$

Hence the expected level of the maximum quality amongst m (non-negative) qualities is:

$$E[q_{max}] = \int_{-\infty}^{+\infty} q.f_{max}(q).dq = \int_{-\infty}^{+\infty} q.m.F^{m-1}(q).f(q).dq = m.\int_{-\infty}^{+\infty} q.F^{m-1}(q).f(q).dq$$

Sakamoto and Bao (2011), in their results section, provided figures for the distributions of ideas generated by crowd. They showed that the distributions of the qualities - in terms of both practicality and

originality - resemble the normal distribution especially in their upper ends. Thus, assuming normal distribution $N(\mu, \sigma^2)$ for the quality of the suggestions, the above equation becomes:

$$E[q_{max}] = m.\int_{-\infty}^{+\infty} q.\left\{\Phi\left(\frac{q-\mu}{\sigma}\right)\right\}^{m-1}.\varphi\left(\frac{q-\mu}{\sigma}\right).dq = \mu + m.\sigma.\int_{-\infty}^{+\infty} x.\Phi(x)^{m-1}.\varphi(x).dx$$

Wherein we have:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}} \quad \text{and} \quad \Phi(x) = \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}\,dt$$

Here, I define:

$$g(m) = m.\int_{-\infty}^{+\infty}\Phi(x)^{m-1}.\varphi(x).x.dx \qquad (10-1)$$

This function is concave as figure 10-1 illustrates.

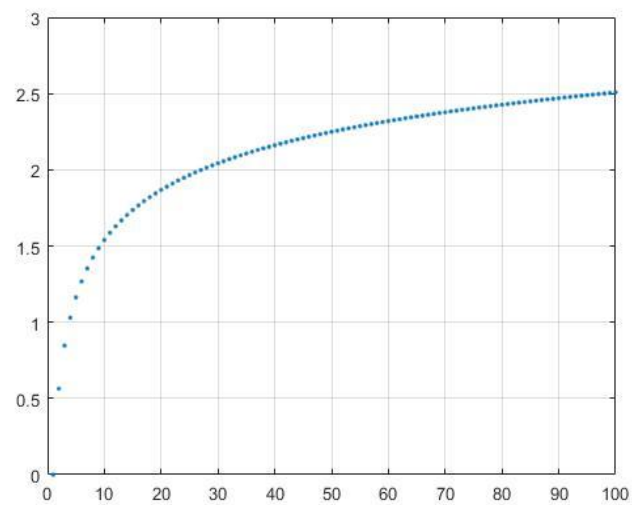


Figure 10-1: Numerical Approximation of Function g(m) for m = 1 to 100 using MATLAB®

Also, I define μ' as the expected quality of the selected suggestion if it is not the best version. Therefore, each round results in this amount of improvement:

$$E[\Delta q] = p.\mu + p.\sigma.g(m) + (1-p).\mu'$$

Approximating μ' with μ simplifies it to the following:

$$E[\Delta q] = \mu + p.\sigma.g(m) \qquad (10-2)$$

Now considering this improvement repeats for z iterations, the objective becomes maximizing:

$$E[Q] = \sum_{0}^{z} E[\Delta q_i] \approx h(z).\{\mu + p.\sigma.g(m)\} \qquad (10-3)$$

Wherein $Q = \sum \Delta q_i$ is the final quality compared to the initial quality, and $h(z)$ is a concave function of the number of iterations reflecting the saturation of quality after the design evolves to higher qualities: ($h(0)=0$, $h(1)=1$, $h'(z) \geq 0$, $h''(z) \leq 0$). A good example would be $h(z) = z^{d}$ where $0 \leq d \leq 1$ is the degree of robustness to saturation. So, $d = 1$ means that the quality of design can improve indefinitely, and $d = 0$ means it can improve once. Therefore, equation 10-3 becomes:

$$E[Q] = z^{d}.\{\mu + p.\sigma.g(m)\} \qquad (10-4)$$

11. Propositions and Relationships

In equation 10-4, the number of iterations (z) is important when d is large and quality of the solution does not saturate fast. Shortening the suggestion or selection periods or increasing the total time (T_Z) can increase z. If the solution is simple and quickly approaches perfection, there is no reason to have many iterations. Relating to the 14-tuple, the state transition function (δ) determines the termination condition and thus the number of rounds.

Figure 11-1 illustrates other parameters and their relationships with the quality (improvement) in one period. The orange boxes are direct antecedents of quality according to equation 10-2, with multiplication presented as moderation. They are the main mediators or moderators for quality. Hereafter, I refer to them as mediators. The green boxes are controllable variables and the blue boxes are the secondary mediators for the primary mediators (mediated mediation). The black arrows represent positive associations and the red arrows represent negative ones. They provide testable propositions that are supported by economic theories as the justificatory knowledge, thereby addressing the fifth and sixth components of an ISDT (Gregor & Jones, 2007). They establish a "proto-theory" (Niederman & March, 2012) that can be tested.

Ren et al. (2014) proposed a brief model composed of three forces that affect the quality of ideas: domain, actors and process. The second force includes motivations and skills of the actors. Higher incentives (willingness) and expertise (ability) can increase the average quality (μ). As equation 10-2 shows, the average quality of suggestions (μ) matters more when m, σ or p are small. Filtration of low quality suggestions and submission costs can also increase average quality but obviously decrease the number (m) and diversity (σ) of the suggestions. Per equation 10-2, when the suggestion variance (not mean) is large, the number of suggestions (m) and the accuracy of selection (p) become more important. Linus Pauling said, "The best way to have a good idea is to have many ideas." (Mulgan, 2006). Longer suggestion periods (T_P) and larger maximum level (M) can increase m, but they can decrease the number of iterations (z) under a fixed time limit. Higher incentives and a larger population can increase participation and m. Relating to the 14-tuple, the state transition function (δ) includes the conditions to end suggestion

and selection periods, and thus it determines the lengths of suggestion and selection periods. Meanwhile, I_V, F and G_V in figure 11-1, are the components of the selection process in the 14-tuple.

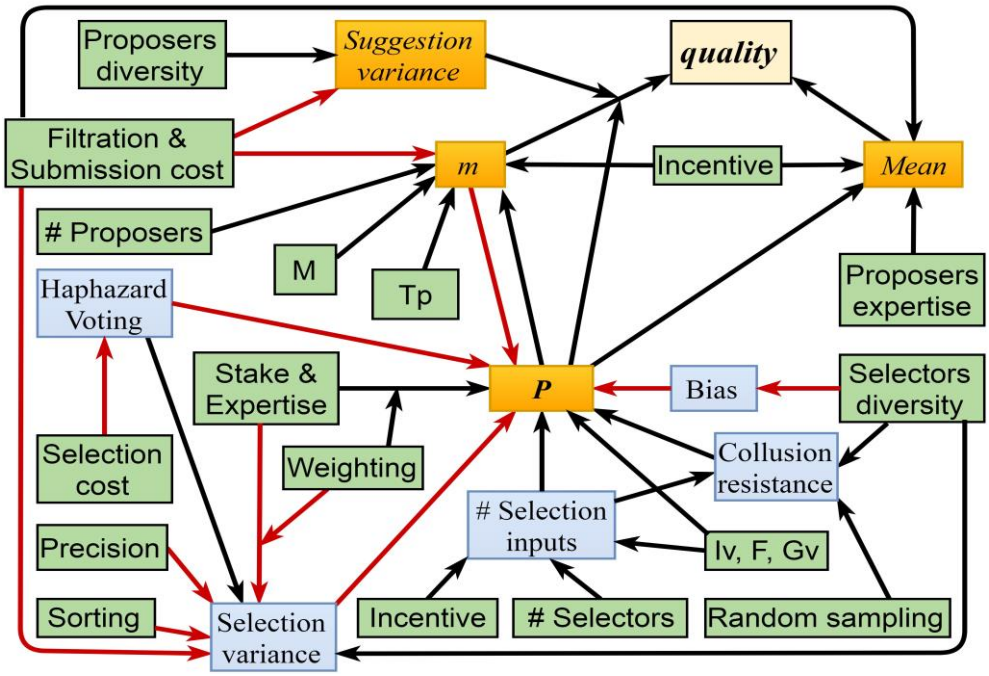


Figure 11-1: Antecedents of Constitution Performance (Quality) and their Relationships

Remarkably, in equation 10-2, function *g(m)* does not have an upper bound, thus with large number of suggestions (e.g. crowd), diversity is more important than the average quality. Diversity of the proposing population can increase the diversity of suggestions. That may partially explain why diverse teams are more effective (Hansen, 2009). In fact, any good model for crowdsourcing contests should take into account the heterogeneity of participants' expertise (Archak & Sundararajan, 2009). In the 14-tuple, the first component (N) determines the population of participants and their heterogeneity, size, expertise, etc. In addition, the filtration condition Φ determines which and whose suggestions can go through.

Chan et al. (2016) linked divergence to novelty and convergence to value. They explained that one way to increase the number and diversity of ideas is to recruit many and more diverse participants. On the other hand, selection and voting should ensure convergence to result in more valuable and feasible ideas. Anonymity of the proposers can also increase diversity, but the winning proposers may prefer to receive

recognition for their contributions. In the 14-tuple, η_V determines what information about the proposers are revealed to the selectors.

The third force (process) that Ren et al. (2014) identified includes the idea selection and evaluation. The value of p reflects the accuracy of the selection process and is associated with its unbiasedness, impartiality and perceived impartiality. Therefore, it affects the participants' incentives and expectations for winning and thus the average quality (μ) and quantity (m) of their contributions. In fact, concerns about the accuracy of output could make participants suspicious about manipulation and undermine their participation (Bonabeau, 2009). Malone et al. (2017) stressed the importance of using a systematic way to measure the quality of proposals (suggestions) in crowdsourcing. Conversely, Clarkson and Alstyne (2007) showed that if the participants believe the system is "fair enough", it will motivate them to produce efficient outcomes and it does not need to be perfect or optimal. In the 14-tuple, the set I_V and functions F and W determine the selection process and the social choice function that aggregates individual preferences.

The number of suggestions can affect the accuracy of selection. Ren et al. (2017) argued that if a crowd is not motivated enough, they submit too many mediocre ideas and make evaluations costlier. Therefore, high incentives can improve the quality of submissions and thereby reduce the effort in the selection stage. However, their argument may not be complete. While higher incentives may (or may not) increase the average quality of suggestions (μ); they certainly increase the number of suggestions (m), which requires more evaluation effort in the selection stage. When the evaluation is costly or time consuming, the large number of suggestions (m) can have an adverse effect on the quality of selection (p).

More selection inputs can improve the accuracy of the selection results (p) directly and indirectly by making collusion harder. Longer selection periods (T_V) and higher incentives for selection can attract more participation and thus more selection inputs. T_V is determined by G_V as described in chapter seven. A larger population of selectors can also increase selection inputs. Involving a more diverse population can decrease bias and the possibility of collusion, while increasing selection variance. An advantage of distributed systems like blockchain is the large and heterogeneous population of selectors who make the selection process more reliable and secure. Some blockchains like Bitcoin also use random sample of selectors to

makes collusion harder. Referring to a random sample of selectors in each iteration can deter collusion without collecting too many selection inputs. Random sample voting makes it harder to manipulate the selection process, because no one knows who is going to vote in the next period (Chaum, 2015). Moreover, a random sample of few voters are more motivated because each vote carries more weight and is more meaningful. Chaum suggested that when there is not enough voting turnout or enough agreement among the random voters, more (random) voters can be invited to vote, up until the outcome becomes statistically conclusive. The main objective of random sampling of voters is to prevent collusion. To this end, we may use voters' demographic information, location, and IP addresses to make randomization more purposive toward maximizing heterogeneity.

Ertekin et al. (2013) explained that when there is a "right" choice or a "ground truth" that the majority of the crowd can detect, the majority vote is a good criterion. They took the majority opinion as the ultimate criterion (true label) and tested two algorithms to approximate the majority opinion using votes of a representative subset of the crowd instead of everybody. They balanced between "exploration" and "exploitation" of the high-quality voters (labellers). Each round the algorithm (i.e. constitution) tries a random subset of the crowd to find the best voters. Then it gives more weight to the votes of those who aligned with the majority. However, when there is no ground truth and the choices are harder to evaluate, we need to rely on people's subjective judgements on the relative *quality* of different choices. Then, compliance with the majority is not necessarily a good criterion especially when the evaluation requires some expertise which may be lacking in most of the crowd. In this type of setting, there is still a "best" choice, but a smaller proportion of the crowd (experts) can detect that choice thereby disqualifying the majority in favour of the experts. A constitution can give heavier weights to the votes of experts or stakeholders who may not align with the majority. Chapter 8 explains different weighting techniques to improve the accuracy of selection. In the 14-tuple, the function *W* determines the weights of the selection inputs and the pool of selectors (positive weight). It can also pick a random sample of selectors by giving zero or positive weight to participants based on a random number.

Generally, when the problems and solutions are more complex and less precise, accurate evaluation requires more expertise (ability) and incentive (willingness). The level of disagreement among the voters increases when the evaluation of choices becomes more difficult (Gillick & Liu, 2010). In such cases, the difference between the evaluation score (e.g. number of votes) of the winning choice and the other choices is small and the variance of the distribution of scores is high. Particularly, the selection variance depends on the gap between the evaluators' knowledge and the knowledge needed to evaluate solutions. The variance among the selectors' opinions is small when there is "knowledge match" (Ren, et al., 2017). For example, 99% of physicists may agree on the answer to an equation, but general population may largely disagree. If the evaluation variance is small enough, we do not need weighting or collecting many selection inputs for accurate selection, but large variances compel more attention to improve the selection accuracy.

Haphazard and unthoughtful selection activities (voting) add noise and increase selection variance thereby reducing the accuracy of selection. Incentives do not deter such activities but can increase them. Excluding outlier selectors can reduce noise. Ertekin et al. (2013) excluded the voters who did not align with the majority. However, restricting a minority in favour of the majority can diminish diversity and lead to a more homogeneous voter population. This can then increase bias or the possibility of collusion. Another way to detect and exclude haphazard voters to use computer generated inapt suggestions (ploys).

An ex-ante deterrence for haphazard selection activities is to make them costly via imposing time or a fee for selection. The selection cost should be smaller than its reward, so not to deter valid voting. An endogenous selection cost is via weighting votes based on the voters' betting amounts on their choices, which reflect their confidence in their choices. However, as chapter eight describes this opens the door for manipulation and 51% attack especially when it can yield large enough profits. The problem is to obtain thoughtful but unbiased evaluations. Generally, selection cost can filter out noise and haphazard inputs, but cannot deter malicious activities and collusion attacks. Excessive costs can deter honest participation for small rewards, but leave out malicious participation, which aims for high profit. Higher fees increase the seriousness of the participants but may not change what they are serious about. The weighting function W, in the 14-tuple can depend on the payment of selection fee to impose the fee.

Another way to reduce the selection variance is to facilitate better selection. Making the problem and the evaluation criterion more precise can reduce the variance and improve accuracy. When the quality is mathematically precise, computers can evaluate them and reach minimum variance (consensus). This is how blockchain protocols can differentiate between valid and invalid blocks and result in consensus. However, the proof of work also imposes cost (bidding) to deter haphazard inputs. Relating to the 14-tuple, the sets η_P and I_P specify the problem and the format of its solutions respectively.

Another facilitation method is (ex-post) filtration of spam and irrelevant suggestions. When there are many suggestions, filtering out low quality ones can help selectors focus on evaluating amenable choices. Spam or irrelevant suggestions can slow down the process and waste evaluation resources. However, such filtration needs a precise formula to classify suggestions based on meeting some minimum criteria. The challenge is to minimize type two error without getting into type one error and undermining diversity. In the 14-tuple, the condition Φ does the filtration. It can depend on some properties of suggestions, some properties the proposers of the suggestions or the payments made by proposers for their suggestions.

A classification formula may specify a population like experts, shareholders or elected participants to act as moderators and perform (ex-post) filtration, resulting in a multi stage selection. The classification formula could also limit proposers to experts, shareholders or representatives in the first place. An ex-ante filtration method is to exclude or suspend the lower performance proposers for a period or indefinitely. Fullerton & McAfee (1999) suggest that a contest should only include the two most skilled participants. However, that likely reduces diversity. A more moderate approach is filtering out only the worst performing participants. The proposer of the least voted suggestion in a period could be considered low performance. The following clause is an example:

After each selection period, the proposer of the least-voted suggestion is suspended for $\boldsymbol{T_E}$ ***time****, unless the least-voted choice is the Updated Edition. A suspended proposer cannot propose a suggestion but still can vote during selection periods and receive rewards for voting for the winning choice.*

Another ex-ante filtration method to deter spam suggestions is to make suggestion costly so that only serious proposers, who have enough confidence in their ideas, submit suggestions. This cost can be imposed via a submission fee. Taylor (1995) found that free entry is not optimal in contests. Assessing suggestions is costly and the proposers know best if their suggestions are worth that cost. The cost of suggestion should be less than the expected benefits, otherwise, it can deter risky and novel suggestions.

Generally, filtration can deter fresh viewpoints that differ from the popular belief (Chan, et al., 2016). Sorting is another facilitation method that can get the spam suggestions out of the way without eliminating any suggestion. Computers can sort choices based on precise measures like time of submission (i.e. chronological). Sorting criterion can be based on proposers' characteristics like number of shares, reputation and past performance. Those who have stake in the outcome are more likely to propose valuable suggestions. The proposers' voting or selection activities can reflect their awareness of and attention to the problem, and their previous suggestions can reflect their expertise and ability. In the 14-tuple, the vector function Ψ sorts out the suggestions.

Here I suggest *advertisement auctions* as a new sorting technique. In each suggestion period, there is an auction for ranking places so that the suggestions of the proposers, who pay more, appear higher and become more visible for selection. The proposers' investments in their suggestions reflect their confidence. They know about the values of their suggestions and this method elicits that information. Like betting, it imposes an endogenously costly competition. The following clause implements advertisement auction:
Proposers can pay to have their suggestions shown in higher places. The suggestions are sorted based on the amounts paid for them, so that the highest paid suggestions are first after the "Updated Edition."

One may also combine multiple layers of sorting criteria, linearly or hierarchically, or let each selector customize them. Sorting can be dynamic and depend on the selection inputs, but this can bias the later inputs due to information cascade. Notably, the order of the presentation of the ideas can introduce position bias, but using randomized order for each selector can reduce the overall bias (Malhotra, 1982).

12. Research Method

This chapter outlines a research method to design and improve e-constitutions based on the proposed design model. I refer to this method as a meta-design method, because its goal is to design a design process (i.e. e-constitution). Table 12-1 illustrates the three levels of the artifacts involved in this research. The levels 1 and 2 resemble levels 1 and 2 described by Purao (2002). This chapter is about level 2 and proposes a systematic methodology to design constitutions.

Level 2	*Research Method*	*Meta-Design Method*	*2nd Level Constitution*
Level 1	*E-Constitution*	*Design Process (Meta-Artifact)*	*1st Level Constitution*
Level 0	*Solution*	*Design Outcome*	*Policy / Decision*

Table 12-1: Three Levels of Design Artifacts and Methods

12.1. Experimentation Procedure

Modeling a constitution as a process allows the use of response surface methodology (RSM) as the meta-design method to improve constitutions. To estimate the effect of multiple factors on the performance of a process, RSM provides many methods such as factorial designs, fractional factorial designs, central composite design, Box-Behnken design and Plackett-Burman design (Khuri & Mukhopadhyay, 2010). These methods try to spread the experiment points over the design space (factorial combinations) to obtain most information with fewest experiments without resulting in multicollinearity. The choice of design depends on the number of factors, levels for each factor and the number of experimental runs (sample size).

Full factorial design is a good choice when the sample size is large, and factors are few. Zhang, et al. (2011) considered four design features (out of five) as factors that affect the performance of collaboration systems. Full factorial design would be a good choice. However, they evaluated the performance of 190 teams using systems with or without the four features, so that some teams had all four design features (treatment) and some teams had none (control).

Despite the large sample size, this study does not detect the effect of any particular factor due to perfect multi-collinearity among the factors. Their claimed findings about the specific features and manipulations are due to their specific assumptions.

On the other hand, when there are many factors and the number of data points is limited, the choice of factor combinations becomes challenging. Even though RSM is sequential by nature (Khuri & Mukhopadhyay, 2010), it needs to test all factors in the early steps particularly in the screening phase (phase 0) in order to detect and filter out unimportant factors. However, constitutions have too many parameters to screen out as factors. Therefore, we need a better approach to detect important factors.

RSM does not consider the nature of the process and the mediational mechanisms through which the factors affect the performance of the process. Therefore, it treats all factors equally and aims for symmetry, rotatability and orthogonality. However, in most practical situations, different factors have different magnitudes of effect and some factor levels may result in much higher or lower performance levels making a subset of data points (low performance) irrelevant because those parameter levels fall out of the region of interest. For example, a full factorial design with 5 factors requires 32 runs. If one level of two factors produce very low performance, 24 out of 32 runs will be out of the region of interest and 8 out of the 32 runs will provide valuable information for estimating the response surface near the optimal point.

This section suggests an experimentation procedure that might be labeled as *Sequential Factorial Design with Alternative Treatments.* This procedure tries to balance between collecting data (exploration) and using data (exploitation) to detect effects and to collect new data efficiently. This procedure does not result in rotatable or symmetric experimental design and does not satisfy any of the alphabetic optimality criteria, rather it collects just enough data to make decisions including the design of further experiments.

Running all treatments simultaneously would allow for assigning subjects to all treatment groups randomly, but would not allow for using some treatment results to design other

treatments. Moreover, in many situations (e.g. MTurk), it may not be practical to recruit a large number of subjects at the same time to assign them to all treatments simultaneously.

While most of RSM methods consider all factors at the same time, the procedure suggested here introduces batches of factors to the model in sequential rounds. This is important because when a process has many parameters (e.g. constitutions), we would need a large sample size and many data points to have enough degrees of freedom for considering all parameters as factors at the same time. To select a subset of parameters as factors, we need to analyze the process in deeper level and consider the nature of the factors and how (mediation) they can affect the performance of the process. For constitutional design, the next section provides the guidelines to select a subset of parameters that are good candidates as important factors. The suggested procedure uses that information for efficient improvement of constitutions. The procedure has the following steps:

0- Collect data on an initial constitution. It can be a crowdsourcing protocol that has already been tried. Alternatively, one may experiment a best guess based on experience. This is treatment *(1)*.[1]

1- Apply the factor selection guidelines on the results and select k parameters as factors $X_1 \ldots X_k$ and form hypotheses about their effects. The number of factors depend on the guidelines, the problem and number of runs the budget allows. A safe choice is two factors when the behavior of the response variable is unknown and unpredictable.

2- Generate a factorial or fractional factorial set of treatments with the k factors. The choice of factorial resolution depends on k and the amount of noise in the response. We need a small statistical power to detect large significant effects and anomalies. The goal is not to test a hypothesis using this set of trials, but rather to detect if any levels of the factors are likely to fall outside the region of interest. With one factor, three additional runs including two runs of treatment A and one replicate of the

[1] By convention, treatments are named as the sequence of their upper level factors (but in lowercase) or *(1)* if all factors are in their lower levels (Myers, et al., 2009).

initial treatment *(1)* can yield enough information. With two factors, three additional runs (*a, b, ab*) and the initial treatment *(1)* can form a 2^2 full factorial that can reveal large effects and anomalies. With three factors, seven additional runs (*a, b, c, ab, ac, bc, abc*) and the initial treatment can make a 2^3 full factorial to yield enough data for detecting large effects. With four factors, seven additional runs (*ab, ac, ad, bc, bd, cd, abcd*) and the initial treatment can form a 2^{4-1} fractional factorial (resolution IV), which has enough degrees of freedom without too many trials.

3- Experiment the factorial treatments preferably simultaneously or in random order. Use random assignment without replacement to assign users to treatments. The inclusion of one or two replicates of the initial treatment or the best performing treatment from the previous round (block) will allow comparison of blocks, and the estimation of pure error (SS_{PE}) and lack of fit.

4- Run stepwise linear regression to detect significant effects. This is the screening phase of RSM, trying to detect the important factors with the available data as the degrees of freedom grow. Stepwise regression also helps to reduce the possible collinearities among predictors.

5- If the regression resulted in significant coefficients for one or more continuous variables, go to the next step. Otherwise, go back to step one using the highest performance treatment as the initial constitution and apply the factor selection guidelines to include more factors or different levels of the same factors if the guidelines suggest already included factors. This will add dimensions (factors) to the model while increasing degrees of freedom and statistical power.

6- Use the highest performance treatment as baseline and estimate the first-order model for continuous factors while keeping discrete factors at their high performance levels. Dropping some low performance points may increase R^2_{adj} of the model and improve the model. That is because of some plausible interaction effects that are not relevant for steepest ascent.

7- Start from the treatment with the highest performance and use the linear model on continuous variables to move in the direction of the steepest ascent (Myers, et al., 2009) while keeping the discrete variables at their high performance levels. The model not only confirms the importance of specific factors but also shows the direction of the steepest ascent for the continuous ones.

8- Regress a second-order model on the direction of the steepest ascent (one-dimensional) and find the optimal constitution with respect to the factors in the model. Experiment this constitution.

9- If the optimal constitution is close enough to the points that formed the first-order model, go to the next step. Otherwise, go back to step 1 using this optimal constitution as the initial constitution and apply the factor selection guidelines to include more factors or different levels of the same factors.

10- Regress a second-order model on the continuous variables (multi-dimensional) and find the stationary point (zero gradient). If there are not enough data for a variable, run more trials around the optimal constitution to have enough sample size for the second-order model.

