The Architecture of Economic Organization: Toward a General Framework*

Michael Christensen[†] Thorbjørn Knudsen[‡]

January 30, 2002

Abstract

This paper extends previous project-selection models to allow the study of a wide variety of explicit structures including any number of agents. The previous models are limited to the study of a single family of simple structures, including the 2-member hierarchy and polyarchy. These models are included as simple special cases. A number of ambiguities (regarding internal structure, causality, time ordering and cost model) in the previous project-selection models are identified and clarified, a general framework encompassing essentially all intuitively sound architectures of economic organizations is outlined, and some of the simple models from the different families of architectures are exemplified. Methods are introduced for the purpose of extracting information about screening standards and performance.

1 Motivation

The way information traverses an economic system will influence the errors of judgment that are expressed at the system level. This follows from the basic insight emerging from Sah & Stiglitz's models of organizational architecture [1], [2]. According to these models, the structure of a decision-making economic system, its architecture (in terms of a description of the constituent components and how they are connected), affects the emergence of error at the collective level. When individual agents experience limits to rationality, they make errors of judgment. In economic systems, agents with limited rationality reject projects with positive value (Type-I error) and accept projects with negative value (Type-II error). It is therefore interesting to ask whether some economic systems help individuals to collectively deliver better or worse judgments, and why?

Providing answers to this question has obvious implications for the design of economic systems, but also promises to fill a void in studies of important phenomena that are intimately related to the architecture of economic organization, such as the design of marketing or audit teams, vertical integration of business firms, and the possible emergence of new organizational forms in response to electronic communication networks.

 We are grateful to John A. Christensen and James G. March for comments and inspiration. Any remaining errors were produced without any help.

†University of Southern Denmark, Department of Physics.

‡Corresponding author (e-mail: tok@sam.sdu.dk). University of Southern Denmark, Department of Marketing.

In theories of accounting, the individual agent is usually portrayed as fallible. When an audit team or a self-evaluation team is involved, how is the flow of information among team members best organized to reduce team-level error? Similarly, it would be interesting to know how a marketing team is best organized to minimize error in the screening of market information.

The constituent components of an economic organization such as the firm defines an internal structure that determines how information traverses this organization. Using the firm as illustration and assuming individual error in judgment, alternative internal structures can be associated with alternative firm-level differences in judgment of profitable ideas, including the possible hazards of trade. Since most theories of economic organization tend to ignore the profit contribution of alternative internal structures with respect to error in judgment at the level of the organization, it is possible that their predictions must be adapted.[1] For example, when Transaction Cost Economics predicts vertical integration of two independent economic organizations, it is likely that information traverses the new merged entity in a different way. Perhaps the new merged entity will adopt a structure that increases error in judgment, implying a loss that exceeds the gains from the reduction in transaction costs?

Generally, the design of alternative internal structures of economic organization is constrained by the costs of organization. If the claims are true that the costs of processing information are decreasing because of the spread of electronic communication networks, the constraints on the feasible internal structures of economic organization may have loosened during the last decades. Will this lead to the emergence of new forms of internal structures that further decrease the firm-level error in judgment?

The aim of the present paper is to provide the outline of a general theory of the architecture of economic organization that may be useful in addressing such questions. The paper is organized as follows. First, section 2 describes Sah & Stiglitz's project-selection model including an explicit structure of two economic entities. We point out a number of ambiguities that must be considered in order to generalize this model to an explicit internal structure including any number of agents. These ambiguities include the correspondence of alternative architectures to known forms of economic organization (firms, hierarchies and hybrids) as well as a number of technical issues. Section 3 outlines a general framework that can be used to model n-member structures and provides clarification of the mentioned ambiguities. This general framework includes the Sah & Stiglitz model as a special case. Section 4 illustrates some of the simple models from the different families of architectures and analyzes them using the newly introduced methods. Analytical results for two particularly simple models are provided in section 5. Finally, section 6 concludes by considering further developments and implications for the study of phenomena that are intimately related to the architecture of economic organization.

2 Previous Models of 2-Member Structures

The purpose of this section is to describe the model of organizational architecture provided by Sah & Stiglitz [1, 2].

[1] The notable exception is Marschak and Radner [3].

The first project-selection model of Sah & Stiglitz [1] is limited to two explicit structures each consisting of only two economic agents, yet it captures many of the essential properties that a correct model must display. The extended model of Sah & Stiglitz [2] is both a generalization and a unification of the previous as it describes a single family of organizational structures to which the above two-member structures belong. Unfortunately, a number of implicit assumptions must be clarified (internal structure, causality, time ordering and cost model) in order to extend (or even complete) their model for n-member architectures.

2.1 The Context of the Project-Selection Model

The key ingredient in the context presented by Sah & Stiglitz is the *agent* along with the realization that agents are fallible when they evaluate the profit potential of prospects. Uncertainty is added on behalf of the finite processing capabilities of the agent. This finiteness is an expression of limits as a characteristic of the decision-maker, to be understood in any possible way, ranging from the failure to obtain, assimilate, interpret and evaluate information relevant to a project to a state of profound uncertainty. Because agents with limited rationality sometimes reject projects with positive value and accept projects with negative value, the sole source of uncertainty is the limits that characterize the agent, expressed as an imperfect ability to screen information. If the agent were capable of perfect screening, profits would be maximized for any distribution of project values. Thus there is no need for project appearances beyond the control of the agent or added irrational human behavior. Consistent with the line of thought advanced by Simon (see [4, 5, 6, 7]), it is the limits of the agent, not a part of his technological environment, that is the source of uncertainty. This is the strength of the model.

Following the ideas and terminology of reference [1], the basic process of making a decision goes loosely as follows. A decision-making structure will (repeatedly) be confronted with a *project* drawn from an *initial portfolio*. The project enters the structure through one of its agents and traverses the structure until it is either rejected or accepted ultimately. Rejection means that the project is terminated and no profit can be earned from it. Accept means that the project is realized, symbolized by storing it in a *final portfolio*, and profit is earned according to the quality of the project. In both cases a cost is paid for making the decision. If a project capable of producing any income, thereby reducing the costs and perhaps even produce profit, is rejected, then the organization made an error, denoted a *Type-I error*. If, on the other hand, the organization accepts a project incapable of producing a positive income, it is said to have made a *Type-II error*. In both of these cases, the decision was a *failure*, and in all other cases it was a *success*. The ultimate fate of the project depends on how the agents are interconnected, thereby motivating the study of different organizational architectures.

At this point the ambiguity arises that the Sah & Stiglitz model omits an explicit cost model. The implicit assumption to be inferred from their model is that the cost of making a decision is already included in (or rather extracted from