11- Analyze the eigenvalues and detect the nature of the stationary point. If it is not optimal, move in the proper direction (ridge analysis) and find the optimal point (Myers, et al., 2009). Experiment the optimal constitution.

12- Go back to step 1 using this optimal constitution as the initial constitution.

The main difference between this procedure and the standard RSM is that this procedure uses the factor selection guidelines to look deep into the results and choose factors, whereas RSM looks at the phenomenon as a black box and considers all possible factors. Moreover, most of the RSM designs aim for symmetry, uniform distribution of prediction variance and rotatability, which overlook the fact that different factors have different magnitudes of effects. The suggested procedure is asymmetric toward collecting more data points closer to the region of interest and plausible optimal point, leading to better prediction variance there. This enables better estimation of first order and second order models in the region of interest with a smaller sample size. It takes advantage of the fact that some factors are more impactful and need less statistical power to show significant effects, while some others have smaller effects needing larger statistical power for significance.

12.2. Factor Selection Guidelines

To decide which parameters should be used as factors, we need to look for improvement opportunities. Referring back to equation 10-4, we can improve the performance of a constitution through five mediators: accuracy of selection (1: *p*), number of versions (choices) per period (2: *m*), average quality of suggestions (3: μ), variance of the quality of suggestions (5: σ) and the number of rounds (5: *z*). Theoretically, we should increase the mediators that have larger effects per unit and have more room to increase. That means we should estimate the direction of the gradient of expected quality ($\nabla E[Q]$) times (element-wise) the range of variations of the mediators *(δp, δm, δμ, δσ, δz)'*. Table 12-2 shows the elements of the element-wise multiplication.

$\frac{\partial E(Q)}{\partial p}.\delta p$	$\frac{\partial E(Q)}{\partial m}.\delta m$	$\frac{\partial E(Q)}{\partial \mu}.\delta \mu$	$\frac{\partial E(Q)}{\partial \sigma}.\delta \sigma$	$\frac{\partial E(Q)}{\partial z}.\delta z$
$z^d.\sigma.g(m).\delta p$	$z^d.p.\sigma.g'(m).\delta m$	$z^d.\delta \mu$	$z^d.p.g(m).\delta \sigma$	$d.z^{d-1}.\{\mu + p.\sigma.g(m)\}.\delta z$

Table 12-2: Elements of the Improvement Direction: $\nabla E[Q] * (\delta p, \delta m, \delta \mu, \delta \sigma, \delta z)$

Note:
p=accuracy of selection results; m=number of versions (choices) per period; μ=average quality of suggestions; σ=variance of the quality of suggestions; z=number of rounds; d=saturation effect on improvements (constant).

Therefore, between *p* and *m*, we should focus on increasing *p* if and only if $g(m).\delta p > p.g'^{(m)}.\delta m$, otherwise we aim to increase *m*. Table 12-3 shows all the pairwise comparisons.

	Not m if:	***Not μ if:***	***Not σ if:***	***Not z if:***
Increase *p* if	$g(m).\delta p > p.g'(m).\delta m$	$\sigma.g(m).\delta p > \delta \mu$	$\sigma.\delta p > p.\delta \sigma$	$z.\sigma.g(m).\delta p > d.\{\mu + p.\sigma.g(m)\}.\delta z$
	Increase *m* if	$p.\sigma.g'(m).\delta m > \delta \mu$	$\sigma.g'(m).\delta m > g(m).\delta \sigma$	$z.p.\sigma.g'(m).\delta m > d.\{\mu + p.\sigma.g(m)\}.\delta z$
		Increase *μ* if	$\delta \mu > p.g(m).\delta \sigma$	$z.\delta \mu > d.\{\mu + p.\sigma.g(m)\}.\delta z$
			Increase *σ* if	$z.p.g(m).\delta \sigma > d.\{\mu + p.\sigma.g(m)\}.\delta z$

Table 12-3: Pairwise Comparisons between the Effects of the Mediators

Increasing each mediator requires manipulation of particular constitutional parameters. This section offers some guidelines to detect the parameters that are more relevant and effective in improving each mediator. The guidelines are based on the discussions in the previous chapter, particularly figure 11-1, but the scope of these guidelines is primarily limited to crowdsourcing applications. The guidelines rely on observations (O_B) from a previous trial or run. Accordingly, I assume we have already observed the outputs of an e-constitution (crowdsourcing protocol) that we want to improve. For every mediator, one of the observations is the comments left by the participants. The comments can be used to direct analysis toward specific objective measures.

1- Accuracy of selection (*p*):

Investigating tables 12-2 and 12-3 reveals that an increase in p has the largest impact on the quality, because $g(m)$ and σ are the largest terms in most practical situations. Therefore, we should focus on improving selection accuracy if it has room for improvement. First, we should examine the previous trials and detect the best choices in each period and measure how often they were selected. This gives us a measure of p. If the best choices won every time with relatively high number of votes, this mediator (p) may not have much room for improvement. However, if the best choices have not won as often as expected, we can look for the possible causes for bad selection and the possible remedies. Generally, the causes and remedies can be classified into two categories: too few good votes and too many bad votes. Votes refer to any kind of selection inputs including approval voting, scoring, etc.

Too few accurate votes: When there is not enough participation in selection of an alternative, the results can become unreliable. There are different ways to collect more votes:

1- If possible, expanding the population of selectors can result in more selection inputs.

2- If budget allows, higher selection rewards can incentivize more voting, but if the reward is not linked to performance, it can bring haphazard votes. The selection reward can be based on individual performance (R_V) or group performance (R_G). Expectedly, individual rewards should provide stronger incentives, but in many situations, the only criterion for individual selection

performance is alignment with the majority. In some circumstances there is no objective measure for the group performance, in which case we should rely on external judgment for performance.

3- Reducing selection costs can increase selection activities.

4- If there are accurate votes towards the end of the selection periods, increasing the time for evaluation (T_V) may bring about more accurate votes.

5- When the selection criterion is plurality and multiple good choices compete at each period, they may steal votes from each other so that an inferior choice wins. A quality score that has a limited range (e.g. correct and incorrect) can exacerbate the situation. The traditional approach to cope with this problem is *multi stage elimination*, but a simpler solution is *approval voting*.

Too many erroneous votes: Haphazard and intentionally wrong votes can reduce the accuracy of selection results (p). One may measure the inaccurate voting rate by averaging the periodical selection variances across all selection periods. Periodical selection variance is the variance of selection scores (e.g. number of votes) across different choices in each period. However, when the selection input is voting and we can detect the best version in each period, a better metric for accuracy of selection is to average the percentage of right votes per period across all periods. This metric takes the best choice into account in addition to the spread of the votes. Moreover, it is not very sensitive to the number of voters whereas the variance is. However, it is sensitive to the number of choices. A small number of choices inflates the percentage of votes cast on the best version, and a large number of choices deflates it even if the best version wins with a high margin. Therefore, I define the *accuracy ratio* in a period as the score (number of votes) that the actual best version received over the maximum score that any other version received in that period. If it is more than one, the best version wins. If it is less than one, a wrong choice wins[2]. The magnitude of this metric reflects the margin of success or failure of the selection. This metric is valid for other selection criterion as well as various voting systems. The average of accuracy ratios across all periods reflects the selection accuracy of the constitution.

[2] When it is equal to one, it becomes dependent on the tie-breaking rule in the constitution, but the accuracy ratio still reflects the selection accuracy in any case.

Depending on the situation, different tactics can control for inaccurate votes:

1- If possible, limiting selectors to experts or stakeholders may improve the accuracy of selection.

2- If the accuracy of votes increases during each selection period on average, letting selectors revise their choices may improve the quality of votes.

3- The selection reward can induce haphazard voting if it does not penalize for wrong choices. Particularly, in approval voting, fixed rewards for selecting the winner can incentivize individuals to vote for all choices (or as many as possible) to improve their chance of voting for the winner. One way to cope with this problem is to limit the number of choices a person can select. Limiting it to one choice makes it regular voting. Another way is to adjust the individual rewards based on the number of choices selected or not selected. For example, the reward can be $(m\text{-}V_i).R_W$, wherein m is the total number of choices in a period and V_i is the number of choices voter i selected in that period. R_W is the reward per rejecting a wrong choice while selecting the winner.

4- One can deter bad selection inputs by imposing a selection fee. However, many settings do not allow charging voters and high selection costs may deter accurate selection as well. Using a betting contest is a better mechanism when the right selection fee is hard to determine or can vary from period to period. Accordingly, selectors decide how much to invest on their evaluations and then the weight of their votes and selection rewards are proportional to their bets.

5- If there is bias toward or against a choice (e.g. last choice) in the list, the constitution may present the choices randomly to each participant to distribute and eliminate the bias. If the bias is toward (against) the updated edition from the previous period (i.e. too conservative or too progressive), its selection reward (R_O) may be lower (higher) for voting for that choice when it wins.

6- If a formula can classify those who provided better selection inputs, the constitution can give higher weights to their votes (i.e. meritocracy). If the participants whose suggestions won made better selections in subsequent periods, their votes are worth more. However, too large of voting weights for the winners can bias the results toward specific viewpoints.

7- Meritocracy can also be on the opposite direction so that the constitution bans (gives zero weight to) voting by the worst selectors if they can be classified algorithmically. If the proposers of the least voted (least scored) suggestions made the worst choices in subsequent periods, the constitution may ban them from voting. This can bias results against unpopular viewpoints though.

8- Presenting too many choices to the selectors can result in inaccurate votes. There are different approaches to control for the number of suggestions per period:

   a. Imposing a suggestion fee can filter out poor suggestions if it is possible to charge participants. However, high suggestion costs can deter good suggestions as well. Particularly, different participants have different utility and cost preferences. Hence, a better approach is sorting based on *advertisement auction*.

   b. Sorting can facilitate better selection. Advertisement auction sorts suggestions based on the amounts participants pay to place their suggestions. However, sorting can bias selection in many situations such as prediction voting.

   c. If there are enough selectors, the constitution can show a random subset of the choices to the selectors so that they can focus on evaluating a smaller number of choices.

   d. Banning the proposers of the least voted (least scored) suggestions can deter low quality submissions. However, we need to investigate the results and see if the least voted choices in each period were actually the worst ones. Otherwise, we may lower the quality of suggestions. This can be combined with the item 7 so that the least voted proposers are banned from participation in voting and suggestion periods.

   e. If the best suggestions were submitted in the middle of the suggestion periods, one can limit the number of choices by shortening the suggestion period (T_P) without losing the best suggestions. One might as well impose (or decrease) limit on the number of suggestions per period (M). The limit M makes it uncertain when a suggestion period ends, but the period length T_S makes the number of choices in each period uncertain.

   f. If there is reward for suggestion, decreasing the reward can reduce submissions.

2- Number of suggestions (*m*):

As figure 10-1 illustrates, *g(m)* is concave. So one should weigh the benefits of additional suggestions against the possible decrease in selection accuracy. Increasing the number of suggestions can be effective only if there are too few suggestions and the selection results are safely accurate. In that case, there are some strategies to increase the number of suggestions per period:

1- If possible, expanding the population of proposers can result in more suggestions.

2- If budget allows, higher rewards for winning suggestions (R_P) can incentivize more suggestions. However, the reward should be linked to the quality of suggestions. Otherwise, it could induce spam suggestions. The relative selection scores and winning in selection period can be effective endogenous criterion for the quality of suggestions.

3- If there are submissions towards the end of suggestion periods, increasing the suggestion time (T_P) or the maximum number of suggestions (M) can bring about more suggestions, depending on which one is limiting the number of suggestions.

4- If the constitution bans or suspends suggestions to control number of suggestions, that could reduce number of submissions. Removing or lightening that rule can increase suggestions.

3- Average quality of suggestions (μ):

When there are not enough high quality suggestions, the process stagnates and does not improve the quality of solution. There are several strategies that can be used to improve the quality of suggestions:

1- If possible, expanding the population of proposers to more experts can improve the average quality.

2- Higher rewards for winning suggestions (R_P) may incentivize better suggestions, but it can also increase the number of mediocre suggestions, thereby jeopardizing the selection accuracy.

3- If better suggestions and winning ones are towards the end of suggestion periods, increasing the suggestion time (T_P) or the maximum number of suggestions (M) can bring better suggestions, depending on which one is limiting the number of suggestions.

4- Generally, letting participants edit their suggestions during each period can improve quality.

4- Variance of suggestions (σ):

Here are some strategies to increase the variance of suggestions:

1- If possible, a more diverse population of proposers can increase the suggestions variance.

2- Some studies (Paulus, et al., 2013) found that more original ideas are generated late in a session. Hence, extending suggestion periods by increasing T_P or M may result in more diversity if there are more novel suggestions towards the end.

3- If the constitution bans or suspends suggestions to control number of suggestions, that could reduce diversity. Removing or lightening that rule can increase diversity.

4- Meritocracy could deter diverse suggestions and reducing extra votes can improve diversity.

5- Number of rounds (z):

If the winning versions settled and stayed a specific version in the process, having more rounds may not be beneficial, but rather it might be better to focus on improving suggestions and selections (perhaps by increasing T_V, T_P or M). Nevertheless, if the versions improved until the end of process, more rounds may increase the quality of outcome. To increase the number of rounds we should either increase the total time or shorten the suggestion periods or shorten the selection periods as described below:

1- If it is possible, extending the total process time allows for more rounds.

2- If the best suggestions are in the middle of suggestion periods (not towards the end), decreasing T_P or M (shortening the suggestion periods) can increase the number of rounds in the process.

3- If enough accurate votes are cast in the middle of selection periods (not towards the end), decreasing T_V (shortening selection periods) can increase the number of rounds in the process. Moreover, if selection results are safely accurate and selection scores do not have large variance, that many votes may not be necessary and shorter selection periods can be more efficient.

Parameter-wise Summary:

Table 12-4 summarizes the factor selection guidelines for each constitutional parameter.

Parameters	***Too Low* [H_1: Higher Level is Better]**	***Too High* [H_1: Lower Level is Better]**
T_V	P<1 & Too few good votes & Good votes at the end of T_V	Need more rounds (z) & Safely Accurate Selection
T_P , M	Too few good or novel suggestions & Good or novel suggestions are toward end of T_P	P < 1 & Too many bad votes & Bad suggestions towards end of T_P & Need more rounds (z)
Approval Voting	P < 1 & Too few good votes & Multiple good choices in plurality	P < 1 & Too many bad votes & Over-selection in *Approval Voting*
Revisable Voting	P < 1 & Too many bad votes & Accuracy increases during T_V	Technical limitations
R_P	Too few good suggestions & Enough budget	P < 1 & Too many suggestions & Low budget
R_V, R_G	P<1 & Too few good votes & Enough budget	Low budget & Safely Accurate Selection
Adjusted Selection Reward ([m-V]. R_W)	P < 1 & Too many bad votes & Over-selection in *Approval Voting*	Technical limitations
Ro	P < 1 & Too many bad votes & Bias against the current updated version	P < 1 & Too many bad votes & Bias towards the current updated version
Selection Cost	P < 1 & Too many bad votes & Can charge selectors	P < 1 & Too few good votes & Selection is too costly
Suggestion Cost	P < 1 & Too many bad votes & Too many bad suggestions	Too few suggestions per period
Sorting (vs. Random Order)	P < 1 & Too many bad votes & Too many bad suggestions	P < 1 & Too many bad votes & Bias towards specific choices
Weighting Votes	P < 1 & Too many bad votes & Can classify the best selectors	Low variance of suggestions
Banning Votes	P < 1 & Too many bad votes & Can classify the worst selectors	Low variance of suggestions
Banning Suggestions	P < 1 & Too many bad votes & Too many bad suggestions & The least voted = The worst	Too few suggestions & Low variance of suggestions
Random Subset of Choices	P < 1 & Too many bad votes & Too many bad suggestions	P < 1 & Too few good votes

Table 12-4: Summary of Guidelines for each Constitutional Parameter

Each row in table 12-4 corresponds to one or two relevant constitutional parameters. The contents of the table determine if the level of a parameter is too low or too high. The guidelines suggest increasing the

level when it is too low and decreasing it when it is too high. For example, for T_V , if we see the conditions (symptoms) on the left column (too low), we may hypothesize that the voting period is too short and increasing it can increase performance, but if the conditions on the right column are met, a better hypothesis would be shortening T_V can improve performance. For approval voting, the high level means that the voters can select as many choices as they want, but lower levels of approval voting refer to fewer (more restrictive) numbers of choices a voter can select. Sorting can be systematic (high level) or random (low level). Meeting conditions in the left column suggests that the choices are too random and hypothesizes that presenting in a better order can improve the selection accuracy. Similar arguments apply for the right column.

12.3. Evaluation of Constitutions

Having constitutions in computer code enables us to evaluate and compare them efficiently via online experimentation. This research evaluated different e-constitutions via between-group experiments using subjects recruited from MTurk. MTurk is a web service that enables outsourcing simple tasks to human workers from all over the world (Davis & Lin, 2011). It provides a reasonable and cost-effective platform with diverse participants that are more representative of a real labor market than university students are (Mason & Watts, 2009). To evaluate the performance of e-constitutions, I crowdsourced solving a problem whose solutions have different levels of quality or performance. I defined the problem as follows:

Problem Definition:

Imagine it is June 1, 2013. You have $1000 to invest in stocks, currencies and precious metals like silver. What would be the best trading plan and strategy to make the most profit during the 5 years period until May 31, 2018? The goal is to maximize the total wealth on June 1, 2018. To explore the most profitable assets during these 5 years, you can use historical data found in financial websites such as finance.yahoo.com/most-active and tradingview.com/chart. For simplicity, assume that there is no transaction fee, no commission, and no dividend.

This problem has several desirable properties. First, the performance of solutions can be evaluated objectively without dealing with raters and interrater reliability issues. Particularly, it has virtually no measurement error. Second, the performance has only one dimension (profit) and there is no uncertainty or risk involved. Zhang, et al. (2011) also used a one-dimensional and objectively evaluable quality measure (bug severity) as the performance of the collaboration processes, but their outcome solutions or products (software programs) had several other important quality dimensions, which they simply ignored instead of combining them or at least justifying their choice. Unlike that study, this project uses a specific problem or artifact that actually has one quality measure. However, I acknowledge that in addition to this one-dimensional quality of the outcome solution, a constitution can have other important performance measures such as cost of the process or time of convergence.

Third, this problem does not require forecasting expertise and a non-expert crowd can understand the problem and make significant contributions in a limited time. Decisions involving financial forecasts would require high knowledge and long time to yield meaning variations in performance. Essentially this is an investment planning problem without the forecasting part. Fourth, the quality of the solutions can improve to very large numbers depending on participants' activities and perhaps constitutional parameters. Therefore, it is likely to result in high statistical power.

To start a design process, we need an initial edition. The quality of the initial edition is a pre-test observation and is denoted by $\boldsymbol{O_0}$. In the experiments of this project the initial solution is the following plan:

Initial Edition:

On June 1, 2013: Use 100% of Cash to Buy Dow Jones Industrial Average

On Feb 23, 2015: Sell Dow Jones

On Dec 31, 2015: Use 50% of Cash to Buy IBM & Use 50% of Cash to Buy BTCUSD

On May 31, 2018: Sell IBM & Sell BTCUSD

This plan is feasible, but it is relatively poor and has some obvious rooms for improvement so that most constitutions can result in improvements in the plan. Its performance is about $11k. That means it turns $1000 to about $11,000 in five years. Now we see if the constitutions can govern the crowd to improve this initial edition to an edition with a higher return.

13. Proof of Concept

13.1. Implementation

I used ASP.NET and SQL Server to implement a generic e-constitution in a dynamic website, which currently runs on *https:// Hamed-Constitution.Broad.MSU.edu*. This generic e-constitution has several parameters that cover a small subset of the possible constitutions based on the meta-model. The website has a "Control Panel" page through which the administrator (i.e., experimenter) can access constitutional parameters and adjust them to instantiate different constitutions for collective design. This website enables online experimentation of e-constitutions as treatments. Hereafter, a treatment means an e-constitution with specific parameter values.

This website provides proof by construction and an expository instantiation of the e-constitution design model as the eighth component of ISDT (Gregor & Jones, 2007). Appendices B, C and D present the screenshots of the pages of the website. Appendix F presents the C# code for the generic e-constitution and appendix G illustrates the data structure diagram of the database for the website. The main page in the website enables the participants to login or sign up. To sign up, the users need to accept the consent form agreement and complete the registration form. The administrator can also use this page to log into the website and go to the "Control Panel" page as appendix E shows. In the control panel, the administrator can see the results of completed treatments. When participants sign up, the website directs them to the "Constitution" page, in which they read the constitution as "Instructions" and answer a few questions (the test) about the constitution. If they answer correctly, they can participate in the process and the final survey, if they fail the test they are not allowed to participate. This ensures that all participants understand the basic rules in the constitution. This test has a deadline and the subjects should pass the test before its deadline to participate. After passing the test, if the design process has not started yet, the participants wait until it starts. Once the design process starts, it iterates between two pages: suggestion and selection. Participants can propose modifications during suggestion periods and/or vote during selection periods. At the end, participants are asked to complete the final survey and answer a few questions about the process.

Participants would not receive payment from MTurk if they failed to complete the final survey. During the suggestion or selection periods, participants can click on a button to go to the "Version History" Page and see previous versions in the process. Figure 13-1 illustrates the flow of operations among these pages.

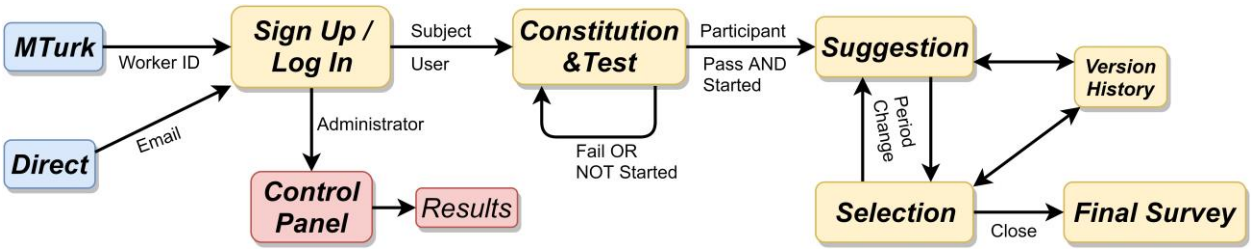


Figure 13-1: Flow of Control among the Webpages in the Website and MTurk

As figure 13-1 shows, users can come from MTurk or directly by registering their emails. This project recruited subjects from MTurk. Appendix H shows the HIT (Human Intelligence Task) page. The HIT page sends the worker ID to the registration form in the website as username. However, outside MTurk, users may sign up directly with their email addresses as username. In such case, a user receives a verification message that includes a link for the user to verify his/her email address. In either case (MTurk or Direct), the website assigns each subject randomly to an upcoming treatment (constitution) based on timing and availability. Figure 13-2 shows the timeline for the events and actions starting from HIT in MTurk.

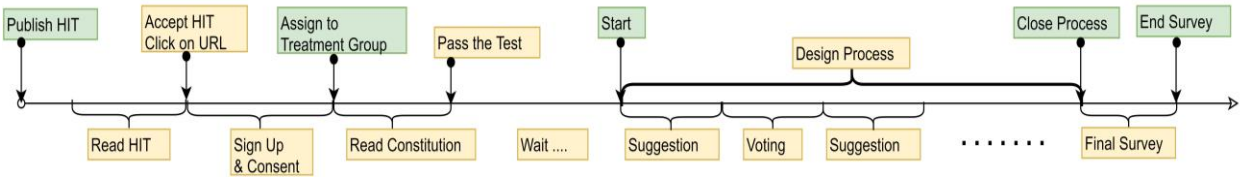


Figure 13-2: Timeline for one Treatment from the HIT in MTurk to the Final Survey

Figure 13-3 shows the sequence of events and actions in different pages. Blue boxes present participant's actions and green boxes present the actions performed by the code or experimenter. At the end, when the participants complete the final survey, the website provides them with a code and the experimenter approves their assignment in MTurk and pays them.

Web Page	Participant	Experimenter / Code
		Post the Treatment ➔ Period = 0 ; Publish HIT
MTurk HIT	Read Task ; Accept HIT ; Click on Link	
		Get Worker ID ; Open First Page (Registration Form)
Log In	Read Consent ; Fill the Form ; Sign Up	
		Assign a Treatment-Group ; Show Constitution
Constitution	Read Constitution ; Answer Questions ; Submit Answers	
		If Correct Prompt the Participant to Wait
	Wait until the Process Starts …	
		Process Starts: Switch to Suggestion Period ➔ Period = 1
Suggestion	Edit the Initial Edition ; Submit Suggestion ; Wait …	
		Switch to Selectin Period ➔ Period = 2
Selection	Read Versions ; Vote ; Wait …	
⋮	⋮	⋮
		Switch to the Final Period ➔ Period = -1
Final Survey	Complete the Final Survey ; Submit Completion Code	
		Approve Work and Pay Balance

Figure 13-3: Sequence of Events and Actions for one Participant in one Treatment

Variable *Period* controls the timing and period change. When *Period* is null, it means that the constitution (treatment) is not operational yet. Once the administrator sets the values of the constitutional parameters, and decides to release it, s/he sets the value of period to zero (*Period* ← *0*). Then at the starting time of the treatment group, *Period* automatically becomes one (*Period* ← *1*) and the first suggestion period begins. Then at specific times the value of period increases one by one. Odd numbers of *Period* are suggestion periods and even numbers (except zero) are voting periods. At the end of the design process (Closing Time), *Period* becomes minus one (*Period* ← *-1*) indicating the survey period. At the end of the survey period, *Period* becomes -11 indicating the end of the experiment for that treatment group.

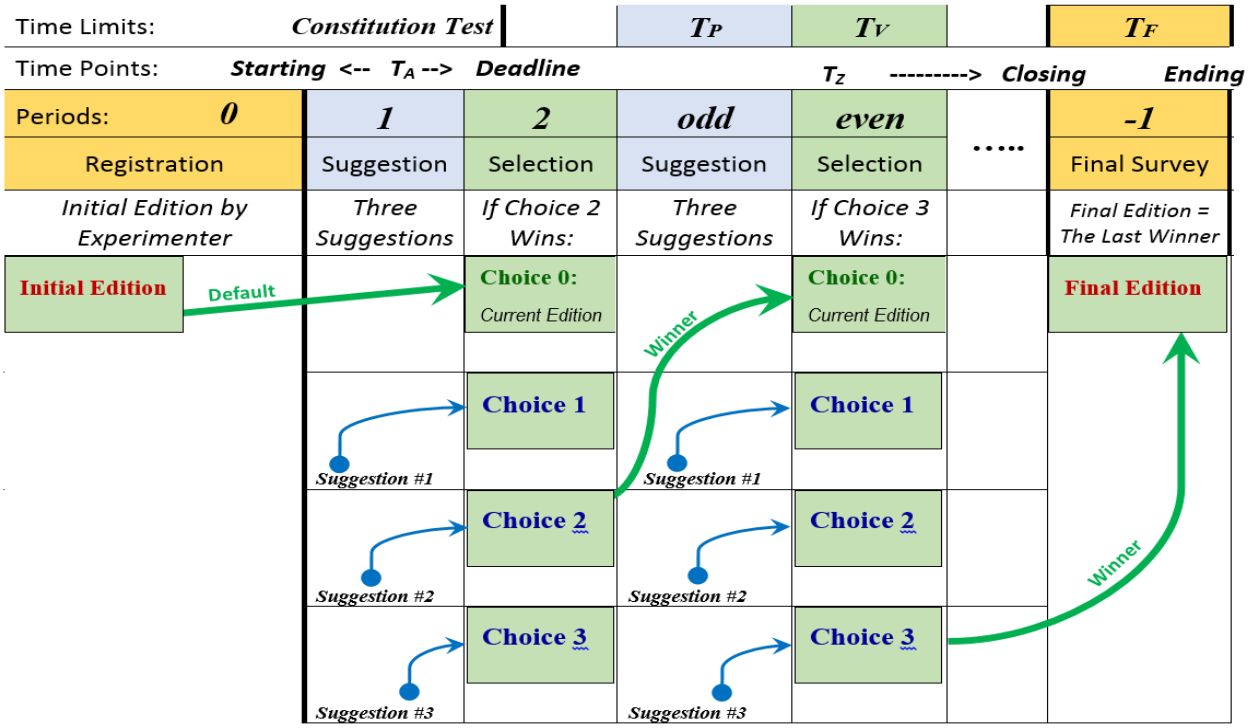


Figure 13-4: Flow of Versions in an Incremental Design Process (Example)

Figure 13-4 illustrates more details about how the website governs the incremental design process in one treatment. At first, the experimenter sets the value of period to *zero* and releases the constitution. When the constitution is available for the participants, they can read it and pass the constitution test by answering a few questions. They can do so until its deadline, which is T_A time after the *Starting* time of the process as figure 13-4 shows. At the *Starting* time, the value of *Period* automatically becomes *one* and the design process starts with the first suggestion period. For technical reasons, the versions of the solution are assigned to the even periods and the initial edition is assigned to period zero. Hence, at the beginning of each suggestion period, the winner from the previous selection period is copied to the next selection period as choice zero and becomes the updated edition. This makes it easier to track the versions across the periods. For each version of the solution, there is a proposer, except for the initial edition, which is ascribed to the "Experimenter" as proposer. If no suggestion wins in

any period, the initial edition is carried from period zero to the last period and the final edition is the same as the initial edition with Experimenter as its proposer.

As figure 13-4 shows, the design process ends at the *Closing* time point, which is T_Z minutes after the *Starting* time point. Then the final survey begins and lasts for T_F minutes ending at the *Ending* time, which is the end of that treatment. At the *Closing* time point, the last winning version is considered as the final edition and the outcome of the treatment. During the final survey period, participants answer a few questions to receive the compensation for participation in the experiment.

13.2. Pilot Experiments

To make sure of the functionality of the website, several pre-test or pilot experiments were conducted. Pilot #0 or pre-pilot was with some students in the Broad college of business at MSU. That pre-pilot verified that the website works properly and is functional. However, based on some students' comments, I added the "Version History" button to enable participants to see the previous versions during the process. Even with that very small sample, two students colluded and voted for each other to win more rewards.

The first pilot with MTurk (pilot #1) was not successful, and there was no effective participation, even though many workers signed up in the website. That was because the HIT in MTurk let everybody click on the link to the website and sign up even without accepting the HIT. As a result, the treatment group became full whereas the worker list was empty. Therefore, I cancelled the HIT and paid the one worker who contacted me. Based on that result, I made a JavaScript code that lets each worker to see the website link only after accepting the HIT. Moreover, the code passes the worker's ID to the website, and the website only takes ID from the JavaScript code in the HIT. This prevents others to sign up and ensures the authenticity of the worker IDs. Another improvement after this pilot was direct login with signup, so that when a participant signs up, they become logged in automatically. This speeds up the process and the participants do not need to enter their username and password unless they sign out. Another improvement

was removing the sign out button from every page, and also remind the workers to leave the page open and stay logged in. This is because the website needs to notify the workers about the start of the experiment and change of periods through desktop notifications and alerts. The website notification is the only practical option because MTurk does not allow asking for workers' emails or any contact info, and the "Notify Worker" Operation in the MTurk API cannot send messages to workers until after the end of the experiment.

Pilot #2 was the first pilot experiment with some results. However, some workers accepted the HIT but signed up too late when the experiment had already begun. This early stage attrition can cause large variation in the number of participants across treatments. While I statistically control for the number of participants, large variations can bring about nonlinearities. Hence, I modified the HIT to tell the workers that they need to sign up before the start of the experiment, so they should sign up immediately or return the HIT. Moreover, every HIT expires 5 minutes before the start of its experiment, so there is enough time to sign up for everybody. Another change is that the new HIT does not allow a worker to participate twice and asks the workers not to share details of the experiment for one month. That is because the participants of later experiments may access that information and then the order of experiments would matter.

There were useful comments received from the participants in pilot #2. Some commented that the voting period was too long and not necessary. The records also showed that most of the participants voted in the first half of the period. Another comment was about letting the workers communicate with each other through some discussion forum or chat room. That might be reasonable but it can introduce many uncontrollable variables. So instead, I modified the system to let the participants include their reasons with their submissions. An additional observation in pilot #2 was that some workers submitted the updated edition without any change. Therefore, I added a restriction that suggestions must be different from the updated edition.

After some modifications, I conducted pilot #3 with an iterative design process, in which the suggestion periods end after one submission. I then discovered that when a suggestion period ends because of a worker's submission, the website did not notify the other participants until they refreshed the page. Many workers commented that they were waiting for the end of suggestion period, even though it had already

ended. To address this problem, I embedded a JavaScript code in the suggestion page to check the value of period every 8 seconds so when the period changes, the code notifies the users to refresh the page. This allows the users to decide when to refresh the page so that if they were working on a suggestion, they could copy and paste that content and not lose it.

Another problem encountered in pilot #3 was the small number of participants (only six workers). Hence, I removed all the restrictions on participation for pilot #4, so anyone could participate in pilot #4. However, this did not increase the number of participants in pilot #4, but rather decreased the quality of participation. Hence, I put back restrictions for pilot #5. In pilot #5, as another attempt to increase the number of participants, I increased the base payoff from $5 to $10. This usually does not improve the quality of participation, but can increase the number of people participating (Mason & Watts, 2009). This resulted in more participants (nine workers) in pilot #5, but not enough. Therefore, for pilot #6, I also included rewards for good suggestion and good voting. This increased the number of subjects to 13, but only six of the workers stayed and finished the experiment.

In pilot #7, I published four HITs with total of 36 assignments which is twice as many workers as I wanted (18 workers). I published all HITs about one hour before the start of the experiment. 34 workers signed up in the experiment and 17 of them participated and stayed until the end and finished the final survey. This result was satisfactory, but there were some comments about the long waiting time. Moreover, most of the workers signed up in the first few minutes and had to wait more than 30 minutes before the experiment. Therefore, in pilot #8 I published the HITs about 40 minutes before the start of experiment instead of one hour. I also modified the interface, made it more user friendly, and asked the workers to explore financial markets while waiting for the experiment. I also increased the reward for winning suggestions from $0.50 to $1.00 because of some comments regarding low incentives. I also shifted the period from the period between May 1, 2013 and April 30, 2018 to the period between June 1, 2013 and May 31, 2018 to be more recent. More importantly, as before, I published twice as many assignments as I needed, but this time I cancelled all HITs once, I got 20 workers. This limits excessive variations in the number of participants.

Pilot #8 resulted in active participation and smooth functioning of the website, but the final solution was not a valid feasible plan. Many participants commented that the task description and instructions were confusing. Therefore, I edited the task description and improved the instructions after consulting with some English speakers. Pilot #9 resulted in a valid and feasible plan, but relatively low quality plan. Close investigation of the results revealed that the votes were not very accurate (low selection accuracy). To incentivize better participation and particularly voting activities, pilot #10 included a group bonus ($2 each participant) for the best performing group. This simulates how shareholders and stakeholders in corporations share the success or failure of the firm. However, this resulted in much worse performance than previous trials. Therefore, I eliminated the group bonus. Particularly, a high performing constitution that does not rely on any external judgement would be more applicable to crowdsourcing situations, because it separates the decision-making and shareholding roles.

After a few additional minor changes in the interface and some tests, I proceeded to conduct the experiments with two treatments at a time. During the first pair of experiments, the server crashed for a few minutes. This made the results not comparable to other results. Therefore, I considered these two experiments as pilots #11 and #12 and analysed their results to improve the website. Accordingly, I added a help page with visual directions for using Yahoo Finance and better contributions. Moreover, that incident showed that experiments might not always go as expected, so I modified the rigid factorial design of standard RSM to a more flexible procedure presented in the previous chapter.

14. Results and Analysis

14.1. Participants

The subjects were recruited from MTurk and were restricted to workers who live in the US and have 97% approval rate with more than 100 assignments completed. Experiments were conducted following the sequential procedure and factor selection guidelines suggested in chapter 12. In such sequential experimentation, some participants from earlier groups may share information to the later groups (*treatment diffusion*). Fortunately, this cannot affect the treatments (rules of the constitutions), the participants' decision rights, or their incentives except different reward levels in different constitutions may result in *resentful demoralization*. If participants find out they were paid less than another group, this may have a negative impact on their performance. This problem is controlled in two ways. First, the rewards are the same across all constitutions except the last two, which have a higher selection reward. Therefore, reward never decreased across treatments. Second, I tried to minimize treatment diffusion, by stating both in the HIT description and in the consent form that the subjects should not share information about the experiment for one month. The program does not let signing up without accepting the consent form as appendix B shows. Appendix A presents the IRB approval document related to informed consent.

Those who gave informed consent and signed up to participate in experiment are counted as *subjects*. When signing up, a subject can declare his/her age, gender, education and whether s/he is a native English speaker. Once signed up, each subject is randomly assigned to a treatment, and can see its constitution as instructions. The number of subjects per treatment is a measure of *treatment delivery*. To participate in experiments, subjects answered a few questions about their constitution. Those who answered correctly were considered *participants* and could propose suggestions and/or vote for versions in their experiment. This is a measure of *treatment receipt.* The program kept record of all previous participants and did not let anyone participate twice. The HIT description informed workers that they cannot participate twice.

The number of participants in each treatment can affect the outcome of treatment. To minimize variation of the number of participants across treatment groups, the program (randomly) assigned workers

to the least populated groups, thereby filling them as evenly as possible. In addition, I statistically controlled for the number of participants per group in the model. The number of participants who stayed in the experiment until the end and participated in the final survey measures the *treatment adherence* for each treatment group. The *measurement attrition* of each treatment group is the percentage of participants who abandoned the experiment at some stage and did not complete the final survey. Appendix D shows the final survey page, which includes three long answer questions and four rating questions about the participant and the experiment as posttest observations (O_B). The scores are between 1 and 10 like other experimental studies on cooperative design (McComb, et al., 2015; Little, et al., 2010). In every treatment, workers who completed the survey received a base compensation of $10 plus a bonus depending on their contributions.

14.2. Dependent Variables

As dependent variables, I measured the quality of the final solutions (plans) in terms of *feasibility*, (actual) *return*, *truthfulness* and *total cost*. Feasibility is a binary variable that indicates if a solution is well formatted and valid and results in an unambiguous return. The return is the primary indicator of the performance of a constitution and is the main response variable that is defined only for feasible solutions. If the final plan is feasible and claims a specific return, truthfulness measures the proximity of the claimed return to the actual return calculated by the experimenter. Truthfulness is not defined for a plan that is not feasible (no actual return) or does not include claim. The *total cost* is the sum of payments for each treatment group. Other dependent variables include *total bonus*, *average bonus per person*, *number of versions per period (m)*, *number of votes per period*, *selection accuracy (p)* and *accuracy ratio* as defined in chapter 12.

The following sections present the treatments experimented based on the research method proposed in chapter 12, starting from the initial constitution or treatment *(1)*. These experiments serve as proof of concept for the model and the methodology. To minimize the time effect, I conducted every set of experiments on a weekday (Monday-Thursday) starting at *7:00pm EST*. Each experiment lasted about one hour ($T_Z = 60$ *mins*) followed by a final survey with 20 minutes time limit ($T_F = 20$ *mins*). I published four HITs per treatment with 9 assignments per HIT about 40 minutes before the start of experiment (*6:20 pm*).

14.3. Treatment (1)

In the initial constitution, the suggestion and selection periods are 4 minutes ($T_P = T_V = 4$) and there is no limit on the number of suggestions ($M = \infty$). The selection process is one vote per person with the plurality rule as the criterion for winning. The program lets participants edit their suggestions during suggestion periods but votes are not revisable in this treatment. Appendix I presents the initial constitution in plain English text. 36 workers signed up after publishing the HITs, but five of them finished the registration process too late after the experiment had started. Hence, 31 subjects were assigned to this treatment. All of them passed the constitution test and qualified as participants. Table 14-1 presents a summary of the results of this treatment.

Metrics	Treatment *(1)*: The initial constitution
Subjects (Treatment delivery)	31
Participants (Treatment receipt)	31 (100%)
Abandoned (Measurement attrition)	9 (29%)
Completed (Treatment adherence)	22 (71%)
% Female	48%
% English speaker	95%
Average age	37.15
Actual return (performance)	**$ 0.78 M**
Claimed return	$ 7056.21 M
Truthfulness	0.01%
Total Cost	$ 273.54
Total bonus	$ 7.95
Average bonus per participant	$ 0.36
Number of rounds	8
Average number of versions per period (m)	*8.0*
Average number of votes per period	*21.5*
Accurate Selections (p)	*4 out of 6 rounds (0.67)*
Average accuracy ratio	*2.31*

Table 14-1: Summary of the Outcomes for Treatment (1)

This treatment resulted in a feasible final plan that would turn $1000 into about $780k. This is the actual return that I calculated based on the historical prices for the transactions in the final plan. The plan came with a claim of about $7 billion return, which indicates a low truthfulness of .01%.

This treatment had eight rounds but the last round was not a full round due to time limit and had only one suggestion. Moreover, I consider the first round as the warm up phase because the participants are learning how the program works. Therefore, the selection accuracy (p) and the metrics averaging across periods (average m, average number of votes per period and average accuracy ratio) are based on the six rounds in the middle, disregarding the first and last rounds. They are the italicized metrics in table 14-1.

As for the selection accuracy, in four out of six rounds, the (actual) best choice won, so a rough estimate of p is 4/6= 66.7%, which has room for improvement. Since the selections are far from accurate and the number of suggestions ($m\text{-}1 = 7$) is reasonable, so increasing the number of suggestions is not a priority. Hence, I analyzed the possible improvements in the selection process. The selection criterion was plurality and most rounds had multiple good choices competing against each other. Hence, I chose *approval voting* as factor A and hypothesized (H_A) that it improves performance. This factor resulted in the following sentence being added to the voting clause of the constitution:

"In each period, you can vote for multiple choices, but not all choices."

No bias toward or against any choice order was detected, but during each voting period, the later votes were more accurate. Therefore, increasing T_V may help, but this would decrease the number of rounds and fewer rounds can decrease quality because the winning version improved until the last period (did not settle). Moreover, some comments mentioned that the voting periods were too long. Hence, instead of increasing T_V, I chose *revisable voting* as factor B and hypothesized (H_B) that it improves performance. Including this factor resulted in the following sentence being added to the voting clause:

"You can change your vote within a voting period."

Therefore, according to step 2 in the suggested procedure, the next treatments are *a*, *b* and *ab*, thereby forming a 2^2 full factorial scheme along with the results of treatment *(1)*.

14.4. Treatments a, b and ab

Treatment *a* uses approval voting and lets participants select multiple choices in each voting period. Treatment *b* allows voters revise their votes in each voting period. Treatment *ab* allows both multiple choices and revising in voting. Everything else is identical to the initial constitution in all treatments.

After publishing the HITs for these treatments, total of 102 workers signed up, but 10 of them finished registration too late after the experiments had started. Hence, 31, 31 and 30 subjects were assigned to treatments *a*, *b* and *ab* respectively and 27, 28 and 29 of them passed the constitution test and qualified as participants. Table 14-2 summarizes the results of these three treatments.

Metrics	Treatment *a*	Treatment *b*	Treatment *ab*
Subjects (Treatment delivery)	31	31	30
Participants (Treatment receipt)	27 (87%)	28 (90%)	29 (97%)
Abandoned (Measurement attrition)	10 (37 %)	8 (29%)	10 (34%)
Completed (Treatment adherence)	17 (63%)	20 (71%)	19 (66%)
Average age	30.29	39.8	32.22
% Female	41%	35%	32%
% English speaker	100%	95%	100%
Actual return (performance)	**$ 36.8 M**	**$0.49 M**	**$ 11.45 M**
Claimed return	$ 79.95 M	$ 0.99 M	$ 12.69 M
Truthfulness	46.06%	49.19%	90.25%
Total Cost	$ 214.49	$ 262.42	$ 238.42
Total bonus	$ 8.74	$ 8.68	$ 8.68
Average bonus per participant	$ .51	$ .43	$ .46
Number of rounds	8	8	8
Average m (versions per period)	*10.83*	*8.67*	*10.83*
Average number of votes per period	*20.33*	*17.17*	*27.83*
Average number of selected choices	*1.4*	*1.0*	*1.88*
Accurate Selections (p)	*6 out of 6 (1.0)*	*3 out of 6 (0.5)*	*3 out of 6 (0.5)*
Average accuracy ratio	*1.71*	*1.2*	*0.9*

Table 14-2: Summary of the Outcomes for Treatments a, b and ab

Treatments *a*, *b* and *ab* resulted in feasible final plans with actual returns of 36.8, 0.49 and 11.45 million dollars and claimed returns of 80, 1 and 12.7 million dollars respectively. Therefore, treatment *a* yielded the highest performance and treatment *ab* yielded the most truthful claim. Each treatment had eight rounds and for the same reasons mentioned under treatment *(1)*, only six middle rounds were considered for estimating the last five metrics (italicized) in table 14-2. In addition to the previous metrics, there is a new metric: *average number of selected choices (per voter per period)*. That is because in approval voting, voters can select different numbers of choices each period and this metric is the average of the number of selected choices across voters and periods. Obviously, it is 1.0 for treatments *(1)* and *b* because they only allow voting for one choice at a time. The average number of selected choices was 1.4 and 1.88 for treatments *a* and *ab* respectively. These levels are surprisingly small, because the average number of choices in each treatment was 10.83 and selecting more choices would increase a voter's chance of voting for the winning choice and winning the selection reward. That means on average voters in treatments *a* and *b* selected 13% and 17% of the choices each period respectively.

To perform a preliminary analysis on the effects of the two factors *a* and *b* on the performance of the constitution, I defined dummy variables X_A and X_B as predictors. $X_A=1$ when selection criterion is approval voting and $X_A=0$ otherwise. Also, $X_B=1$ when votes are revisable and $X_B=0$ otherwise. Moreover, I included variable *n*, the number of participants (treatment receipt) as a control variable. As mentioned before, the performance of each constitution (actual return) is the response variable *y*. Due to the multiplicative and exponential nature of returns, I used the logarithm of the response variable as the dependent variable and fitted the following linear model to the data:

$$Ln(y) = \beta_0 + \beta_n .n + \beta_A . X_A + \beta_B . X_B + \varepsilon$$

Table 14-3 shows the data and the numerical values of the variables and table 14-4 shows the estimates of the model coefficients as well as the relevant statistics.

Treatment	*Performance y ($M)*	*Dependent Variable: Ln(y)*	*Participants n*	*Approval Voting* X_A	*Revisable Voting* X_B
(1)	0.78	-0.245	31	0	0
a	36.83	3.606	27	1	0
b	0.49	-0.722	28	0	1
ab	11.45	2.438	29	1	1

Table 14-3: Levels of the Dependent and Independent Variables in the First Four Treatments

Term	Coefficient	Standard Error	Standardized beta Coefficient	p
Intercept	4.045	0		-
n	-0.138	0	-0.113	-
X_A	3.298	0	0.912	-
X_B	-0.892	0	-0.247	-
$R^2 = 1$, $df = 0$, $N=4$				

Table 14-4: Regression Results for the First Four Treatments

Three predictors and four data points leave no degree of freedom for the residuals and does not allow for any test of significance. However, the number of subjects (*n*) had the least effect on the response variable and only varied between 27 and 31 with a small standard deviation of 1.7, which was less than 6% of its mean. Moreover, *n* is the least significant variable ($p=.872$) using forward stepwise regression. Therefore, for a preliminary test of the significance of factors, I excluded *n* from the model and estimated the coefficients with one degree of freedom for the residuals. Table 14-5 presents the results.

Term	Coefficient	Standard Error	Standardized Beta Coefficient	p
Intercept	-0.072	0.30		0.85
X_A	3.506^*	0.35	0.969	0.06
X_B	-0.822	0.35	-0.227	0.25
$R^2 = 0.991$, $R^2_{adj} = 0.973$, $F=54.2^*$, $df = 1$, $N=4$				

Table 14-5: Regression Results for the First Four Treatments Excluding Variable n
*Note: * ≡ p <0.1*

As table 14-5 shows, approval voting significantly improved the performance of the constitutions, but revisable voting had an insignificant negative effect. These results are preliminary, but can direct the design of the next experiments and collection of additional data. Therefore, the next set of treatments will use approval voting but not revisable voting. In fact, since treatment *a* was the best performing constitution it became the baseline (initial) constitution for the next set of treatments according to step 5 of the proposed procedure. Moreover, I applied the factor selection guidelines on the results of treatment *a* to include new factors for the next treatments.

As for the selection accuracy, throughout all rounds in treatment *a*, the best choice won, giving an estimate of 100% for *p*, which does not have room for improvement. This leads to other improvement opportunities specially regarding the quantity and quality of suggestions. Three comments mentioned that the suggestion periods were too short and they could not make a good plan in that short period. Moreover, out of seven winning suggestions, four were submitted in the last minute and three of them were submitted in the last 10 seconds. Hence, I chose T_P as factor C and increased it from four to six minutes, hypothesizing (H_C) that T_P has a positive effect on performance. I changed the suggestion clause accordingly.

The winning version improved until the last period (did not settle) and thus more rounds may improve the outcome. Since the selection was accurate ($p \approx 1$) for the best performing constitution (treatment *a*), we may increase the number of rounds (z) by shortening the voting periods. Moreover, a couple of participants commented that the voting periods were too long. Hence, I chose T_V as factor D and decreased it from four minutes to three minutes, hypothesizing (H_D) that T_V has a negative effect on performance. I changed the voting clause in the constitution accordingly.

Therefore, based on the suggested procedure, the next treatments are *ac*, *ad* and *acd*, thereby yielding a total of seven points including the previous results.

14.5. Treatments ac, ad and acd

Treatment *ac* had longer suggestion periods ($T_P = 6\ min$) and treatment *ad* had shorter voting periods ($T_V = 3\ min$). Treatment *acd* had longer suggestion periods and shorter voting periods. Everything else was identical to treatment *a* including approval voting.

After publishing the HITs for these treatments, total of 82 workers signed up, but 3 of them did it too late after the experiments had started. Hence, 26, 27 and 26 subjects were assigned to treatments *ac*, *ad* and *acd* respectively and 26, 26 and 25 of them passed the constitution test and qualified as participants. Table 14-6 summarizes the results of these three treatments.

Metrics	Treatment *ac*	Treatment *ad*	Treatment *acd*
Subjects (Treatment delivery)	26	27	26
Participants (Treatment receipt)	26 (100%)	26 (96%)	25 (96%)
Abandoned (Measurement attrition)	6 (23%)	7 (27%)	7 (28%)
Completed (Treatment adherence)	20 (77%)	19 (73%)	18 (72%)
Average age	37.05	37.3	37.89
% Female	20%	40%	50%
% English speaker	90%	100%	100%
Actual return (performance)	**$ 12.14 M**	**$24 M**	**$ 427 M**
Claimed return	$ 2,186 M	No Claim	No Claim
Truthfulness	0.56%	N/A	N/A
Total Cost	$ 248.78	$ 252.96	$ 224.82
Total bonus	$ 7.32	$ 10.8	$ 7.35
Average bonus per participant	$ .37	$ .57	$ .41
Number of rounds	6	9	7
Average m (versions per period)	*12.4*	*16.25*	*11.5*
Average number of votes per period	*24.6*	*37.5*	*27.17*
Average number of selected choices	*1.48*	*1.99*	*1.94*
Accurate Selections (p)	*3 out of 5 (0.6)*	*5 out of 8 (0.63)*	*3 out of 6 (0.5)*
Average accuracy ratio	*1.77*	*1.07*	*1.57*

Table 14-6: Summary of the Outcomes for Treatments ac, ad and acd

Treatments *ac*, *ad* and *acd* resulted in feasible final plans with actual returns of 12.14, 24 and 427 million dollars. Treatment *ac* resulted in a claimed return of about 2.2 billion dollars with a low truthfulness of 0.56%, but the other two plans did not include a claim. Therefore, treatment *acd* resulted in the highest actual return but without a claim. Due to different period lengths, treatments had different numbers of rounds as table 14-6 shows. The last five metrics (italicized) in this table did not include the first round, for the same reason mentioned under treatment *(1)*. However, these treatments had complete last rounds, which were considered in estimations.

The seven data points can give an estimate of the effects of the four factors. To include the additional factors, I defined variables $X_C = T_P$ and $X_D = T_V$ as predictors in the new model:

$$Ln(y) = \beta_0 + \beta_n .n + \beta_A . X_A + \beta_B . X_B + \beta_C . X_C + \beta_D . X_D + \varepsilon$$

Table 14-7 shows the data and table 14-8 shows the estimates of the model coefficients as well as the relevant statistics.

Treatment	*y ($M)*	*Ln(y)*	*n*	X_A	X_B	$T_P = X_C$	$T_V = X_D$
(1)	0.78	-0.245	31	0	0	4	4
a	36.83	3.606	27	1	0	4	4
b	0.49	-0.722	28	0	1	4	4
ab	11.45	2.438	29	1	1	4	4
ac	12.14	2.497	26	1	0	6	4
ad	24	3.178	26	1	0	4	3
acd	427	6.057	25	1	0	6	3

Table 14-7: Levels of the Dependent and Independent Variables in the First Seven Treatments

Term	Coefficient	Standard Error	Standardized Beta Coefficient	p
Intercept	2.48	22.2		0.93
n	0.061	0.77	0.055	0.95
X_A	3.1	2.32	0.653	0.41
X_B	-0.293	1.94	-0.062	0.90
X_C	0.473	1.16	0.199	0.75
X_D	-1.627	2.32	-0.343	0.61
$R^2 = 0.876, R^2_{adj} = 0.259$, $F=1.42$, $df = 1$, $N=7$				

Table 14-8: Regression Results for the First Seven Treatments

This model has one degree of freedom for the residuals, and none of the predictors has a significant effect. After applying stepwise regression (forward or backward), factor A is significant (positive effect) with five degrees of freedom for the residuals. Table 14-9 shows the results of stepwise regression.

Term	Coefficient	Standard Error	Standardized Beta Coefficient	p
Intercept	-0.483	0.94		0.63
X_A	4.038**	1.12	0.851	0.015
$R^2 = 0.724$, $R^2_{adj} = 0.669$, $F=13.1$** , $df = 5$, $N=7$				

Table 14-9: Stepwise Regression Results for the First Seven Treatments
*Note: ** ≡ p <0.05*

This supports hypothesis H_A that approval voting significantly improved the performance of the constitutions, but we need more power to detect smaller effects. Since treatment *acd* resulted in the highest performance, it became the baseline and initial constitution for the next set of treatments. I applied the factor selection guidelines on the results of treatment *acd* to choose new factors for the next treatments.

This treatment resulted in low selection accuracy ($p \approx 0.5$) and average accuracy ratio (1.57). Only in 3 out of 6 rounds, the best version won. The average number of selected choices per period was 1.94 with 11.5 choices per period on average. That means each voter selected about 17% of choices each period on average. Looking into the data reveals that some participants took advantage of the approval voting system and selected almost all choices to get selection rewards without evaluating versions. One way to cope with

this problem is to make the selection reward dependent on the number of selected choices to deter selecting bad choices. Hence as factor E, I changed the selection reward from fixed amount of *$.03* to variable amount of *$.01*(number of choices not selected*) and hypothesized (H_E) that it improves the performance Accordingly, I changed the reward clause in the constitution to the following:

Rewards: *After each voting period, if your suggestion wins, you will receive $1.00 bonus, and if the choice you voted for wins, you will receive a $0.01 bonus for every choice you didn't select in that period. That is $0.01*(Number of Choice – Number of Choices Selected). For example, if there were 10 choices and you voted for 3 of them and one of them wins, then you receive 7 cents for voting in that period.*

I defined the dummy variable X_E to represent the new factor in the model. It equals one when the selection reward depends on the number of selected choices and zero otherwise. Generally, the reward function is a categorical variable when there are multiple options, but here we compare only two options.

Another improvement opportunity is the quantity and quality of suggestions. Again, two participants commented that they could make better plans if the suggestion periods were longer. Moreover, out of seven winning suggestions, three were submitted in the last minute. Hence, I increased factor C further to new level $X_C = T_P = 8$ *minutes* denoted by C' hereafter. I changed the suggestion clause accordingly. This provides more data to test hypothesis H_C. As a result, based on the suggested procedure, the next treatments are *ac'd*, *acde* and *ac'de*, thereby yielding total of ten points including the previous results.

14.6. Treatments ac'd, acde and ac'de

Treatment *ac'd* has longer suggestion periods ($T_P = 8\ min$) and treatment *acde* uses the variable reward function for voting. Treatment *ac'de* applies both changes. Everything else is identical to treatment *acd* from the previous section. After publishing the HITs for these treatments, total of 75 workers signed up, but 9 of them did it too late after the experiments had started. Hence, 23, 22 and 21 subjects were assigned to treatments *ac'd*, *acde* and *ac'de* respectively, and 22, 19 and 20 of them passed the constitution test and qualified as participants. Table 14-10 summarizes the results of these three treatments.

Metrics	Treatment *ac'd*	Treatment *acde*	Treatment *ac'de*
Subjects (Treatment delivery)	23	22	21
Participants (Treatment receipt)	22 (96%)	19 (86%)	20 (95%)
Abandoned (Measurement attrition)	6 (18%)	2 (11%)	5 (25%)
Completed (Treatment adherence)	18 (82%)	17 (89%)	15 (75%)
Average age	37.06	33.53	31.36
% Female	39%	41%	53%
% English speaker	94.4%	100%	100%
Actual return (performance)	**$ 101.8 M**	**$ 1.203 M**	**$ 15.93 M**
Claimed return	$ 121.97 M	$ 1.5 M	$ 22.69 M
Truthfulness	83.46%	80%	70.2%
Total Cost	$ 224.12	$ 229.03	$ 199.30
Total bonus	$ 6.77	$ 20.86	$ 16.08
Average bonus per participant	$ .376	$ 1.227	$ 1.072
Number of rounds	6	7	6
Average m (versions per period)	*9.8*	*13.33*	*9.2*
Average number of votes per period	*31.0*	*30.33*	*19.20*
Average number of selected choices	*2.14*	*2.0*	*1.63*
Accurate Selections (p)	*5 out of 5 (1.0)*	*4 out of 6 (0.67)*	*2 out of 5 (0.4)*
Average accuracy ratio	*1.42*	*1.03*	*0.8*

Table 14-10: Summary of the Outcomes for Treatments ac'd, acde and ac'de

These treatments resulted in feasible final plans with actual returns of about 102, 1.2 and 16 million dollars with truthfulness of between 70% and 80. Treatment *ac'd* resulted in the highest actual return with

the highest truthfulness of 83%. It also had the best selection accuracy ($p \approx 1.0$) with the best choice winning every period. Its average accuracy ratio (1.42) was large compared to the other two (1.03 and 0.8). Apparently, the new reward function was not very effective. It could not even lower the average number of choices selected by the voters (18% and 15%). That is despite the largest total and average bonuses in these treatments. Perhaps this new rule was too complex for subjects. The next section tests that.

14.7. Including Control Variables

Due to different period lengths, the treatments had different numbers of rounds. The last five metrics in the table did not include the first round, but included the last round because the last rounds in these treatments were complete. Now we have ten data points to estimate the effects of five factors and some control variables. Table 14-11 shows the data for the response variable *y*, five predictors (factors) and six control variables. The control variables from left to right are *n, day, average expertise, average comprehension, female percentage* and *total bonus.* As before, *n* is the number of participants per group. *day* is the day of experiment among the four weekdays starting from Monday as day one. The experiments were in blocks of one or three treatments and each block was in a different day of the week. This might have brought about a block effect on the performance.

Treatment	*y ($M)*	***Ln(y)***	X_A	X_B	X_C	X_D	X_E	*n*	*Day*	*Exp*	*Comp*	*F%*	*Bonus*
(1)	0.783	**-0.245**	0	0	4	4	0	31	1	3.227	6.818	0.476	7.95
A	36.83	**3.606**	1	0	4	4	0	27	2	3.882	7.294	0.412	8.74
B	0.486	**-0.722**	0	1	4	4	0	28	2	4.100	7.450	0.316	8.68
ab	11.45	**2.438**	1	1	4	4	0	29	2	4.000	7.000	0.316	8.68
ac	12.14	**2.497**	1	0	6	4	0	26	3	4.700	6.900	0.200	7.32
ad	24	**3.178**	1	0	4	3	0	26	3	3.368	7.368	0.421	10.80
acd	427	**6.057**	1	0	6	3	0	25	3	3.667	8.056	0.500	7.35
ac'd	101.8	**4.623**	1	0	8	3	0	22	4	3.167	5.389	0.389	6.77
acde	1.203	**0.185**	1	0	6	3	1	19	4	4.235	7.412	0.412	20.86
ac'de	15.93	**2.768**	1	0	8	3	1	20	4	3.200	7.000	0.533	16.08

Table 14-11: Levels of the Dependent and Independent Variables in All Ten Treatments

Exp is the average expertise score of participants in each group, and *Comp* is the average comprehension score of participants in each group. I defined *comprehension* of participants as their self-reported score (between one and ten) in response to the following rating question in the final survey:

Was the task description clear and understandable?

[1= Very confusing ... 10= Completely clear and easy to understand]

I also defined *expertise* as the score (1 to 10) a participant gave to the following question in the final survey:

What is your level of expertise in financial investment and stock market?

[1= Never heard of it ... 10 = A professional trader in financial markets]

Appendix D provides screenshots of the final survey including the above questions. *F%* is the percentage of female participants in each group, and *Bonus* is the total bonus earned by participants in each group. Table 14-12 presents the average and standard deviation of the factors and control variables, as well as the Pearson correlations among them.

	Mean	*Std. Dev.*											
Ln(y)	2.44	2.16	*Ln(y)*										
X_A	.80	.42	.71**	A									
X_B	.20	.42	-.38	-.38	B								
X_C	5.40	1.65	.41	.45*	-.45	C							
X_D	3.50	.53	-.45	-.5*	.5	-.64**	D						
X_E	.20	.42	-.23	.25	-.25	.51	-.5	E					
n	25.3	3.89	-.22	-.57*	.43	-.8***	.79***	-.79***	*n*				
Day	2.80	1.03	.38	.66**	-.41	.84***	-.82***	.61*	-.95***	*Day*			
Exp	3.75	.52	-.26	.09	.3	-.25	.46	-.04	.07	-.07	*Exp*		
Comp	7.07	.69	-.07	-.05	.12	-.46	.04	.10	.14	-.22	.34	*Comp*	
F%	.40	.10	.22	.01	-.43	.24	-.57*	.4	-.28	.14	-.77***	.18	*F%*
Bonus	10.32	4.58	-.34	.23	-.19	.26	-.47	.94***	-.7**	.53	.06	.23	.32

Table 14-12: Descriptive Statistics and Correlations among Group-Level Variables
*Note: * ≡ p <0.1 , ** ≡ p <0.05 , *** ≡ p <0.01*

First, I ran the regression on the five factors and control variable *n*, but not other control variables so the results be comparable to previous results. Table 14-13 shows the results of this regression with ten treatments. This model had three degree of freedom for the residuals, and none of the predictors had a significant effect except X_A as before. The insignificance of the *F* statistic indicates lack of fit of the model. However, after applying stepwise regression (backward), factors A and E are significant with seven degrees of freedom for the residuals. Table 14-14 shows the results of stepwise regression. This demonstrates that approval voting significantly improved the performance of the constitutions, and the new reward function significantly reduced it. The control variable *n* was not significant.

Term	Coefficient	Standard Error	Standardized Beta Coefficient	p
Intercept	-5.185	11.424		0.681
n	0.333	0.404	0.598	0.471
X_A	3.597*	1.428	0.701	0.086
X_B	-0.25	1.258	-0.049	0.855
X_C	0.617	0.494	0.47	0.3
X_D	-1.864	1.495	-0.454	0.301
X_E	-2.153	2.076	-0.419	0.376
$R^2 = 0.874$, $R^2_{adj} = 0.621$, $F=3.46$, $df = 3$, $N=10$				

Table 14-13: Regression Results for the Ten Treatments
*Note: * ≡ p <0.1*

Term	Coefficient	Standard Error	Standardized Beta Coefficient	p
Intercept	-0.483	0.97		0.634
X_A	4.216***	1.12	0.821	0.007
X_E	-2.257*	1.12	-0.44	0.084
$R^2 = 0.687$, $R^2_{adj} = 0.598$, $F=7.7$** , $df = 7$, $N=10$				

Table 14-14: Stepwise Regression Results for the Ten Treatments
*Note: * ≡ p <0.1 , ** ≡ p <0.05 , *** ≡ p <0.01*

Now with more data points, I consider other control variables in the model as well:

$$Ln(y) = \beta_0 + \beta_A . X_A + \beta_B . X_B + \beta_C . X_C + \beta_D . X_D + \beta_E . X_E +$$

$$\beta_n .n + \beta_{Day} . Day + \beta_{Exp} . Exp + \beta_{Comp} . Comp + \beta_{F\%} . F\% + \beta_{Bonus} . Bonus + \varepsilon$$

Since the number of parameters is more than the number of data points, I used forward and backward stepwise regression to achieve the best fitness statistics (R^2_{adj}, F and p values). In this process, the total bonus (*Bonus*) showed a significantly negative effect on performance, but it has large collinearity with factor E, so when factor E is included, *Bonus* does not have a significant effect anymore and is eliminated from the model. Table 14-15 shows the coefficient estimates for this model after stepwise regression.

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	-12.16***	1.65		0.002
X_A	3.538***	.284	0.689	0
X_C	0.732***	0.103	0.557	0.002
X_E	-4.486***	0.325	-0.874	0
F%	8.605***	1.18	0.394	0.002
Comp	0.748**	0.197	0.239	0.019
$R^2 = 0.995$, $R^2_{adj} = 0.98$, $F=87.67$*** , $df = 4$, $N=10$				

Table 14-15: Stepwise Regression Results for the Ten Treatments with all Control Variables
*Note: ** ≡ p <0.05 , *** ≡ p <0.01*

The result has four degrees of freedom for the residuals. Factors A and C have significantly positive effects, but factor E has a significantly negative effect. In addition, percentage of female participants had a significantly positive effect on performance. Average comprehension also had a significantly positive effect on performance, but average expertise did not have a significant effect after including average comprehension. However, average expertise was significantly positively associated with comprehension (*standardized beta coefficient* = 1.18, p =.016) when controlling for the percentage of female participants. Thus, average comprehension fully mediates the effect of average expertise on performance.

Interestingly, factor C did not have significant effect before including the control variables as tables 14-13 and 14-14 show. In fact, the control variables *F%* and *Comp* were suppressors such that excluding them from the model suppressed the effect of factors C. After controlling for suppression, the results demonstrated that longer suggestion periods (factor C) improved performance as expected, supporting hypothesis H_C. Nonetheless, the results rejected hypothesis H_E and did not support hypotheses H_B or H_D.

At this point, we can move in the direction of the steepest ascent using the estimated coefficient of factor C for direction ($\partial Lny/\partial X_C = \partial y/(y.\partial X_C) = +.732$). The other factors are set at their best performing levels, so $X_A = 1$ and $X_E = 0$. Factor B was not significant, so we can set $X_B = 0$, which was the level used in seven treatments including the best performing ones. Factor D was not significant either, but because its coefficient was negative, we use X_D = 3 minutes for the steepest ascent. The rest of the procedure follows standard RSM.

14.8. Subject Level Analysis

Throughout the ten treatments, total of 295 subjects signed up in the website (Appendix B) and 253 of them signed up in time and passed the constitution test (Appendix C). Even though the program collected demographic information on all the subjects who signed up, the survey scores are only avilable for the ones who stayed in the experiment and completed final survey. Moreover, the subjects who did not participate in the experiment might have provided false demographic information. Therefore, for subject level analysis, I only used the data from the 185 final participants who completed the final survey. Table 14-16 presents some basic descriptive statistics for the subjects. In the analyses, I used the average age (35.68) for the five participants who did not specify their ages, and the expected value (.3934) of the gender dummy variable for the two participants without gender specification. There was no other missing information.

Variable	Average	Standard Deviation (Sample)	Mininum	Maximum
Bonus Earned	$0.562	0.756	0	$3.45
Age	35.68	9.838	20	70
Comprehension	7.065	2.43	1	10
Expertise	3.762	2.154	1	10
Language	97.3% English Speaker	2.7% Not English Speaker		
Gender	39.34% Female	60.66% Male		

Table 14-16: Subject-Level Descriptive Statistics on the 185 Final Participants from All Treatments

The most relevant performance measure for participants is the bonus each participant earned, because bonus was based on participants' contributions and reflect their performances. This section proposes a structural model to explain the antecendents of bonus as the response variable. Figure 14-1 illustrates the path diagram of the proposed structure with the relevant hypotheses labeling associations.

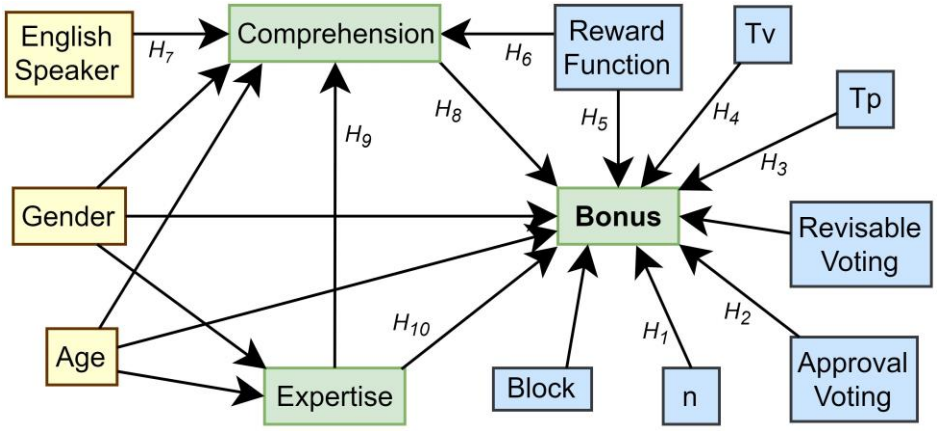


Figure 14-1: Path Diagram for the Full Model with Labeled Associations

This model includes treatment factors and participant characteristics to predict/explain a participant's bonus in a treatment. This model has four control variables among which, gender and age can affect all three endogenous variables as figure 14-1 shows. Block (day of experiment) and voting reversibility are control variables for the response variable bonus. There is no latent variable and all variables are directly measured. While the main purpose of this section is exploration rather than confirmation, the following ten hypotheses justified the proposed structure:

***H_1*:** The number of participants in a treatment group is negatively associated with the bonus for each participant. That is simply because more competition reduces the chance of each participant to win.

***H_2*:** Approval voting is positively associated with the bonus. That is because each participant can select multiple choices and have a higher chance of winning the prediction voting reward.

***H_3* & *H_4*:** Longer suggestion periods (T_P) and longer selection periods (T_V) have negative effects on bonus. That is because longer periods lead to fewer periods thereby fewer winners.

***H_5*:** Using reward function instead of fixed reward (Factor E) is positively associated with bonus. That is because this reward function pays more to incentivize more selective votes.

***H_6*:** Using reward function is negatively associated with how well participants understood the task and instructions (comprehension). That is because it is relatively more complex than a fixed reward.

***H_7*:** English speakers better understand (comprehension) the task and process. That is because the task and instructions are in English.

***H_8*:** Higher comprehension is positively associated with higher bonus. That is because those who understood the task better, could participate better and win more rewards.

***H_9*:** The participants with higher expertise in the subject have better task understanding and comprehension. That is because of their familiarity with the concepts.

***H_{10}*:** The participants with higher expertise in financial markets earn more bonus. That is because of their experience and knowledge in using the tools.

I used SPSS and AMOS software for path analysis on the structural model. Tables 14-17, 14-18 and 14-19 present the path coefficients for the endogenous variables bonus, comprehension and expertise respectively. Figure 14-2 provides a screenshot of the output (standardized) of the AMOS software.

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	1.97	3.001		0.512
N	-0.025	0.085	-0.121	0.771
Approval Voting (X_A)	-0.023	0.192	-0.013	0.904
Revisable Voting (X_B)	-0.006	0.157	-0.003	0.972
$T_P = X_C$	-0.046	0.066	-0.094	0.483
$T_V = X_D$	-0.178	0.191	-0.118	0.354
Reward Function (X_E)	0.62**	0.302	0.311	0.041
Comprehension	0.037*	0.022	0.12	0.091
Expertise	0.073***	0.025	0.209	0.004
Gender	-0.222**	0.109	-0.143	0.044
Age	-0.008	0.005	-0.099	0.152
Block (Day)	-0.061	0.303	-0.08	0.84
$R^2 = 0.272$, $R^2_{adj} = 0.226$, $F=5.882$*** , $df = 173$, $N=185$				

*Table 14-17: Path Coefficients for **Bonus** as the Dependent Variable*
*Note: * ≡ p <0.1 , ** ≡ p <0.05 , *** ≡ p <0.01*

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	7.198***	1.33		0
Reward Function (X_E)	0.254	0.457	0.04	0.579
Expertise	0.304***	0.084	0.27	0
English Speaker	-0.834	1.06	-0.056	0.432
Gender	-0.671*	0.372	-0.135	0.073
Age	-0.007	0.018	-0.028	0.699
$R^2 = 0.119$, $R^2_{adj} = 0.094$, $F=4.833$*** , $df = 179$, $N=185$				

*Table 14-18: Path Coefficients for **Comprehension** as the Dependent Variable*
*Note: * ≡ p <0.1 , *** ≡ p <0.01*

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	3.663***	0.574		0
Gender	-1.398***	0.312	-0.316	0
Age	0.018	0.015	0.083	0.24
$R^2 = 0.102$, $R^2_{adj} = 0.093$, $F=10.386$*** , $df = 182$, $N=185$				

*Table 14-19: Path Coefficients for **Expertise** as the Dependent Variable*
*Note: *** ≡ p <0.01*

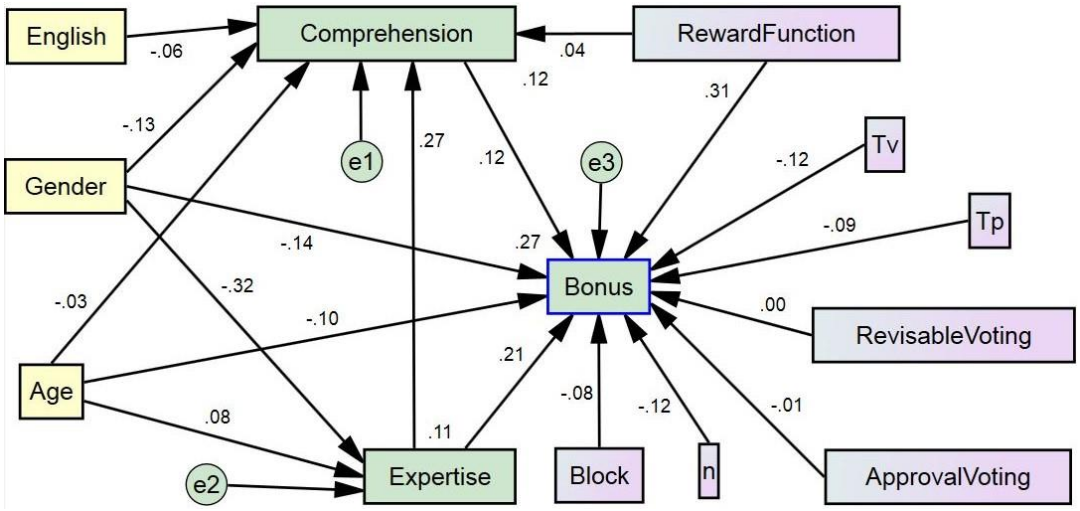


Figure 14-2: Standardized Results of Path Analysis from AMOS for the Full Model

The results support hypotheses H_5 , H_8 , H_9 , H_{10} , but could not reject the null hypothesis for the other hypothesized relationships (H_1 , H_2 , H_3 , H_4 , H_6 and H_7). Moreover, two control variables age and block did not have significant relationships with the relevant endogenous variables, but gender had a significantly negative association with comprehension, expertise and bonus. Gender was coded as zero for male and one for female. Particularly, on average, men ($.69) earned about twice as much as women ($.37). This finding is interesting given that in the previous analysis, the percentage of female participants had a significantly positive effect on performance.

The goodness of fit statistics for this model are relatively poor: CMIN = 1452 ($p<0.01$), NFI = .065, GFI = .456, AGFI =.175, RMR = .735, RMSEA = 0.355 and as for parsimony PNFI = .05. Therefore, to better illustrate and estimate the effects of significant variables, I ran backward elimination and found the path coefficients in a reduced model. Tables 14-20, 14-21 and 14-22 present the significant path coefficients for bonus, comprehension and expertise respectively. Figures 14-3 and 14-4 respectively present screenshots of the standardized and unstandardized outputs of the AMOS software for the reduced model. Figure 14-5 presents another illustration of the path diagram for the reduced model, declaring the significance of the unstandardized coefficients.

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	-0.03	0.178		0.866
Reward Function (X_E)	0.755***	0.129	0.379	0
Comprehension	0.043**	0.021	0.137	0.047
Expertise	0.066***	0.025	0.188	0.008
Gender	-0.225**	0.106	-0.145	0.035
$R^2 = 0.225$, $R^2_{adj} = 0.239$, $F=15.413$*** , $df = 180$, $N=185$				

*Table 14-20: Path Coefficients for **Bonus** as the DV in the Reduced Model*
*Note: ** ≡ p <0.05 , *** ≡ p <0.01*

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	6.179***	0.417		0
Expertise	0.306***	0.083	0.271	0
Gender	-0.671**	0.366	-0.134	0.068
$R^2 = 0.114$, $R^2_{adj} = 0.104$, $F=11.696$*** , $df = 182$, $N=185$				

*Table 14-21: Path Coefficients for **Comprehension** as the DV in the Reduced Model*
*Note: ** ≡ p <0.05 , *** ≡ p <0.01*

Term	Coefficient	Standard Error	Standardized Beta Coefficient	P
Intercept	4.3***	0.194		0
Gender	-1.367***	0.311	-0.309	0
$R^2 = 0.096$, $R^2_{adj} = 0.091$, $F=19.339$*** , $df = 183$, $N=185$				

*Table 14-22: Path Coefficients for **Expertise** as the DV in the Reduced Model*
*Note: *** ≡ p <0.01*

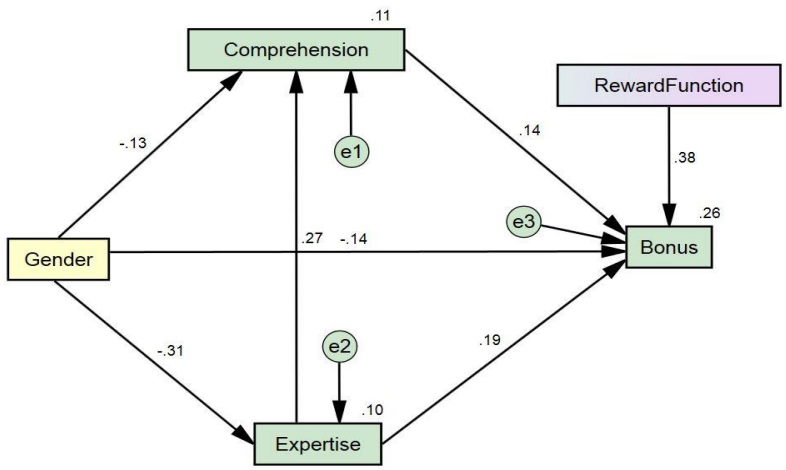


Figure 14-3: Standardized Results of Path Analysis from AMOS for the Reduced Model

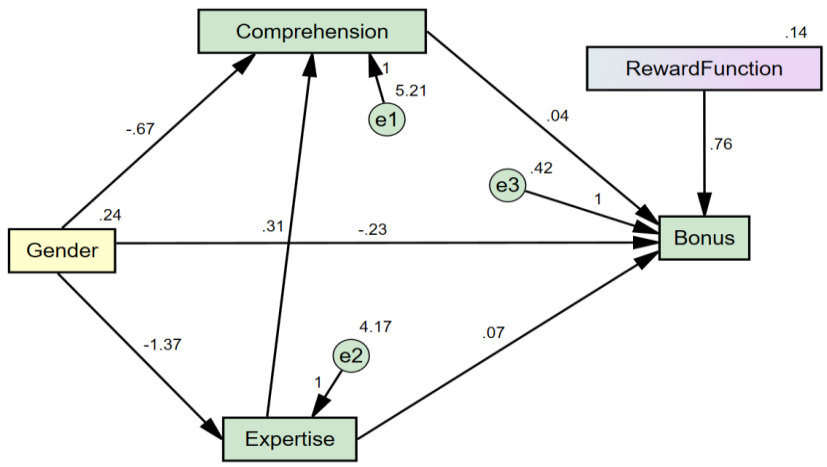


Figure 14-4: Unstandardized Results of Path Analysis from AMOS for the Reduced Model

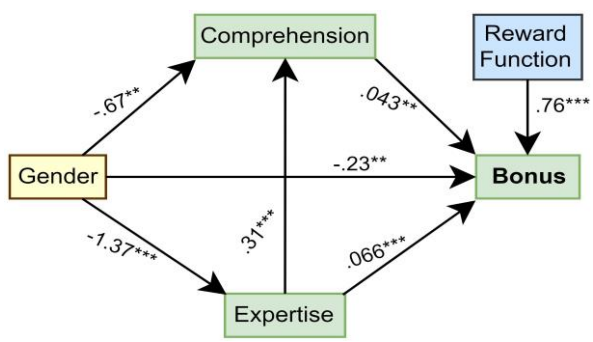


Figure 14-5: Path Diagram with the Unstandardized Path Coefficients for the Reduced Model
*Note: ** ≡ p <0.05 , *** ≡ p <0.01*

The reduced model has far better goodness of fit statistics: CMIN is 1.325 and is not significant anymore (*p=0.723*). For the reduced model NFI = .986, GFI = .997, AGFI =.986, RMR = .01 and RMSEA is less than .01, and regarding parsimony PNFI = .296 for the reduced model. Furthermore, the reduced model better revealed the three mediational relationships. First, comprehension partially mediated the effect of expertise on bonus. The direct effect of expertise on bonus was .066 (*p=.008*) and the indirect effect was .013, for which the Sobel test statistic is 1.79 with *p* value of .07, indicating a significant indirect effect. Therefore, the total effect of expertise on bonus was .079 of which, 84% was direct and 16% was through increasing comprehension. Participants with more expertise earned more rewards partially because they could better understand the task and process, but mostly (five times more) because of other reasons above and beyond the resulting improvement in comprehension.

Second, expertise partially mediated the association between gender and comprehension. Gender had a negative significant direct effect of amount -.67 (*p=.068*) on comprehension. Its indirect effect on comprehension was -.42, for which the Sobel test statistic is -2.82 with *p* value of .005, implying a significant mediation. Therefore, the total effect of gender on comprehension was -1.09, of which 62% was direct and 39% was because of a negative association with expertise.

Third, in addition to a significantly negative direct effect of -.225 (*p=.035*), gender also had indirect negative effects on bonus through comprehension (-.029), expertise (-.09) and the comprehension resulted from expertise (-.02). The numbers in parentheses are the estimated indirect effects of the three mediational paths. The Sobel test statistic for the first mediator (comprehension) is -1.37 with *p* value of .17, indicating a non-significant path. For the second mediator (expertise), it is -2.26 with *p* value of .024, indicating a significant mediation. Hence, expertise partially mediated the relationship between gender and bonus. The total effect of gender on bonus was -.36, of which 62% was direct and 38% was indirect. While expertise can partially explain the gender wage gap, one should look for other plausible mediators such as attention span and care for money, in order to explain the direct effect. Also, the effect of gender deserves further study as the percentage of female participants had a significantly positive effect on performance, yet when examined on a subject level, females received a lower bonus.

15. Discussion and Limitations

The subject level results demonstrated the role of expertise and gender in the success of participants to earn bonus. As figure 14-3 illustrates, the standardized direct effect of expertise on bonus (.19) was the largest direct effect among gender, comprehension and expertise (not factor E). Expertise also had an indirect effect through improving comprehension. The total standardized effect of expertise on bonus (.225) was comparable to the total standardized effect of gender on bonus (-.232) in the subject level analysis.

One important finding was the significantly negative effect of gender on bonus. It is consistent with the findings of Niederle and Vesterlund (2011) that women's performance is lower than men's in tournaments. Since the e-constitution code could not discriminate against or in favour of any group, this result demonstrated the possibility of a significant gender pay gap under zero possibility of discrimination. On the other hand, in group level, the percentage of female participants had a significantly positive effect on the group performance as table 14-15 shows. This might be because female participants cared more about teamwork and group performance rather than winning rewards, while perhaps male participants were more selfish and aggressive in competing inside groups to win more rewards. Another possible explanation is that most groups had disproportionately more male participants than female (40% female on average) and thus groups with more female participants were more balanced and diversified, which led to generating and selecting more novel and superior ideas. As explained in chapter 11, Hanson (2009) found diverse teams are more effective. All in all, this is an area that requires further research.

While gender had large significant effects on all endogenous variables in the subject level analysis, age did not have any significant effect on any variable even on expertise. That is surprising because older people often have more experience and knowledge. One possibility is that older people with expertise in financial markets are too busy to work in MTurk. Another surprising finding is that, being English speaker did not have a significant effect on comprehension. Perhaps, that is because the registration form asked if the subject is "Native English speaker" and those non-native English speakers were fluent in English and could understand the task and instructions well enough to earn a good bonus. Factor E did not have significant

effect on comprehension either, rejecting the hypothesis that the new reward function made the task harder to understand. The significant effect of factor E on bonus was predictable, because the new reward function paid more on average. However, one cannot interpret this bonus increase as performance improvement. Particularly, the two treatments with the new reward function did not yield better outcomes.

The insignificant effect of approval voting on bonus is because most of participants did not select too many choices to take advantage of approval voting scheme. As tables 14-2, 14-6 and 14-10 show, the average number of choices selected was between one and two in most treatments with approval voting as opposed to one in other treatments. Two other unexpectedly insignificant factors in subject level are C (T_P) and D (T_V), which determined the number of rounds. Shorter periods resulted in more rounds and more rounds should have brought more winning rewards. However, as table 14-20 shows, the effects while negative were not large enough to be significant. That seems to be due to the small variation in the number of rounds, which was obscured by larger significant effects of other variables such as gender and expertise. The same argument goes for the insignificant effects of the number of participants in each group (n).

It is worth noting that the effects of variables on individual bonuses are different from their effects on the performance of group. Expertise had direct and indirect positive effects (through comprehension) on bonus in the subject level analysis, but average expertise had only indirect positive effect (through average comprehension) on group performance. Factor E had opposite effects on individual bonus and group performance. Approval voting improved group performance, but did not increase individual rewards. Same thing holds for longer suggestion periods (T_P). In fact, the total bonuses paid per treatment reflect the cost of treatment as another response. While a higher group performance implies a more effective constitution, a larger bonus means a more costly and less efficient constitution. This makes approval voting and longer suggestion periods even more desirable, because they improved performance while not increasing costs.

The group level results demonstrated the effectiveness of approval voting in improving the performance of constitution. That is probably because approval voting does not fall under the impossibility theorems (Maniquet & Mangin, 2011), while plurality voting violates the independence of irrelevant alternatives (Nisan, et al., 2007). Remarkably, approval voting resulted in superior performance despite rewarding the

voters based on prediction voting. The predication voting incentive schemes reward voters for voting for the winner choice and approval voting allows voters to vote for as many choices as they want. This should incentivize rational voters to vote for every choice to maximize their chance of winning. However, it did not happen as often as utility theory would predict. Voters rarely voted for more than three choices and they were more selective than approval voting allowed. However, a mechanism that deters selection of inferior choices may improve the performance of approval voting even further. A reward function that penalized voting for wrong choices (Factor E) was one such attempt, but the results did not support its hypothesized effect. One may try to ascribe this observation to the complexity of the function and hypothesize that participants did not understand it, but the subject level results rejected the hypothesized negative relationship between comprehension and factor E. Another possible explanation is that participants were selecting too few choices anyway, and this incentive could not reduce their number of selected choices any further, but rather distracted them. Perhaps a less distractive mechanism is to limit the number of choices each participant can select. Accordingly, one may define variable X_A as a continuous variable between zero and one, indicating the proportion of choices a voter can select in each period.

Another finding in the group level analysis was that longer suggestion periods (T_P) improved the performance of the design process. In fact, several participants commented in the final survey that they needed more time to edit and create a new suggestion. That can also be ascribed to the relative reward amounts for suggestion and voting. The reward for the winning suggestion ($1) was much larger than the reward for right selection ($.03), therefore participants wanted to spend more time on making suggestions rather than voting. As future research, one can investigate the interaction effects between relative rewards and period lengths. The significance of control variables in the group level analysis has an important theoretical implication. It shows that missing a confounding variable in the model can suppress some significant effects and give rise to misleading results. Hence, it is imperative to consider every plausible cause and test their effects. When the degrees of freedom are very limited (as in this project), stepwise regression can be considerably helpful in detecting significant confounding variables.

Generally, the results demonstrated that the characteristics of the design process (factors A, C and E) as well as the designers (expertise, gender) can have significant effects on the quality of the outcome. However, others (Brooks, 2010; Deng & Ji, 2018) proclaimed that it is not the design process, but rather it is the designer that drives the quality of design. This research showed that the process often complements the designer(s) particularly when there are many designers (collective design) involved. For example, meritocratic schemes give more weights to the inputs from more expert designers. Further research can analyze the relative impacts of design process and designer characteristics on performance.

The group level results also demonstrated how the suggested procedure could improve the performance of constitution from $.78M to $427M in just ten experiments. The main advantage of the suggested procedure was that the effects of factors C and E were estimated mostly at the more effective levels of factors A and B, because that is the region of interest and thus the effects of interest. Standard RSM would require more than 24 runs to detect the same effects and reach the same conclusion. Only six out of those 24 runs would be irreversible approval voting (region of interest). Other trials would estimate effects outside the region of interest that do not help in moving toward the optimal point. Conversely, the suggested procedure yielded 7 out of ten runs in that region of interest as tables 14-11 shows. That is because standard RSM emphasizes on symmetry and orthogonality, whereas the suggested procedure distorts the experimental design towards more experiments inside or closer to the region of interest. Moreover, RSM sees the process as a black box and considers all factors equally as important while the suggested procedure applies specific guidelines to utilize deeper analysis of the mediators to decide which factors can better improve the performance of the process. This allows for using information from prior experiments to design later ones by introducing more factors to the model. The suggested procedure is particularly useful when there are many parameters and we do not know which ones are more important to change as factors. One may modify the guidelines to incorporate other constitutional parameters and features and apply the suggested procedure to improve crowdsourcing protocols, blockchain protocols or other types of constitutions. Practitioners may develop similar guidelines for other processes and applications so that they can use the suggested procedure. To this end, they need to specify parameters for their process as in RSM.

One limitation of the suggested procedure is that it is not an algorithm, but rather depends on a set of guidelines which require subjective judgements. As a result, different practitioners or experimenters may go in different paths to improve constitutions. Particularly, the outputs of experiments are subject to noise and sample variance, thus even RSM can result in different paths for the improvement of constitutions in different trials with different subjects.

Another limitation of this study is the small number (10) of experiments. A larger sample size (in terms of treatments) could give more statistical power and enable us to estimate the effect of less impactful factors such as revisable voting (factor B). Another limitation is the lack of replicated runs and estimate of pure error (SS_{PE}). To test the lack of fit, one should estimate the pure error by replicating some experiments. Testing lack of fit is particularly important in forming and using second order model and the final steps of RSM. One limitation of the subject level analysis is that there was one item to measure each of the two constructs comprehension and expertise. It might pose a threat to construct validity. A better approach would include two or three questions for each construct. Another threat to construct validity is that the expertise was measured in the final survey after the experiment, so the performance of the subject in the experiment could affect his/her perception of his expertise in the financial markets. A better approach would be to include multiple questions about subject's expertise in the registration page.

Perhaps the most important limitation of the results is external validity. The groups were very small (about 20 subjects), the duration of the process was one hour and the prediction voting was the only incentive for better selection. Other incentive mechanisms such as group reward could result in different outcome and different effects for factors, perhaps, an even stronger effect for approval voting.

Moreover, the design problem (retroactive investment plans) was unique. While this particular problem brought many advantages regarding internal validity and reliability, it poses a threat to external validity and ecological validity because it does not have a real world application. For future research, one may try to use the suggested procedure to solve a more practical problem such as designing a financial portfolio. However, other problems may need more elaborate evaluation for the quality of solution.

16. Conclusions

This project started with the problem of collective design in the cyberspace. While it is akin to the problem of collective action, information technology has brought about new aspects to this problem. On one hand, computers while being impartial and more trustworthy, can execute rules faster than humans. On the other hand, the anonymity of users poses new challenges to collective decision making in cyberspace.

To better understand various aspects of this problem, I introduced the concept of e-constitution and developed a design model including a structured representation and formalization for it. This design model as a meta-artifact decomposes an e-constitution into 14 components and parameters including a state transition function and a weighting function. This model implies that most of collective action structures are special cases of the same phenomenon but mostly with different weighing functions. Moreover, this model highlights the importance of the convexity or concavity of the weighting function in distributing and controlling power.

The constitution model provides a framework to design various governance structures such as crowdsourcing schemes, collective intelligence systems, blockchain protocols, DAOs and organizational bylaws. The main problem in most of these situations is how to aggregate individual inputs into one collective output. In this regard, the social choice function and weighing function play crucial roles.

This model defines the objective of a constitution as collective design of a solution for a problem. The solution can be a decision, a policy, a product or a financial plan as in the experiments of this project. This objective provides quantifiable performance measures for constitutions, namely quality, cost and speed. The value of the outcome solution reflects the quality of the constitution. This project discussed the antecedents of a constitution's quality and presented testable propositions and relationships about its mediators. They imply several hypotheses regarding constitutional parameters and features. One may consider this model a nascent design science theory for e-constitutions.

Another contribution of this project is proposing a method to improve the performance of constitutions systematically. It includes a procedure based on RSM and a set of guidelines to introduce factors to the

model in batches instead of simultaneous factorial design. The procedure improves a constitution more efficiently by utilizing information obtained in the prior experiments to design the next experiments. To this end, the constitution design model and its parameterization was essential in development of guidelines.

As a proof of concept, I used the method and design model to improve a simple collective design process through online experiments on MTurk. The results demonstrated the utility and effectiveness of the model and method in improving the performance of constitutions. An important finding is that some constitutional characteristics such as the voting scheme (social choice function) can significantly affect the quality of the outcome design. Hence, the quality of outcome design can be used as an indicator for the performance of constitutions. This enables us to evaluate and compare constitutions objectively free from any value assumption. The ultimate goal is to move towards an optimal constitution.

APPENDICES

APPENDIX A: IRB Application Determination Letter

MICHIGAN STATE
UNIVERSITY

Initial IRB Application Determination

Flexibility Exempt

SEE SPECIAL EXCLUSIONS BELOW

(e.g. no federal funding)

June 5, 2017

To: Joshua Introne
jintrone@msu.edu

Re: **IRB# X17-649eD**
Category: Exempt 8
Exempt Determination Date: June 5, 2017

Title: A Systematic Methodology to Build, Evaluate and Enhance "Distributed Convergent Design Process" based on Digitally Enforceable Constitutional Architectures.

Your project has been determined to be exempt under the Flexibility Initiative Exemption Category 8.

Exemption Category: This project has qualified for the Flexibility Initiative Exemption Category 8: Research involving benign interventions in conjunction with the collection of data from an adult subject through verbal or written responses (including data entry) or video recording if the subject prospectively agrees to the intervention and data collection and at least one of the following criteria is met:

(A) The information obtained is recorded in such a manner that human subjects cannot be identified directly or through identifiers linked to the subjects; or
(B) Any disclosure of the human subjects' responses outside the research would not reasonably place the subjects at risk of criminal or civil liability or be damaging to the subjects' financial standing, employability, educational advancement, or reputation.

See HRPP Manual 8-8-B, Exemption Category 8, for the full text of Exemption Category 8 (http://hrpp.msu.edu/msu-hrpp-manual-section-8-8-b).

Exclusions: To continue to qualify for Exemption Category 8, the project must not include:

- Federal funding or federal training grants
- FDA regulated
- Sponsor or other contractual restrictions
- Clinical interventions (including clinical behavioral interventions)
- Prisoners as subjects
- Receipt of an NIH issued certificate of confidentiality to protect identifiable research data
- Be a project for which MSU serves as the IRB of record
- Children as research subjects

If any of the above criteria become applicable to a project determined exempt under this flexibility initiative, the IRB office must be promptly notified prior to implementation of the criteria and the project must be reviewed and approved in accordance with the appropriate review level (e.g. expedited, full board).

Principal Investigator Responsibilities: The Principal Investigator assumes the responsibilities for the protection of human subjects in this project as outlined in HRPP Manual Section 8-1, Exemptions (http://hrpp.msu.edu/msu-hrpp-manual-section-8-1).

Renewals: Exempt projects do not need to be renewed.

Revisions: In general, investigators are not required to submit changes to the IRB once a research study is designated as exempt as long as those changes do not affect the exempt category or criteria for exempt determination (changing from exempt status to expedited or full review, changing exempt category) or that may substantially change the focus of the research study such as a change in hypothesis or study design. See HRPP Manual Section 8-1, Exemptions, for examples.

Office of Regulatory Affairs
Human Research Protection Program

Biomedical & Health Institutional Review Board (BIRB)

Community Research Institutional Review Board (CRIRB)

Social Science Behavioral/Education Institutional Review Board (SIRB)

Olds Hall
408 West Circle Drive, #207
East Lansing, MI 48824
(517) 355-2180
Fax: (517) 432-4503
Email: irb@ora.msu.edu
www.hrpp.msu.edu

Figure A-1: IRB Approval and Application Determination Letter

APPENDIX B: Registration Page Including the Consent Form

Your Username is your Worker ID.

WorkerID: AWORKER1

Password: ••••••••

Please log in if you already registered.

Log In

Edit Profile

Registration Form:

Nickname: *

Education:

Age:

Gender: Male Female

Native English speaker

Research disclosure (agreement):

The purpose of this research is to study how people can collaborate to incrementally design and improve a solution (plan) to a problem (task).
After signing up, you will see a green box containing the instructions and the rules of the game. To participate, you must read the instructions and answer a few questions correctly. During the experiment, you can see versions of the plan in yellow boxes and you can modify or evaluate them depending on the period. You must complete a short survey at the end, to receive a completion code and get paid. Otherwise, you will not be eligible for payment.

Moreover:
* You must be at least 18 years old to participate in this research.
* Your participation in this research is voluntary and you may choose not to participate at all or to stop participating at any time or refuse to participate in certain procedures or answer certain questions.
* To participate, you should choose a name (nickname) for yourself. You are not required to reveal your real identity, but you may choose to do so for recognition on your contributions.
* No sensitive information will be collected, and all your information except your nickname will be kept confidential.
* **For one month, you should not share information about this experiment.**

If you have any questions or concerns, you can contact the experimenter: *Mr. Hamed Khaledi* or *The MSU Institutional Review Board*. [IRB approval number: IRB# X17-649eD]
By clicking on the button below, you indicate your informed consent to participate in this study.

I accept the above agreement.

Sign Up

Worker ID : *

Password: *

Confirm password: *

Designed and Developed by *Hamed Khaledi* © 2017

Figure B-3: Screenshot of the Main Page in the Web Application

APPENDIX C: Screenshots of the Webpages for the Experiment

Congratulations! You passed the test. Please use internet to explore markets in the last five years!

The game will start at *3:00 PM Eastern Daylight Time*

in about *3 hours , 1 minute*

Please read the instructions and answer the questions. Keep this page open until the end of the experiment.

Instructions:

Problem: Imagine it is June 1, 2013 and you have $1000 to invest in stocks, currencies and precious metals like silver. What would be the best trading plan and strategy to maximize your profit over the 5-year period from June 1, 2013 until May 31, 2018? **Your only goal is to reach maximum wealth on June 1, 2018.**
You can use historical data on financial websites such as Finance.Yahoo.com and CoinRanking.com. Assume there is no transaction fee, no commission, and no dividend.

Process: The game begins with an initial plan. Then you improve the plan in several editing rounds. Each round consists of a suggestion period followed by a voting period that results in an Updated Edition for the next round. This game iterates for about one hour. Then you must complete a short survey to receive $10 plus your rewards.

Suggestion: In each suggestion period, you can submit one suggestion and modify it if you want to. You are not required to submit a suggestion every round. Suggestions should be different from the Updated Edition. Each suggestion period ends after 8 minutes if at least one suggestion is submitted. Otherwise, the program waits until one submission.

Voting: Each voting period begins with a minimum of two choices and lasts 3 minutes. The choices are the Updated Edition and the submitted suggestions. In each period, you can vote for multiple choices, but not all choices.
You cannot vote for your own edition. You are not required to vote every round. You cannot change your vote within a voting period.

Winning: After each voting period, the most voted version wins and becomes the Updated Edition for the next period. Then the winning version and the winner's name are announced. Other suggestions and all votes remain anonymous. Then, the next suggestion period begins.

Rewards: After each voting period, if your suggestion wins, you will receive $1.00 bonus, and if the choice you voted for wins, you will receive $0.03 bonus.

How many suggestions (at most) can you submit per period?	1	Suggestions
What is the minimum number of choices in each voting period?	2	Versions
How long is each voting period?	3	Minutes
How long is the game?	60	Minutes
The game begins with:	Voting ○ Suggestion ◉ Survey ○	

Correct! **Please keep this page open!** **Click here for tips and visual directions!**

Figure C-1: Constitution Page for Participants to Read the Constitution as Instructions

Hi Worker 2 ! Your Balance = $ 0.03 Versions History

Make sure your submission is valid and feasible and has accurate numbers!

Waiting for suggestions until *2:19 PM Eastern Daylight Time*

in about *5 minutes , 10 seconds*

You submitted the following plan:

This is a plan suggested by Worker2:

On June 1, 2013: .....
.....

_____

REASONS:

On June 1, 2013 .....
.....

Thanks! Edit Submission

Instructions:

Visual Directions

Problem: Imagine it is June 1, 2013 and you have $1000 to invest in stocks, currencies and precious metals like silver. What would be the best trading plan and strategy to maximize your profit over the 5-year period from June 1, 2013 until May 31, 2018? **Your only goal is to reach maximum wealth on June 1, 2018.** You can use historical data on financial websites such as Finance.Yahoo.com and CoinRanking.com. Assume there is no transaction fee, no commission, and no dividend.

Process: The game begins with an initial plan. Then you improve the plan in several editing rounds. Each round consists of a suggestion period followed by a voting period that results in an Updated Edition for the next round. This game iterates for about one hour. Then you must complete a short survey to receive $10 plus your rewards.

Suggestion: In each suggestion period, you can submit one suggestion and modify it if you want to. You are not required to submit a suggestion every round. Suggestions should be different from the Updated Edition. Each suggestion period ends after 8 minutes if at least one suggestion is submitted. Otherwise, the program waits until one submission.

Voting: Each voting period begins with a minimum of two choices and lasts 3 minutes. The choices are the Updated Edition and the submitted suggestions. In each period, you can vote for multiple choices, but not all choices.
You cannot vote for your own edition. You are not required to vote every round. You cannot change your vote within a voting period.

Winning: After each voting period, the most voted version wins and becomes the Updated Edition for the next period. Then the winning version and the winner's name are announced. Other suggestions and all votes remain anonymous.
Then, the next suggestion period begins.

Rewards: After each voting period, if your suggestion wins, you will receive $1.00 bonus, and if the choice you voted for wins, you will receive $0.03 bonus.

Figure C-2: Suggestion Page for Participants to Submit their Suggestion

Hi Worker 2 ! Your Balance = $ 0.03 Versions History

Please evaluate all plans and vote! Do not vote for an invalid or inferior plan!

This voting poll closes at *2:22 PM Eastern Daylight Time*

in about *2 minutes , 41 seconds*

Select one or multiple of the best valid plans:

☑ *Initial Edition:*

On June 1, 2013: Use 100% of Cash to Buy Dow Jones Industrial Average
On Feb 23, 2015: Sell Dow Jones
On Dec 31, 2015: Use 50% of Cash to Buy IBM & Use 50% of Cash to Buy BTCUSD
On May 31, 2018: Sell IBM & Sell BTCUSD

REASONS:
Dow Jones was 14,701 on June 1, 2013, and became 17,758 on July 1, 2015. That turns $1k to $1208
IBM was $138 on Dec 31, 2015, and became $141 on May 31, 2018. That turns $604 (= 50% of $1208) to $617.
Bitcoin was $434 On Dec 31, 2015, and became $7463 on May 31, 2018. That turns $604 to $10386.
Hence, this plan would turn $1k to $11,003 in five years!

☑ *Suggestion 1:*

This is a plan suggested by Worker2:

On June 1, 2013: .....
.....

REASONS:

On June 1, 2013 .....
.....

☐ *Suggestion 2:*

This is a plan suggested by Worker1:

On June 1, 2013: .....
.....

REASONS:

On June 1, 2013 .....
.....

Submit **Voting for *2 choices.***

Instructions:

Visual Directions

Problem: Imagine it is June 1, 2013 and you have $1000 to invest in stocks, currencies and precious metals like silver. What would be the best trading plan and strategy to maximize your profit over the 5-year period from June 1, 2013 until May 31, 2018? **Your only goal is to reach maximum wealth on June 1, 2018.** You can use historical data on financial websites such as Finance.Yahoo.com and CoinRanking.com. Assume there is no transaction fee, no commission, and no dividend.

Figure C-3: Voting Page for Participants to See Versions and Submit their Selection

Your ID: AWORKER2

Return

This page refreshes at *2:22 PM Eastern Daylight Time*

in about *1 minute , 22 seconds*

Previous Versions:

Versions in Round 2:

Initial Edition:

On June 1, 2013: Use 100% of Cash to Buy Dow Jones Industrial Average
On Feb 23, 2015: Sell Dow Jones
On Dec 31, 2015: Use 50% of Cash to Buy IBM & Use 50% of Cash to Buy BTCUSD
On May 31, 2018: Sell IBM & Sell BTCUSD

REASONS:
Dow Jones was 14,701 on June 1, 2013, and became 17,758 on July 1, 2015. That turns $1k to $1208
IBM was $138 on Dec 31, 2015, and became $141 on May 31, 2018. That turns $604 (= 50% of $1208) to $617.
Bitcoin was $434 On Dec 31, 2015, and became $7463 on May 31, 2018. That turns $604 to $10386.
Hence, this plan would turn $1k to $11,003 in five years!

> *Suggestion 1:*

This is a plan suggested by Worker2:

On **June 1, 2013: .....**
.....

REASONS:

On **June 1, 2013 .....**
.....

> *Suggestion 2:*

This is a plan suggested by Worker1:

On June 1, 2013: .....
.....

REASONS:

On June 1, 2013 .....
.....

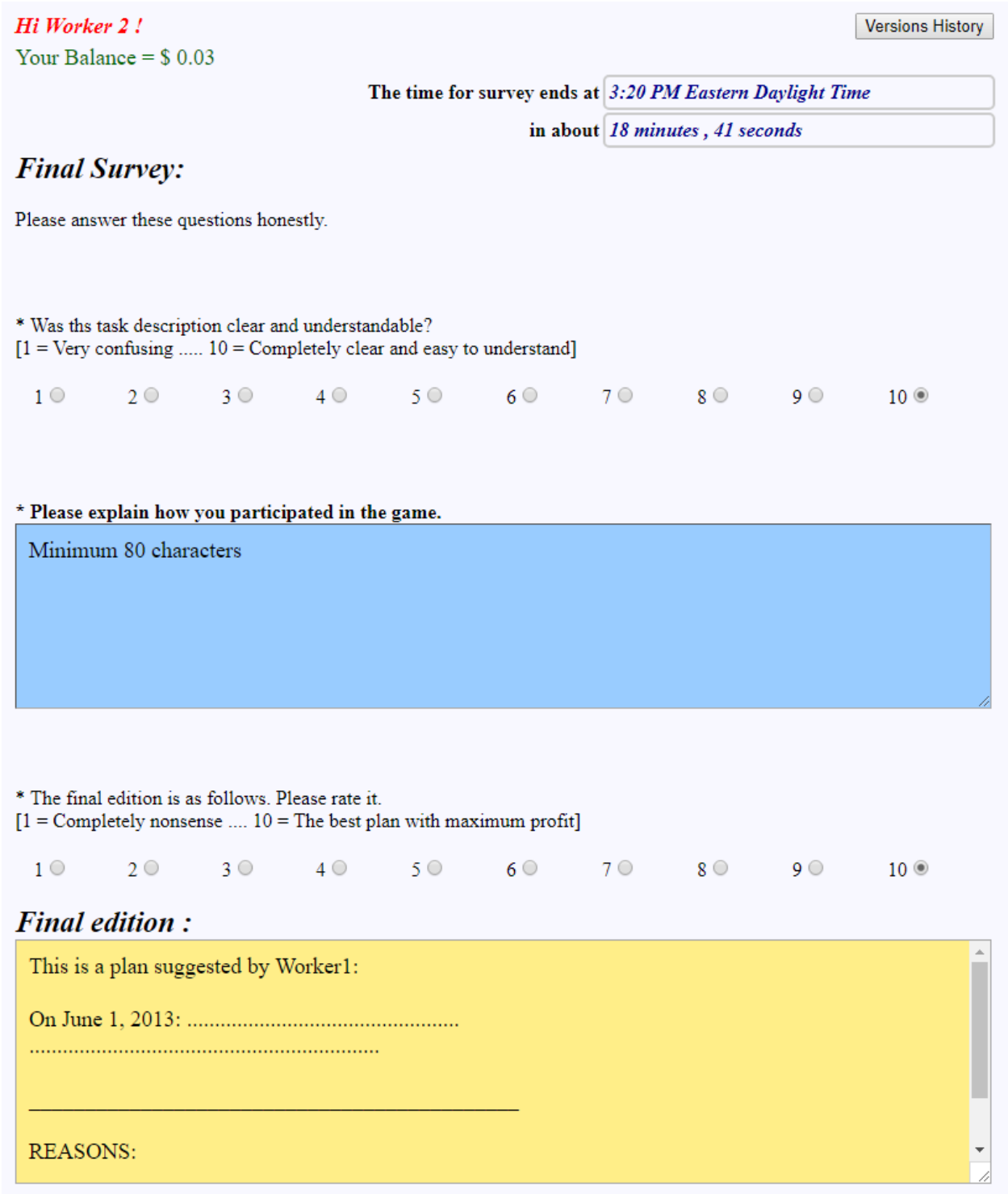
Versions in Round 1:

Initial Edition:

On June 1, 2013: Use 100% of Cash to Buy Dow Jones Industrial Average
On Feb 23, 2015: Sell Dow Jones
On Dec 31, 2015: Use 50% of Cash to Buy IBM & Use 50% of Cash to Buy BTCUSD
On May 31, 2018: Sell IBM & Sell BTCUSD

Figure C-4: History Page for Participants to See the Versions from Previous Periods

APPENDIX D: Final Survey Webpage



Hi Worker 2 !

Versions History

Your Balance = $ 0.03

The time for survey ends at *3:20 PM Eastern Daylight Time*

in about *18 minutes , 41 seconds*

Final Survey:

Please answer these questions honestly.

* Was ths task description clear and understandable?
[1 = Very confusing ..... 10 = Completely clear and easy to understand]

1 2 3 4 5 6 7 8 9 10

*** Please explain how you participated in the game.**

Minimum 80 characters

* The final edition is as follows. Please rate it.
[1 = Completely nonsense .... 10 = The best plan with maximum profit]

1 2 3 4 5 6 7 8 9 10

Final edition :

This is a plan suggested by Worker1:

On June 1, 2013: .....
.....

_____

REASONS:

Figure D-1: Survey Page for Participants to Answer Seven Questions after Experiment

Figure D-1 (cont'd)

* What is your level of expertise in financial investment and stock market?
[1 = Never heard of it.... 10 = A professional trader in financial markets]

1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ◉

*** Please explain your reasons and motives behind your activities in the game.**

Minimum 80 characters

* How was the overall quality of the game and website?
[1 = Very poor .... 10 = Very effective]

1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○ 10 ◉

*** Please describe your experience in this experiment. How can we improve it?**

Minimum 80 characters

Submit Answers

APPENDIX E: Control Panel for the Experimenter

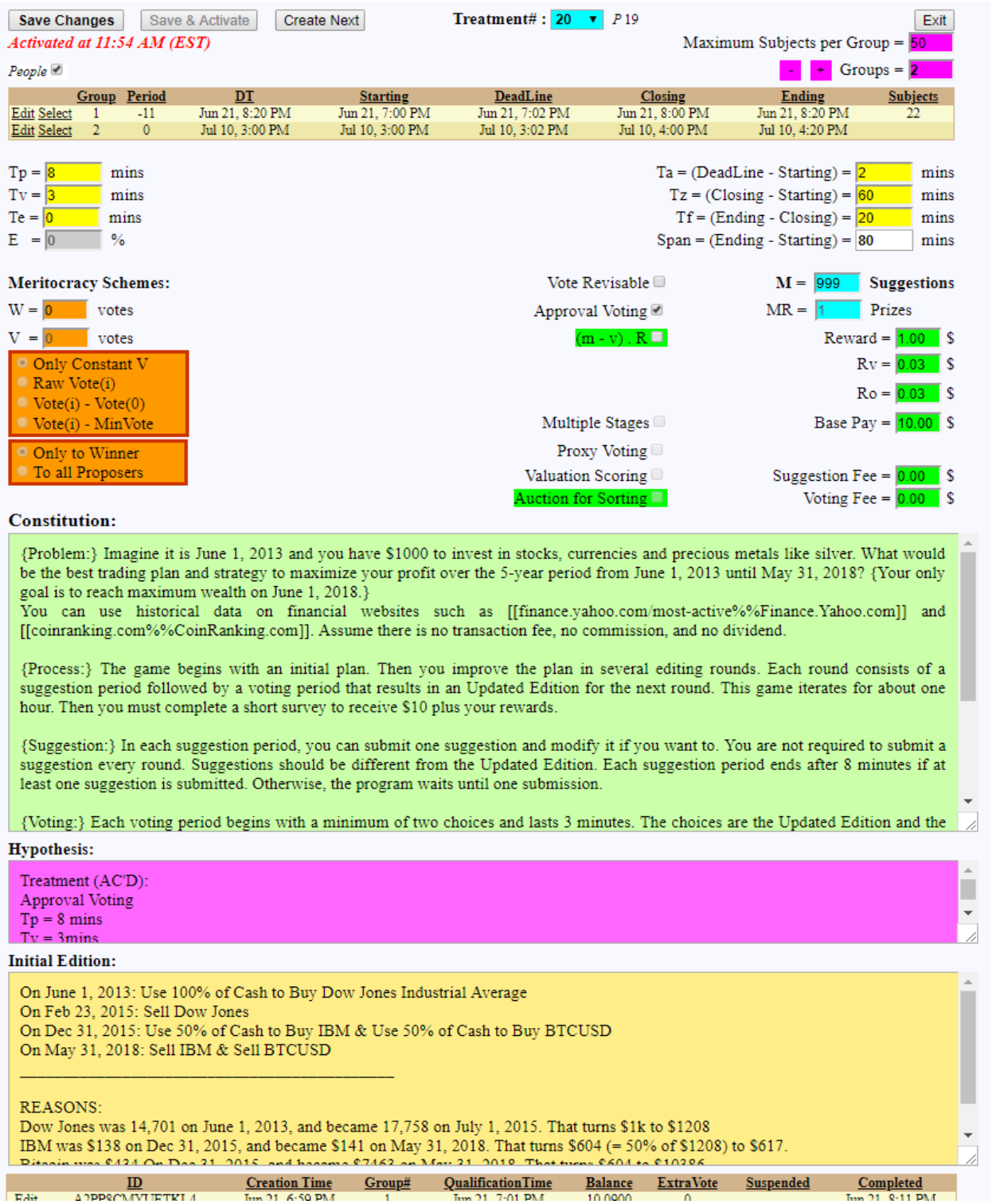


Figure E-1: Control Panel for Experimenter to Instantiate E-constitutions and Treatment Groups

APPENDIX F: Computer Code of the Generic E-Constitution

This appendix presents a computer code that instantiates a generic e-constitution based on constitutional parameters specified in the control panel.

```
/*
Period =
    -9 :  Null
     0 : Before Starting
     1 , 3 , odd : Suggestion
     2 , 4 , even : Voting
    -1 : Final Survey
   -11 : Experiment Ended After Final Survey

DT = The time point indicating the end of the current period.
*/

if (DeadLine < DateTime.Now && !Active) // Did not pass the constitution test and now it is too
late.
{
    ClientScript.RegisterStartupScript(GetType(), "Attention!", "alert('Your time has
expired.');", true);
    return;
}
else if (Period < -10 || !User["Terminated"].Equals(DBNull.Value) )
{
    LabelLogin.Text = User["Name"] + " ! Your final balance is $" +
((float)User["Balance"]).ToString("N2");
    return;
}
else if (Period == -9) // Null : Experiment Not Started Yet
{
    return;
}
else if(Period == 0 || !Active) // Period = 0 or Participant not active yet
{
    Response.Redirect("~/Constitution.aspx");
}
else if (Period == -1) // Final Period and Participant is active
{
    Session["Treat"] = (int)User["Treatment"];
    Session["Group"] = (int)User["Group#"];
    Response.Redirect("~/Survey.aspx");
}
else if ( Period % 2 == 1) // Suggestion Period and Participant is active
    Response.Redirect("~/Suggestion.aspx");
else // if (Period % 2 == 0) // Voting Period and Participant is active
    Response.Redirect("~/Voting.aspx");
```

```
// Submitting Suggestion:
*******************************************************************************
if (DateTime.Now < Suspended)
{
    LabelLogin.Text = "You cannot propose for " + Global.LeftTime(Suspended);
    return;
}
// Already proposed?
query = "select * from Versions where Treatment = " + Treat + " and Group# = " + Group + " and
Period=" +(Period+1)+ " and Proposer=@User and Choice <> 0";
SqlDataReader Version = com.ExecuteReader();
if (Version.Read())
{
    LabelVersion.Text = "You changed your suggestion";

    query = "UPDATE Versions SET Solution = @Solution, Time = @Time, HtmlSolution = @HtmlSolution
" +
        "WHERE Treatment = @Treatment AND Group# = @Group AND Period = @Period AND Choice =
@Choice";

    return;
}
// Number of versions in this period:
query = "select count(*) from Versions where Treatment= " + Treat + " and Group#= " + Group + "
and Period= " + (Period+1);
int m = (int)com.ExecuteScalar();
// Insert the suggestion as a new version:
query = "insert into Versions (Treatment, Group#, Period, Choice, Solution, HtmlSolution,
Proposer, Time) Values (@Treatment, @Group, @Period, @Choice, @Solution, @HtmlSolution,
@Proposer, @Time)";

// Whether it just became enough to close the suggestion period:
if (m >= M || Closing <= DT)
{
    Period++;
    DT = DateTime.Now.AddHours(Tv);
    query = "update Groups set Period=" + Period + " , DT='" + DT + "' where Treatment = " +
Treat + " and Group# = " + Group;

    Global.InviteVoting(Treat, Group, DT);
    Response.Redirect("~/Voting.aspx");
}
```

```
// Casting Vote:
*****************************************************************************************
query = "select * from Versions where Treatment = " + Treat + " and Group# = " + Group + " and
Period = " + Period + " and Proposer = @User and Choice = @Choice";
if(com.ExecuteScalar() != null) // Voting for oneself suggestion?
{
    Message.Text = "You cannot vote for your own edition!";
    return;
}
// Already voted in this period?
query = "select * from Voting where Treatment = " + Treat + " and Group# = " + Group + " and
Period = " + Period + " and Voter = @User";
SqlDataReader Voted = com.ExecuteReader();
if (Voted.Read()) // Already voted
{
    if (!VoteChange) return; // If the constitution does not allow changing votes.
    query = "update Voting set Choice = @Choice, Time = @Time where (Treatment = @Treatment and
Group# = @Group and Period = @Period and Voter = @Voter)";
}
else// Not voted yet
{
    // Meritocracy:
    if (TotalExtra > 0 && W > 0)
    {
        int NewExtra = TotalExtra - VoteW + 1;
        ExtraVotes.Text = "You have " + NewExtra + " Extra Votes left.";
        query = "update People set ExtraVote = " + NewExtra + " where Email = @Voter";
    }
    if (BetFee > 0)
    {
        query = "select Balance from People where Email = @Voter";
        object Balance = com.ExecuteScalar();
        if (BetFee > Balance)
        {
            Message.Text = "Sorry! You do not have enough balance to vote.";
            return;
        }
        LabelLogin.Text = "You spent $" + BetFee + " to vote.";
        LabelBalance.Text = "Your Balance = $" + (Balance - BetFee);

        query = "update People set Balance = " + (Balance - BetFee) + " where Email = @Voter";
    }
    query = "insert into Voting (Treatment, Group#, Period, Choice, Stage, Voter, Time,
VoteWeight, Value) Values (@Treatment, @Group, @Period, @Choice, @Stage, @Voter, @Time,
@VoteWeight, @Value)";
}
Message.Text = "You casted "+ VoteW + " units of vote";
```

```
// At the end of each period: [ if (DT < DateTime.Now) ]

if (Period == -1) // Was final rating period: ********************************************
{
    Period = -11; // Flag the treatment as finished
    DT = DateTime.MaxValue;
}
else if (Period == 0) // Was registration Period: ***********************************************
{
    Period=1;   // Switch to Suggestion Period
    DT = DateTime.Now.AddHours(Tp);
}
else if (Period % 2 == 0)  // Was Voting Period: ********************************************
{
    query = "SELECT CASE WHEN SumVotes IS NULL THEN 0 ELSE SumVotes END AS SumVoteZ,
Versions.Choice, Versions.Proposer, Versions.Solution
FROM(SELECT treatment, Group#, period, choice, sum(VoteWeight) AS SumVotes
FROM Voting GROUP BY treatment, Group#, period, choice) AS VotesOnChoices RIGHT JOIN Versions ON
Versions.Treatment = VotesOnChoices.Treatment AND Versions.Group# = VotesOnChoices.Group# AND
Versions.Period = VotesOnChoices.Period AND VotesOnChoices.Choice = Versions.Choice WHERE
Versions.Treatment=" + Treat + " AND Versions.Group#="+Group+" AND Versions.Period="+Period+"
ORDER BY SumVoteZ DESC, Choice ASC";

    var DataReader = com.ExecuteReader();
    var VersionVotes = new DataTable();
    VersionVotes.Load(DataReader);

    int MinVote = (int)VersionVotes.Rows[VersionVotes.Rows.Count - 1][0];
    int MaxVote = (int)VersionVotes.Rows[0][0];
    int Winner = (int)VersionVotes.Rows[0][1]; // (int)Winning["Choice"];
    string Proposer = (string)VersionVotes.Rows[0][2]; // Version["Proposer"].ToString();
    string Solution = (string)VersionVotes.Rows[0][3]; // Version["Solution"].ToString();

    query = "select * from People where Email = @Proposer";
    User = com.ExecuteReader();

    string ProposerName = (string)User["Name"];
    float Balance = (float)User["Balance"];
    int TotalExtra = (int)User["ExtraVote"];
    query = "insert into Versions(Treatment, Group#, Period , Choice , Solution , HtmlSolution,
Proposer , Time) values(" + Treat + "," + Group + "," + (Period + 2) + ", 0 , @Solution ,
@HtmlSolution, @Proposer , '" + DateTime.Now + "')";
    // Suspend the loser if required by the constitution
    if (Te > 0 && MinVote < MaxVote)
    {
        DateTime Until = DateTime.Now.AddHours(Te);
        query = "update People set Suspended = '" + Until + "' where Email in (' '";
        string Loser;
        for (int i = VersionVotes.Rows.Count - 1; VersionVotes.Rows[i][0] == MinVote &&
VersionVotes.Rows[i][1] != 0;  i--)
        {
            Loser = (string)VersionVotes.Rows[i][2];
            query += ", '" + Loser + "'";
        }
        query += ")";
    }

    // Meritocracy if the constitution merits only the winner.
    if (Winner > 0 && W > 0 && !Merit2All)
    {
        switch (Meritocracy)
        {
            case 1:
                V += MaxVote;
                break;
            case 2:
                V += MaxVote - (int)VersionVotes.Select("Choice=0")[0][0];
                break;
            case 3:
                V += MaxVote - MinVote;
                break;
```

```
    }
    TotalExtra += V;
    query = "update People set ExtraVote = " + TotalExtra + " where Email = @Proposer";
}

// Meritocracy if the constitution merits all the proposers
if (W > 0 && Merit2All)
{
    string Proposeri;
    int Votei;
    int SumVotei;
    switch (Meritocracy)
    {
        // V + Votes(i) --> Every Proposer
        case 1:
            for (int i = 0; i < VersionVotes.Rows.Count; i++)
            {
                if ((int)VersionVotes.Rows[i][1] == 0) continue;

                Votei = (int)VersionVotes.Rows[i][0];
                SumVotei = V + Votei;
                if (SumVotei <= 0)
                    break;

                Proposeri = (string)VersionVotes.Rows[i][2];

       query= "update People set ExtraVote= ExtraVote+ "+ SumVotei+" where Email= @Proposer";
            }
            break;
        // V + Votes(i) - Votes(0) --> Every Proposer
        case 2:
            int Vote0 = (int)VersionVotes.Select("Choice=0")[0][0];
            V -= Vote0;

            for (int i = 0; i < VersionVotes.Rows.Count; i++)
            {
                if ((int)VersionVotes.Rows[i][1] == 0)
                    continue;

                Votei = (int)VersionVotes.Rows[i][0];
                SumVotei = V + Votei;
                if (SumVotei <= 0)
                    break;
                Proposeri = (string)VersionVotes.Rows[i][2];
       query= "update People set ExtraVote= ExtraVote+ "+ SumVotei+" where Email= @Proposer";
            }
            break;
        // V + Votes(i) - MinVotes --> Every Proposer
        case 3:
            V -= MinVote;
            for (int i = 0; i < VersionVotes.Rows.Count; i++)
            {
                if ((int)VersionVotes.Rows[i][1] == 0) continue;

                Votei = (int)VersionVotes.Rows[i][0];
                SumVotei = V + Votei;
                if (SumVotei <= 0) break;
                Proposeri = (string)VersionVotes.Rows[i][2];
       query= "update People set ExtraVote= ExtraVote+ "+ SumVotei+" where Email= @Proposer";
            }
            break;
        // Fixed Votes V --> Every Proposer
        default:
            if (V == 0)  break;
            query = "update People set ExtraVote = ExtraVote + " + V + " where Email in (";
            for (int i = 0; i < VersionVotes.Rows.Count; i++)
            {
                if ((int)VersionVotes.Rows[i][1] == 0) continue;

                Proposeri = (string)VersionVotes.Rows[i][2];
                query += ", '" + Proposeri + "'";
```

```
                }
                query += ")";
                break;
        }
// query = "update People set ExtraVote = ExtraVote + " + V + Votes + " where Email = @Proposer";
    }
    // Reward the winner proposer
    if (Winner > 0 && Reward > 0)
    {
        Balance += Reward;
        query = "update People set Balance = " + Balance + " where Email = @Proposer";
    }

    // Reward the right votes on the winning suggestion
    if (Winner > 0 && Rv > 0)
    {
query = "update People set Balance = Balance + "+ Rv +" where Email in (select Voter from Voting
where Choice= "+Winner+" and Treatment="+Treat+" and Group#="+Group+" and Period= "+Period+")";
    }
    // Reward the right votes on the updated edition
    if (Winner == 0 && Ro > 0 && MaxVote > 0)
    {
query = "update People set Balance = Balance + " + Ro + " where Email in (select Voter from
Voting where Choice=0 and Treatment="+Treat+" and Group#="+Group+" and [Period]="+Period+")";
    }

    if (Closing < DateTime.Now) // If the process ended.
    {
// Switching to the Final Period: ************************************************************
        Period = -1;
        DT = EndingTime;
    }
    else
    {
        Period++;   // Switch to Suggestion Period
        DT = DateTime.Now.AddHours(Tp);
    }
else if (Period % 2 == 1) // Was Suggestion Period: *******************************************
{
    query = "select count(*) from Versions where Treatment=" + Treat + " and Group#=" + Group + "
and Period=" + (Period+1);
    int m = (int)com.ExecuteScalar();
    if (m > 1) // Enough suggestions for voting
    {
        Period++; // Switch to Voting Period
        DT = DateTime.Now.AddHours(Tv);
        InviteVoting(Treat, Group, DT);
    }
    else // if (m==1) : Not enough suggestions for Voting
    {
        if (Closing < DateTime.Now) // Switching to the Final Period:
        {
        Period = -1;
        DT = EndingTime;
        else
        DT = Closing; // Stay in the suggestion period and wait for a suggestion until the end
    }
}
```

APPENDIX G: Database of the Website

Figure G-1 illustrates the data structure diagram of the database used in the website. As it shows, the database has six tables including a table for the treatments. As illustrated, each participant, version, vote and rating score belongs to a treatment and hence treatment number is a foreign key in all other tables and its corresponding relationships are highlighted in green. The treatment table includes the parameters for e-constitutions that the website can instantiate.

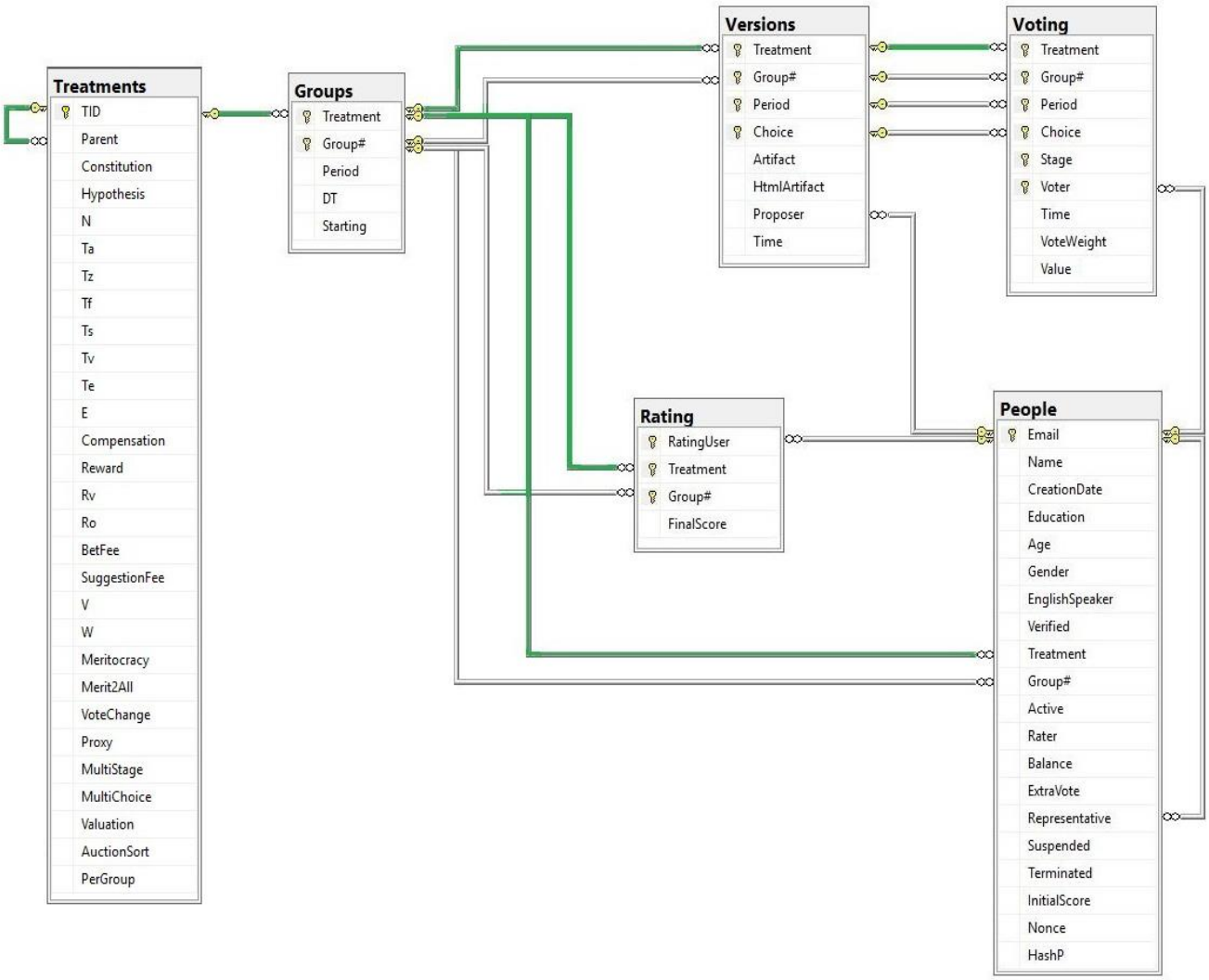
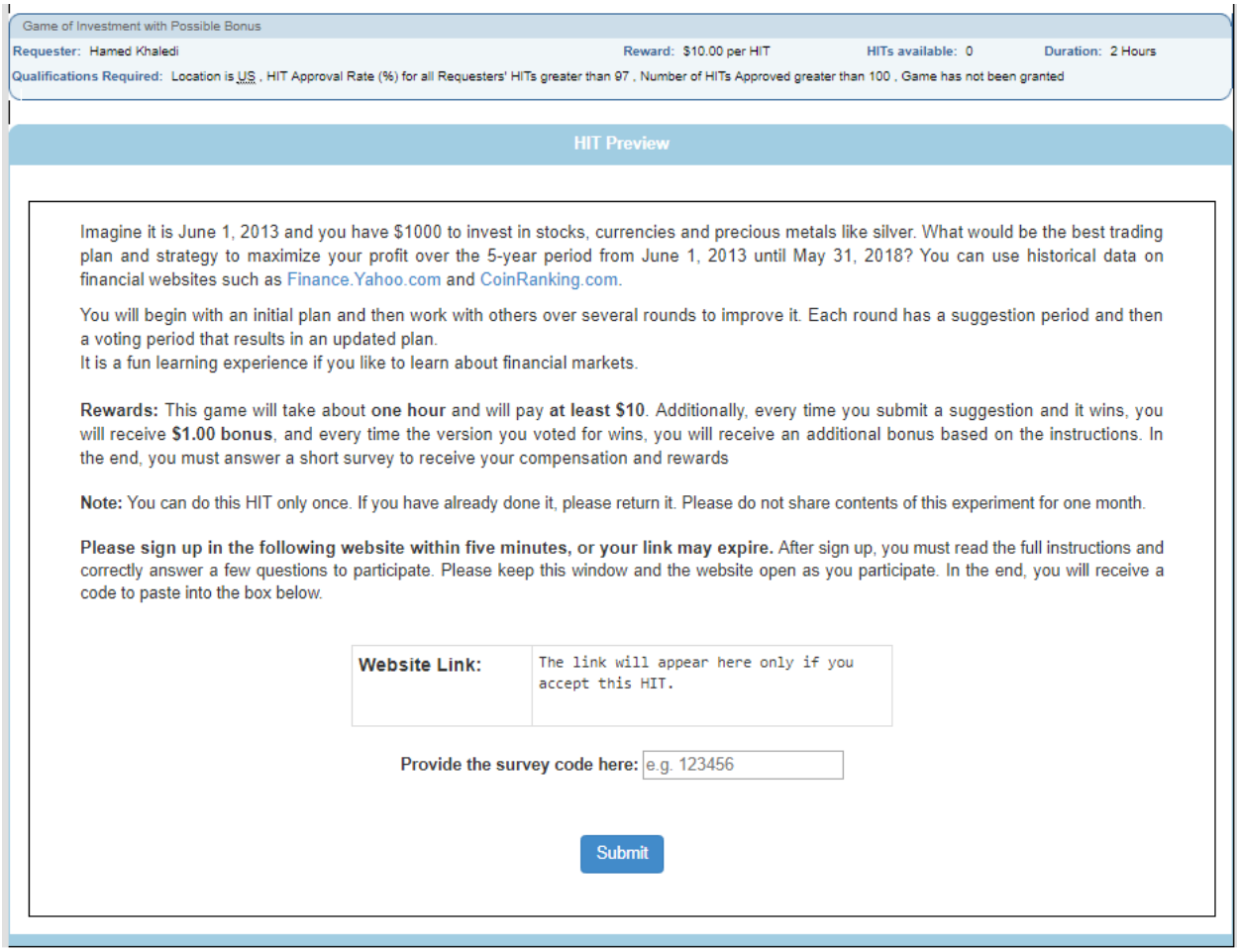


Figure G-1: Data Structure Diagram for the Database Used in the Web Application

APPENDIX H: HIT Description in MTurk



Game of Investment with Possible Bonus

Requester: Hamed Khaledi **Reward:** $10.00 per HIT **HITs available:** 0 **Duration:** 2 Hours

Qualifications Required: Location is US , HIT Approval Rate (%) for all Requesters' HITs greater than 97 , Number of HITs Approved greater than 100 , Game has not been granted

HIT Preview

Imagine it is June 1, 2013 and you have $1000 to invest in stocks, currencies and precious metals like silver. What would be the best trading plan and strategy to maximize your profit over the 5-year period from June 1, 2013 until May 31, 2018? You can use historical data on financial websites such as Finance.Yahoo.com and CoinRanking.com.

You will begin with an initial plan and then work with others over several rounds to improve it. Each round has a suggestion period and then a voting period that results in an updated plan.
It is a fun learning experience if you like to learn about financial markets.

Rewards: This game will take about **one hour** and will pay **at least $10**. Additionally, every time you submit a suggestion and it wins, you will receive **$1.00 bonus**, and every time the version you voted for wins, you will receive an additional bonus based on the instructions. In the end, you must answer a short survey to receive your compensation and rewards

Note: You can do this HIT only once. If you have already done it, please return it. Please do not share contents of this experiment for one month.

Please sign up in the following website within five minutes, or your link may expire. After sign up, you must read the full instructions and correctly answer a few questions to participate. Please keep this window and the website open as you participate. In the end, you will receive a code to paste into the box below.

Website Link:	The link will appear here only if you accept this HIT.

Provide the survey code here: e.g. 123456

Submit

Figure H-1: HIT Description for the Workers in Mechanical Turk

APPENDIX I: Constitution for Treatment (1)

Problem: *Imagine it is June 1, 2013 and you have $1000 to invest in stocks, currencies and precious metals like silver. What would be the best trading plan and strategy to maximize your profit over the 5-year period from June 1, 2013 until May 31, 2018?* ***Your only goal is to reach maximum wealth on June 1, 2018.*** *You can use historical data on financial websites such as Finance.Yahoo.com and CoinRanking.com. Assume there is no transaction fee, no commission, and no dividend.*

Process: *The game begins with an initial plan. Then you improve the plan in several editing rounds. Each round consists of a suggestion period followed by a voting period that results in an Updated Edition for the next round. This game iterates for about one hour. Then you must complete a short survey to receive $10 plus your rewards.*

Suggestion: *In each suggestion period, you can submit one suggestion and modify it if you want to. You are not required to submit a suggestion every round. Suggestions should be different from the Updated Edition. Each suggestion period ends after 4 minutes if at least one suggestion is submitted. Otherwise, the program waits until one submission.*

Voting: *Each voting period begins with a minimum of two choices and lasts 4 minutes. The choices are the Updated Edition and the submitted suggestions. You cannot vote for your own edition. You are not required to vote every round. You cannot change your vote within a voting period.*

Winning: *After each voting period, the most voted version wins and becomes the Updated Edition for the next period. Then the winning version and the winner's name are announced. Other suggestions and all votes remain anonymous.*
Then, the next suggestion period begins.

Rewards: *After each voting period, if your suggestion wins, you will receive $1.00 bonus, and if the choice you voted for wins, you will receive $0.03 bonus.*

APPENDIX J: Price-Based Constitution

This appendix presents a constitution based on market equilibrium prices as the criterion for selection. Price is a collective decision that is least vulnerable to attacks and malicious activities. Market price is the most efficient aggregation of dispersed information (Hayek, 1945). In nonmarket mechanisms, participants lack adequate incentives to estimate or reveal the value of the contributions accurately (Ba, et al., 2001b). Market price is a sufficient statistic that summarizes relative strengths of selectors' preferences over alternatives. The possibility to compare individual preferences yields a group utility function and unshackles us from the impossibility theorems (Scott & Antonsson, 1999). Hence, it is Pareto-efficient, monotone, independent from irrelevant alternatives and non-dictatorial with any number of choices.

Ren et al. (2017) presented a process model that used price to evaluate choices. Perhaps, that is the best attempt thus far to use price for collective selection. However, their method is not applicable to a design process. In the *price-based constitution*, participants can trade shares of versions during selection (trading) periods. Then, at the end of each period, it uses the equilibrium prices as the aggregate evaluation of the versions, judges the version with the highest share price as the most valuable version, and makes it the updated edition. Then, it voids all transactions of all other versions to prevent forking. The e-constitution contains an automatic market maker that buys and sells to match asks and bids. A fully automated electronic exchange can serve as market maker (Ba, et al., 2001b). Automatic market makers can use algorithms to adjust prices based on transactions and give instant feedback to traders (Boer, et al., 2007). One applied market maker is The *Logarithmic Market Scoring Rules* (Jian & Sami, 2010). Meanwhile, markets should have at least 30 participants to be efficient (Christiansen, 2007). Otherwise, a "thin market" with too few traders cannot effectively aggregate information (Healy, et al., 2010).

The price-based constitution inherits an endogenous meritocracy from market competition. Traders are responsible for their decisions and have incentive to obtain information and evaluate versions accurately and then trade wisely or not trade if they do not have the expertise or information to do so. Hence, unqualified traders incur all the costs of their bad decisions and eventually fade out from the market. Smart

traders benefit from their informed selections, survive in the market, make more trades, and exert more influence in selection.

This constitution sorts the versions based on their prices so that the most valued versions are exposed to be traded more often and thus are priced more precisely. After all, the only selection criterion is having the maximum price and thus only the relative prices of the best versions matter. Therefore, the outcome is not sensitive to prices of the average versions or the magnitude of the prices in general. Hence, the typical over-optimism common in prediction markets does not cause a problem here. The constitution in plain text is as follows:

General Process: *At the beginning, for a period of* $\boldsymbol{T_A}$ *time, there is an auction for the shares (100%) of the initial solution and anyone can buy the shares to invest in the initial solution. The shareholders own the solution and its IP rights proportional to their shares. Budget B is the total money invested to be used to reward winning suggestions.*

The solution evolves through several editing rounds. Each editing round consists of a suggestion period followed by a trading (selection) period, repeating until running out of budget B. Anyone can participate in suggestion, trading or both.

Suggestion Period: *A participant can submit only one suggestion during each suggestion period if they want to. The proposers (participants who submitted suggestions) can modify their suggestions until the trading period begins.*

In the first round, the "Updated Edition" is the "Initial Edition." Afterwards, the "Updated Edition" is updated to be the winning version from the previous round.

Ending Suggestion Period: *A suggestion period ends after* $\boldsymbol{T_P}$ *time if AT LEAST one suggestion is submitted. Then the trading period begins with a minimum of two choices including the submitted suggestion(s) and the "Updated Edition." If there is no suggestion in the* $\boldsymbol{T_P}$ *time, the program waits until one suggestion is submitted and then the trading period begins immediately.*

Trading Period: *Each trading period lasts* $\boldsymbol{T_V}$ *time, during which participants can trade the shares of all versions, including the "Updated Edition," and the prices are constantly the result of anonymous asks and bids. Versions are continuously sorted based on their equilibrium share prices so that the highest prices are at the top.*

At the beginning of each trading period, the share price of all suggested versions are initially set to the share price of their parent version (Updated Edition).

Winning Version: *After each trading period, the version with the highest equilibrium share price wins and shall be published with all its transactions confirmed. The transactions and shares of all other versions are voided. Unless the winning version is the "Updated Edition," the proposer of the winning version shall receive a compensation equal to* ***R*** *times the amount of increase in the share price and the winner's (chosen) name is announced.*

Then the next suggestion period begins for further modifications if the design process has not ended.

Accounting: *The shareholders of a version own the shares of all versions derived (forked) from it and can sell all or any of them. Traders can spend the same money in multiple markets, but at the end of each trading period, only the balances for the winning version are valid and all other parallel balances become void. Conversely, when a trader cashes out, the withdrawal amount is deducted from all his parallel balances. In a trading period, the maximum amount that a trader can cash out is the minimum of all his parallel balances.*

One accounting technicality is that a trader cannot cash out an amount that may result in a negative balance on any version. At the end of each trading period, only one version becomes valid and transactions of other versions become voided. The winning version can be any version in a period. Therefore, during a trading period, traders can withdraw the least of their balances on all versions. However, at the end of each round, traders can withdraw all their confirmed balances on the winning version.

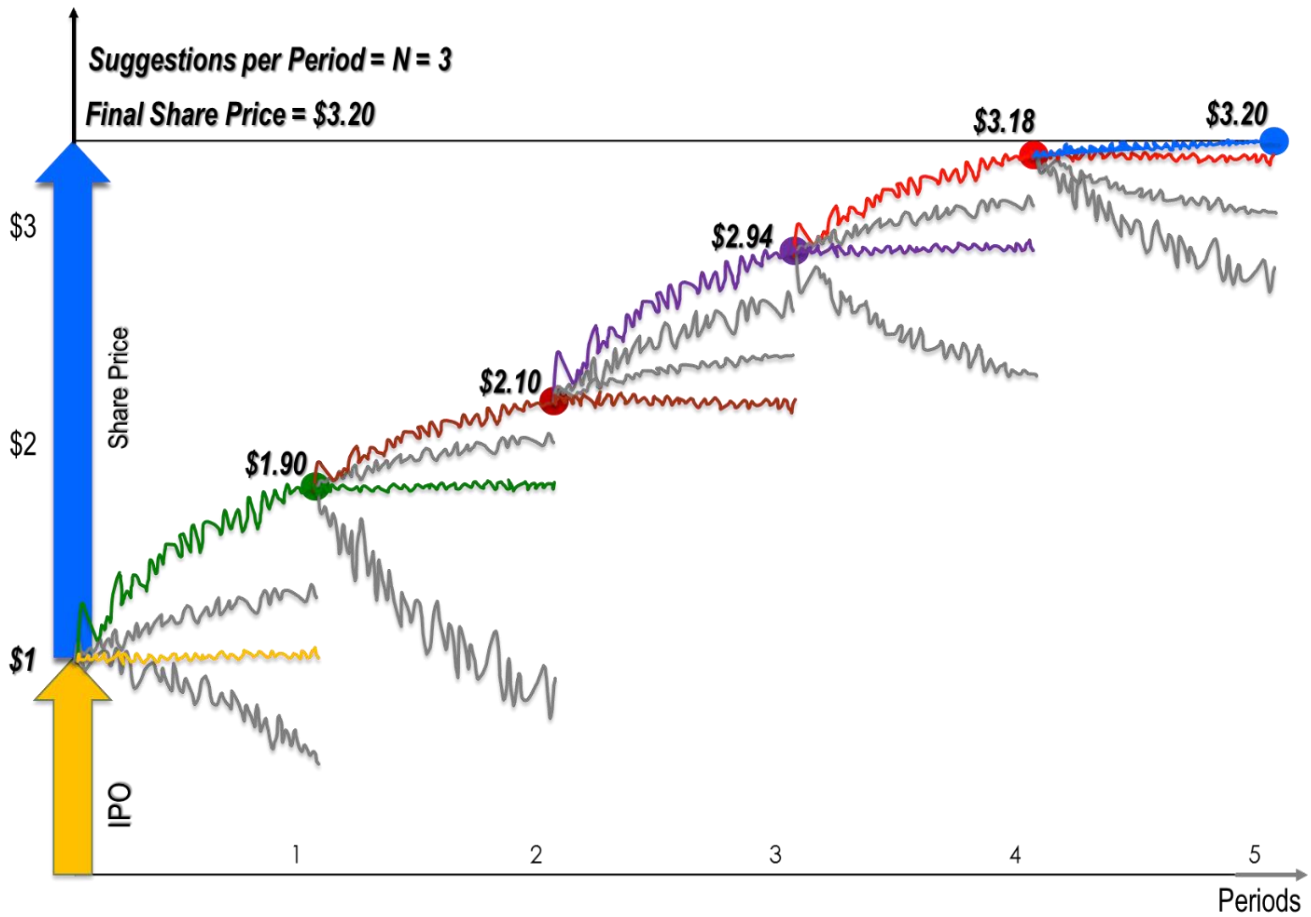


Figure J-1: Prices of Parallel Versions Using a Price-based Constitution (Example)

Figure J-1 illustrates how the prices may change during selection periods. The picture only shows the trading periods back to back hiding the suggestion periods between them because the prices cannot change during the suggestion periods. By default, this constitution does not allow for trade during suggestion periods, and has distinct suggestion and trading periods and all suggestions in a round start competing at the same time at the beginning of the trading period. This prevents participants from copying other suggestions and make the competition fairer in some situations.

Another approach is to reveal each suggestion once submitted and let traders trade at any time even during the suggestion periods. It improves the liquidity of the market and helps to reach the equilibrium faster. Accordingly, each iteration has a suggestion period with the possibility of trading, and then there is an exclusive trading period without any suggestion, so that all prices reach relative equilibrium before the

selection of the maximum price. Allowing trades during the suggestion periods automatically rewards the best proposers while eliciting their information on their suggestions. A proposer of a superior suggestion is the first one who knows about the value of that suggestion and can be the first buyer before its price rises by other investors. Additionally, at the same time the proposer informs others about the value of that suggestion by raising its price and making it more visible in the rankings. If a proposer strongly believes in the quality of his proposed suggestion, he can use his exclusive information to extract the most profit out of it before anyone else. Perhaps, this possibility obviates an explicit compensation clause in the constitution and eliminates the problem of copying others' suggestions.

This constitution while may seem complicated, is easier to implement on a blockchain. Each suggestion forks the blockchain and after a pre-specified trading period (i.e. number of blocks), only the branch with the highest price for the token (share) is retained and all other branches become invalid automatically. Therefore, each suggestion is essentially a controlled purposive forking and the selection criterion determines which branch is valid as the canonical chain.

The price-based constitution can be very effective for making decisions in publicly traded corporations. For any decision, each choice creates a parallel market for the corporate shares, functioning as if that specific choice will be taken. Therefore, the corporate shares are traded in different parallel markets based on hypothetical decisions. The parallel markets function for a period until reaching an equilibrium or the deadline for making the decision and then the choice that yielded the highest share price becomes effective and its transactions will be confirmed. However, shares and transactions corresponding to alternatives will be voided. Everybody can propose a decision or improvement but only decisions will be implemented that maximize the share price. Accordingly, the e-constitution can replace the CEO by making all decisions through financial markets. This eliminates the moral hazard problem altogether.

One important challenge to use this constitution is that it requires an exogenous criterion to value the final edition and to pay off the shareholders at the end. This is necessary because the incentives for trading shares come from the prospect of final profit. Corporations associate annual profit and net income to the shares and that directs the trades. Otherwise, if there is a market for the final solution or product, its selling

price or revenue serves as the exogenous evaluation and incentivizes thoughtful investment. However, if such exogenous criteria do not exist, we need to define the quality of choices and incentivize investment on higher quality ones. One approach is to hire some independent raters who do not benefit from any alternative and have not traded any shares, and have them assign financial values to the final edition and then compensate the shareholders based on the median of the values given by the raters. Also, reward the rater(s) whose valuation was closest to the median. To make the rating (valuation) process more reliable and accurate, one may select a random sample of independent raters. However, we are back to the aggregation of voting and rating for evaluation. Prediction markets for idea evaluation need to tie the payoffs to a real observable outcome on which the participants can bet. Otherwise, they bet on expert evaluations instead of the idea quality (Blohm, et al., 2011).

Blohm et al. (2011) explained that markets need several participants and several trades to reach a meaningful equilibrium price as evaluation, whereas other evaluation methods like rating only need one or two participant and one round to result in a meaningful evaluation through an aggregation function. They compared the performance of rating scales with prediction markets, and found that rating scales result in better evaluation accuracy and higher satisfaction for the participants. They concluded that rating is a better mechanism than trading. However, there were two problems with their experiment. First, in the prediction markets, the participants did not use real money to provide incentives, which is the main point of using markets. Second, the results were compared against some other rating scores given by a panel of experts. It suffers a method bias in favour of rating and against the prediction markets.

Similarly, Gottschlich and Hinz (2014) developed a Decision Support System for designing stock portfolios based on crowds' votes. They showed that the crowdsourced investment recommendations outperform the market benchmark and comparable public funds. However, the market benchmark does not result from market equilibrium, but rather a portfolio that is decided by a group of people through some collective decision making process other than equilibrium price.

Blume et al. (2010) claimed that markets are open to gaming because a participant can transfer money from one account to another and trade with himself to manipulate the price. However, they overlook how

bids and asks clear in liquid markets. If a participant tries to make a trade at a price far from the equilibrium price, his bid or ask will first clear the existing asks or bids that are closer to the equilibrium price, not his own opposing offer at his planned price. The only way to trade with oneself (or accomplice) is outside the market, which does not affect the market price. Moreover, the only way to manipulate the price is through large amounts of trading, and incurring all the costs associated with moving the price.

BIBLIOGRAPHY

BIBLIOGRAPHY

Acemoglu, D. & Robinson, J., 2012. *Why Nations Fail: The Origins of Power, Prosperity and Poverty.* 1st ed. New York: Crown Publishers.

Archak, N. & Sundararajan, A., 2009. *Optimal Design of Crowdsourcing Contests.* Phoenix, The 13th International Conference on Information Systems.

Ariely, D., Loewenstein, G. & Prelec, D., 2003. Coherent Arbitrariness: Stable Demand Curves without Stable Preferences. *Quarterly Journal of Economics,* 118(1), pp. 73-105.

Atzei, N., Bartoletti, M. & Cimoli, T., 2017. *A Survey of Attacks on Ethereum Smart Contracts.* Heidelberg, Springer, p. 164–186.

Bandiera, O., Barankay, I. & Rasul, I., 2013. Team incentives: evidence from a firm level experiment. *Journal of the European Economic Association,* Volume 11, pp. 1079-1114.

Bao, J., Sakamoto, Y. & Nickerson, J., 2011. *Evaluating Design Solutions Using Crowds.* Detroit, MI, Americas Conference on Information Systems.

Ba, S., Stallaert, J. & Whinston, A. B., 2001a. Research Commentary: Introducing a Third Dimension in Information Systems Design — The Case for Incentive Alignment. *Information Systems Research,* 12(3), pp. 225-239.

Ba, S., Stallaert, J. & Whinston, A. B., 2001b. Optimal Investment in Knowlege within a Firm Using a Market Mechanism. *Managment Science,* 47(9), pp. 1203-1219.

Benoit, J.-P. & Kornhauser, L., 2010. Only a Dictatorship is Efficient. *Games and Economic Behavior.*

Bix, B., 2013. Boilerplate, Freedom of Contract and Democratic Degradation. *The Tulsa Law Review,* Volume 49.

Blohm, I., Riedl, C., Leimeister, J. M. & Krcmar, H., 2011. *Idea Evaluation Mechanisms for Collective Intelligence in Open Innovation Communities: Do Traders Outperform Raters?* Shanghai, Thirty Second International Conference on Information Systems.

Blume, M., Luckner, S. & Weinhardt, C., 2010. Fraud Detection in Play-Money Prediction Markets. *Information Systems and E-Business Management,* 8(4), pp. 395-413.

Boer, K., Kaymak, U. & Spiering, J., 2007. From Discrete-Time Models to Continuous-Time Asynchronous Modeling of Financial Markets. *Computational Intelligence,* 23(2), pp. 142-161.

Bonabeau, E., 2009. Decisions 2.0: The Power of Collective Intelligence. *MIT Sloan Management Review,* 50(2), pp. 44-53.

Boreham, R. & Rutter, K., 2018. *R3.* Available at: https://www.r3.com/research/

Brabham, D., 2008. Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. *Convergence: The International J. of Research into New Media Technologies,* Volume 14, pp. 75-90.

Brooks, F. P., 2010. *The Design of Design: Essays from a Computer Scientist.* 2nd ed. Upper Saddle River, NJ: Addison-Wesley.

Buchanan, J. & Tullock, G., 1961. *The Calculus of Consent: Logical Foundations of Constitutional Democracy.* 1st ed. Ann Arbor: University of Michigan Press.

Buterin, V., 2013. *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.* Available at: http://ethereum.org/ethereum.html

Cerasoli, C., Nicklin, J. & Ford, M., 2014. Intrinsic Motivation and extrinsic incentives jointly predict performance: A 40-year meta-analysis. *Psychol Bull,* 140(4), pp. 965-980.

Chan, J., Dang, S. & Dow, S., 2016. *Improving Crowd Innovation with Expert Facilitation.* San Francisco, CA, ACM.

Chanron, V. & Lewis, K., 2005. A Study of Convergence in Decentralized Design Processes. *Research in Engineering Design,* Volume 16, pp. 133-145.

Chaum, D., 2015. *Random-Sample Voting: Far lower cost, better quality and more democratic,* New York: Scribd.

Chilton, L., Landay, J. & Weld, D., 2016. *HumorTools: A Microtask Workflow for Writing News Satire.* El Paso, Texas, ACM.

Chilton, L. et al., 2013. *Cascade: Crowdsourcing Taxonomy Creation.* Paris, France, CHI.

Christiansen, J. D., 2007. Prediction Markets: Practical Experiments in Small Markets and Behaviors Observed. *The Journal of Prediction Markets,* 1(1), pp. 17-41.

Clarkson, G. & Alstyne, M. V., 2007. *The Social Efficiency of Fairness: An Innovation Economics Approach to Innovation.* Hawaii, 40th Annual Hawaii International Conference on Systems Sciences.

Collins, J. & Porras, J., 1994. *Built to Last: Successful Habits of Visionary Companies.* 1st ed. New York: William Collins.

Davis, J. & Lin, W. H., 2011. *Web 3.0 and Crowdservicing.* Detroit, MI, America's Conference on Information Systems.

Davis, K., 2015. InnoCentive.com Collaboration Case Study. *Journal of Management Policies and Practices,* 3(1), pp. 20-22.

De Paola, M., Scoppa, V. & Nistico, R., 2012. Monetary incentives and student achievement in a depressed labor market: Results from a randomized experiment. *Journal of Human Capital,* Volume 6, pp. 56-85.

Dechanaux, E., Kovenock, D. & Sheremeta, R. M., 2015. A Survey of Experimental Research on Contests, All-Pay Auctions and Tournaments. *Experimental Economics,* Volume 18, pp. 609-669.

Delfgauw, J., Dur, R., Sol, J. & Verbeke, W., 2013. Tournament incentives in the field: Gender differences in the workplace. *Journal of Labor Economics,* Volume 31, pp. 305-326.

Deng, Q. & Ji, S., 2018. A Review of Design Science Research in Information Systems: Concepts, Process, Outcome, and Evaluation. *Pacific Asia Journal of the Association for Information Systems,* 10(1), pp. 1-36.

Easley, D. & Kleinberg, J., 2010. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* 1st ed. Cambridge, MA: Cambridge University Press.

Edge, A. G. & Remus, W., 1984. The Impact of Hierarchical and Egalitarian Organization Structure on Group Decision Making and Attitudes. *Developments in Business Simulation & Experiential Learning,* Volume 11.

Ertekin, S., Rudin, C. & Hirsh, H., 2013. Approximating the crowd. *Data Mining and Knowledge Discovery,* 1(1), pp. 1-32.

Fidler, D., 2008. A Theory of Open-Source Anarchy. *Indiana Journal of Global Studies,* Volume 15.

Freshtman, C. & Gneezy, U., 2011. The trade-off between performance and quitting in high-power tournaments. *Journal of the European Economic Association,* Volume 9, pp. 318-336.

Friedman, D., 2000. Contracts in Cyberspace. *American Law and Economics Association Mtg,* 4 May.

Fullerton, R. & McAfee, P., 1999. Auctioning Entry into Tournaments. *Journal of Political Economy,* 107(3), pp. 573-605.

Gallupe, R. B. et al., 1992. Electronic Brainstorming and Group Size. *Academy of Management Journal,* Volume 32, pp. 350-369.

Gardiner, P. D. & Stewart, K., 2000. Revisiting the golden triangle of cost, time and quality: the role of NPV in project control, success and failure. *International Journal of Project Management,* 18(4), pp. 251-256.

Gillick, D. & Liu, Y., 2010. *Non-Expert Evaluation of Summarization Systems is Risky.* Los Angeles, California, Association for Computational Linguistics, pp. 148-151.

Gneezy, U. & Rustichini, A., 2000. Pay enough or don't pay at all. *Quarterly Journal of Economics,* 115(3), pp. 791-810.

Gottschlich, J. & Hinz, O., 2014. A Decision Support System for Stock Investment Recommendations using Collective Wisdom. *Decision Support Systems,* 59(1), pp. 52-62.

Gregor, S., 2006. The Nature of Theory in Information Systems. *MIS Quarterly,* 30(2), pp. 611-642.

Gregor, S. & Hevner, A. R., 2013. Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly,* 37(2), pp. 337-355.

Gregor, S. & Jones, D., 2007. The Anatomy of a Design Theory. *Journal of the Association for Information Systems,* 8(5), pp. 312-335.

Grigg, I., 2017. *EOS- An Introduction.* Available at: https://eos.io/documents/EOS_An_Introduction.pdf

Hansen, M. T., 2009. *Collaboration: How leaders avoid the traps, create unity, and reap big results.* 1st ed. Boston, MA: Harvard Business Press.

Hayek, F. A., 1945. The Use of Knowledge in Society. *American Economic Review,* 35(4), pp. 519-530.

Hazelrigg, G. A., 1996. The Implications of Arrow's Impossibility Theorem on Approaches to Optimal Engineering Design. *Journal of Mechanical Engineering,* 118(1), pp. 161-164.

Healy, J., Linardi, S., Lowery, J. R. & Ledyard, J. O., 2010. Prediction Markets: Alternative Mechanisms for Complex Environments with Few Traders. *Management Science,* 56(11), pp. 1977-1996.

Heer, J. & Bostock, M., 2010. *Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design.* Atlanta Georgia, CHI.

Hevner, A., March, S., Park, J. & Ram, S., 2004. Design Science in Information Systems Research. *MIS Quarterly,* 28(1), pp. 75-105.

Heyman, J. & Ariely, D., 2004. Effort for Payment: A Tale of Two Markets. *Psychological Science,* 15(11), pp. 787-793.

Hill, S. & Ready-Campbell, N., 2011. Expert Stock Picker: The Wisdom of (Experts in) Crowds. *International Journal of Electronic Commerce,* 15(1), pp. 73-102.

Hoffman, L., 2009. Crowd Control. *Communications of the ACM,* 52(3), pp. 16-17.

Holmstrom, B., 1982. Moral Hazard in Teams. *The Bell Journal of Economics,* 13(2), pp. 324-340.

Holston, J., Issarny, V. & Parra, C., 2016. *Engineering Software Assemblies for Participatory Democracy: The Participatory Budgeting Use Case.* Austin, TX, USA, ACM 38th International Conference on Software Engineering Companion.

Hong, L. & Page, S., 2001. Problem Solving by Heterogeneous Agents. *Journal of Economic Theory,* Volume 97, pp. 123-163.

Hong, L., Page, S. E. & Riolo, M., 2012. Incentives, Information, and Emergent Collective Accuracy. *Managerial and Decision Economics,* 19 July, Volume 33, pp. 323-334.

Horton, J. & Chilton, L., 2010. *The Labor Economics of Paid Crowdsourcing.* Cambridge MA, ACM.

Hossain, T., Hong, F. & List, J. A., 2014. Framing manipulations in contests: A natural field experiment. *Working Paper.*

Howe, J., 2006. The Rise of Crowdsourcing. *Wired Magazine,* 14(1), pp. 1-5.

Huang, Y., Singh, P. V. & Mukhopadhyay, T., 2012. *Crowdsourcing Contest: A Dynamic Structural Model of the Impact of Incentive Structure on Solution Quality.* Orlando, 33rd International Conference on Information Systems.

Jackson, M., 2003. Mechanism Theory. *Humanities and Social Sciences*, pp. 228-77.

Jain, R., 2010. *Investigation of Governance Mechanisms for Crowdsourcing Initiatives.* Lima, Peru, AMCIS 2010 Proceedings.

Jian, L. & Sami, R., 2010. *Aggregation and Manipulation in Prediction Markets: Effects of Trading Mechanism and Information Distribution.* Cambridge, Massachusetts, 11th ACM Conference on Electronic Commerce.

Johnson, D. & Post, D., 1996. Law and Borders – The Rise of Law in Cyberspace. *Stanford Law Review,* pp. 1367-1377.

Johnson, J., 2007. Social Networks and the Wisdom of Crowds. *Network World,* 24(1), pp. 28-29.

Juels, A., Kosba, A. & Shi, E., 2016. *The Ring of Gyges: Using Smart Contracts for Crime.* Vienna, Austria, 23rd ACM Conference on Computer and Communications Security (CCS).

Kahneman, D., 2011. *Thinking, Fast and Slow.* 1st ed. New York: Farrar, Straus and Giroux.

Keuschnigg, M., Bader, F. & Bracher, J., 2016. Using crowdsourced online experiments to study context-dependency of behavior. *Social Science Research,* Volume 59, pp. 68-82.

Khuri, A. & Mukhopadhyay, S., 2010. Response surface methodology. *WIREs Computational Statistics,* 2(March/April), pp. 128-149.

Kim, S.-H., 2016. On the Optimal Social Contract: Agency Costs of Self-Government. *Journal of Comparative Economics,* 44(1), pp. 982-1001.

Kline, W., Kotabe, M., Hamilton, R. & Ridgley, S., 2017. Organizational Constitution, Organizational Identification, and Executive Pay. *Asia-Pacific Journal of Business Administration,* 9(1), pp. 51-68.

Kornberger, M., 2016. The visible hand and the crowd: Analyzing organization design in distributed innovation systems. *Strategic Organization,* Organizing Crowds and Innovation(1), pp. 1-20.

Kornrumpf, A. & Baumol, U., 2014. *A Design Science Approach to Collective Intelligence Systems.* Hawaii, 47th International Conference on System Science.

Kosorukoff, A., 2000. *Human-Based Genetic Algorithm.* Available at: http://www.HBGA.com

Kosorukoff, A., 2000. *Social Classification Structures: Optimal Decision Making in an Organization.* Las Vegas, Nevada, Genetic and Evolutionary Computation Conference.

Kosorukoff, A., 2001. Human Based Genetic Algorithms. *Proceeding of IEEE Conference on Systems, Man, and Cybernetics,* pp. 3464-3469.

Kosorukoff, A. & D., G., 2002. Evolutionary Computation as a form of Organization. *IEEE.*

Kyriakou, H., Nickerson, J. V. & Sabnis, G., 2017. Knowledge Reuse for Customization: Metamodels in an Open Design Community for 3D Printing. *MIS Quarterly,* 41(1), pp. 315-332.

Lazear, E. P. & Rosen, S., 1981. Rank-order tournaments as optimum labor contracts. *Journal of Political Economy,* Volume 89, pp. 841-864.

Leimeister, J. M., 2010. *Collective Intelligence,* Cambridge, MA: Business & Information Systems Engineering.

Leimeister, J. M., Huber, M., Bretschneider, U. & Krcmar, H., 2009. Leveraging Crowdsourcing: Activation-Supporting Components for IT-Based Ideas Competition. *Journal of Management Information Systems,* 26(1), pp. 197-224.

Lévy, P., 1997. *Collective Intelligence: Mankind's Emerging World in Cyberspace.* New York, Plenum.

Liebenaua, J. & Harindranathb, G., 2002. Organizational Reconciliation and its Implications for Organizational Decision Support Systems: A Semiotic Approach. *Decision Support Systems,* Volume 33, p. 389 – 398.

Little, G., Chilton, L., Goldman, M. & Miller, R., 2010. *Exploring Iterative and Parallel Human Computation Processes.* Washington DC, ACM.

Little, G., Chilton, L., Goldman, M. & Miller, R., 2010. *Turkit: Human Computation Algorithms on Mechanical Turk.* New York City, NY, Proceedings of the 23rd annucal ACM symposium on User Interface software and technology, pp. 57-66.

Liu, T. X., Yang, J., Adamic, L. A. & Chen, Y., 2014. Crowdsourcing with All-Pay Auctions: A Field Experiment on Taskcn. *Management Science,* 60(8), pp. 2020-2037.

Lorge, I., Fox, D., Davits, J. & Brenner, M., 1958. A Survey of Studies Contrasting the Quality of Group Performance and Individual Performance, 1920-1957". *Psychological Bulletin,* Volume 55, p. 337.

Malhotra, N. K., 1982. Reflections on the Information Overload Paradigm in Consumer Decision Making. *The Journal of Consumer Research,* 10(4), pp. 436-440.

Malone, T., Laubacher, R. & Dellarocas, C., 2010. The Collective Intelligence Genome. *MIT Sloan Management Review,* 51(3), pp. 21 - 31.

Malone, T. et al., 2017. *Putting the Pieces Back Together Again: Contest Webs for Large-Scale Problem Solving.* Portland, OR, Proceedings of the ACM Conference on Computer-Supported Cooperative Work and Social Computing.

Malone, T. & Smith, S., 1988. Modeling the Performance of Organizational Structures. *Operations Research,* 36(3), pp. 421-436.

Malone, T. W., Laubacher, R. & Dellarocas, C., 2009. *Harnessing Crowds: Mapping the Genome of Collective Intelligence,* Cambridge, MA: MIT Sloan Research Paper No. 4732-09.

Maniquet, F. & Mangin, P., 2011. *Approval Voting and Arrow's Impossibility Theorem,* Warwick: University of Warwick.

Mason, W. & Watts, D. J., 2009. Financial Incentives and the Performance of Crowds. *SIGKDD Explorations,* 11(2), pp. 100-108.

McComb, C., Goucher-Lambert, K. & Cagan, J., 2015. *Fairness and Manipulation: An Emprical Study of Arrow's Impossibility Theorem.* Milan, Italy, Proceedings of the 20th International Conference on Engineering Design (ICED15).

Melkonyan, T., 2013. Decentralization, incentive contracts and the effect of distortions in performance measures. *The Manchester School,* 1(1), pp. 1-22.

Miller, M. S., Morningstar, C. & Frantz, B., 2001. *Capability-based Financial Instruments,* Cupertino, CA: ODE.

Miller, M. & Stiegler, M., 2003. The Digital Path: Smart Contracts and the Third World. *Information and Communication. Austrian Perspective on the Internet Economy.*

Mookherjee, D., 2005. Decentralization, Hierarchies and Incentives: A Mechanism Design Perspective. *Economic Letters.*

Mulgan, G., 2006. The Process of Social Innovation. *Innovations,* 1(2), pp. 145-162.

Mullen, B., Johnson, C. & Sales, E., 1991. Productivity loss in brainstorming groups: A meta-analytic integration. *Basic and Applied social Psychology,* Volume 72, pp. 3-23.

Musiani, F., 2013. Governance by Algorithms. *Internet Policy Review,* 2(3), pp. 1-8.

Myers, R., Montgomery, D. & Anderson-Cook, C., 2009. *Response Surface Methodology - Process and Product Optimization Using Designed Experiments.* 3rd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.

Nakamoto, S., 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System.*

Nan, N., 2008. A principal-agent model for incentive design in knowledge sharing. *Journal of Knowledge Management,* 12(3), pp. 101-113.

Nickerson, J. V., Sakamoto, Y. & Yu, L., 2011. *Structures for Creativity: The crowdsourcing of design.* Vancouver, BC, Canada, CHI Workshop on Crowdsourcing and Human Computation: Systems, Studies, and Platforms.

Niederle, M. & Vesterlund, L., 2011. Gender and competition. *Annual Review of Economics,* Volume 3, pp. 601-630.

Niederman, F. & March, S., 2012. Design Science and the Accumulation of Knowledge in the Information Systems Discipline. *ACM Transactions on Management Information Systems,* 3(1), pp. 1-15.

Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V. V., 2007. *Algorithmic Game Theory.* 1st ed. New York: Cambridge University Press.

Norta, A., 2015. *Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations.* Singapore, Springer, pp. 3-17.

Norta, A., 2017. *Designing a Smart-Contract Application Layer for Transacting Decentralized Autonomous Organization.* Singapore, Springer, pp. 595-604.

Olafsson, J., 2011. An Experiment in Iceland: Crowdsourcing a Constitution? *Working Paper.*

Orrison, R., Wilson, B. J. & Zillante, A., 2004. Multiperson tournaments: an experimental examination. *Management Science,* Volume 50, pp. 268-279.

Page, S. E., 2012. A Complexity Perspective on Institutional Design. *Politics, Philosophy & Economics,* II(1), pp. 5-25.

Paulus, P., Kohn, N., Arditti, L. & Korde, R., 2013. Understanding the Group Size Effect in Electronic Brainstorming. *Small Group Research,* 44(3), pp. 332-352.

Pederson, J. et al., 2013. *Conceptual Foundations of Crowdsourcing: A Review of IS Research.* Hawaii, 46th International Conference on System Sciences.

Peffers, K., Tuunanen, T., Rothenberger, M. & Chatterjee, S., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems,* 24(3), pp. 45-77.

Prestopnik, N., 2010. Theory, Design and Evaluation - (Don't Just) Pick any Two. *Transactions on Human-Computer Interaction,* 2(4), pp. 167-177.

Prestopnik, N. & Crowston, K., 2012. *Exploring Collective Intelligence Games with Design Science: A Citizen Science Design Case.* Sanibel Island, Florida, ACM Group Conference.

Pullinger, K., 2007. *Living with A Million Penguins: inside the wiki-novel.* Available at: https://www.theguardian.com/books/booksblog/2007/mar/12/livingwithamillionpenguins

Purao, S., 2002. *Design Research in the Technology of Information Systems: Truth or Dare,* Atlanta: Department of Computer Information Systems, Georgia State University.

Radin, M., 2000. Humans, Computers, and Binding Commitment. *Indiana Law Journal,* 75(4), pp. 1125-1162.

Radin, M., 2004. Regulation by Contract, Regulation by Machine. *Journal of Institutional and Theoretical Economics,* Volume 160, pp. 1-15.

Raykar, V. C. et al., 2010. Learning from crowds. *Journal of Machine Learning Research,* 11(7), pp. 1297-1322.

Ren, J., 2011. *Exploring the Process of Web-based Crowdsourcing Innovation.* Detroit, Michigan, AMCIS Proceedings.

Ren, J. et al., 2014. Increasing the Crowd's Capacity to Create: How Alternative Generation Affects the Diversity, Relevance and Effectiveness of Generated Ads. *Decision Support Systems,* 1(1), pp. 1-12.

Ren, J., Ozturk, P. & Yeoh, W., 2017. Online Crowdsourcing Campaigns: Bottom-Up versus Top-Down Process Model. *Journal of Computational Information Systems,* 1(1), pp. 1-12.

Rosen, S., 1986. Prizes and incentive in elimination tournaments. *American Economic Review,* Volume 76, pp. 701-715.

Sakamoto, Y. & Bao, J., 2011. *Testing Tournament Selection in Creative Problem Solving Using Crowds.* Shanghai, 32nd International Conference on Information Systems.

Scott, M. & Antonsson, E., 1999. Arrow's Theorem and Engineering Design Decision Making. *Research in Engineering Design,* 11(1), pp. 218-228.

Shaw, A., Horton, J. & Chen, D., 2011. *Designing Incentives for Inexpert Human Raters.* Hangzhou, China, CSCW.

Shoham, Y. & Leyton-Brown, K., 2010. *MULTIAGENT SYSTEMS: Algorithmic, Game-Theoretic, and Logical Foundations.* 1.1 ed. Cambridge: http://www.masfoundations.org.

Singla, A. & Krause, A., 2013. *Truthful Incentives in Crowdsourcing Tasks using Regret Minimization Mechanisms.* Rio de Janeiro, Brazil, World Wide Web Conference Committee (IW3C2).

Surowiecki, J., 2004. *The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations.* New York, Doubleday.

Szabo, N., 1997. Formalizing and Securing Relationships on Public Networks. *First Monday,* 2(9).

Taylor, C., 1995. Digging for Golden Carrots: an Analysis of Research Tournaments. *The American Economic Review,* 85(4), pp. 872-890.

Thuan, N. H., Antunes, P. & Johnstone, D., 2017. A Process Model for Establishing Business Process Crowdsourcing. *Australasian Journal of Information Systems,* 21(1), pp. 1-21.

Valentine, M. A. et al., 2017. *Flash Organizations: Crowdsourcing Complex Work by Structuring Crowds as Organization.* Denver, ACM CHI.

Vincent, T. L., 1983. Game Theory as a Design Tool. *Journal of Mechanisms, Transmissions, and Automation in Design,* Volume 105, pp. 165-170.

Wightman, D., 2010. *Crowdsourcing Human-Based Computation.* Reykjavik, Iceland, Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp. 551-560.

Wood, G., 2015. *Ethereum: A Secured Decentralised Generalised Transaction Ledger,* s.l.: Ethereum.

Wu, H., Corney, J. & Grant, M., 2015. An evaluation methodology for crowdsourced design. *Advanced Engineering Informatics,* Volume 29, pp. 775-786.

Yu, L., 2011. *Crowd Creativity through Combination.* Atlanta, Georgia, USA, C&C ACM 978-1-4503-0820.

Yu, L. & Nickerson, J., 2011. *Cooks or Cobblers? Crowd Creativity through Combination.* Vancouver, BC, CHI.

Yu, L. & Nickerson, J., 2011. *Generating Creative Ideas through Crowds: An Experimental Study of Combination.* Shanghai, 32nd International Conference on Information Systems.

Yu, L. & Nickerson, J., 2013. An Internet Scale Idea Generation System. *ACM Transactions on Interactive Intelligent Systems,* 3(1), pp. 1-24.

Yu, L. & Sakamoto, Y., 2011. Features Selection in Crowd Creativity. *FAC, HCII, LNAI,* pp. 383-392.

Zhang, X., Venkatesh, V. & Brown, S. A., 2011. Designing Collaborative Systems to Enhance Team Performance. *Journal of the Association for Information Systems,* 12(8), pp. 556-584.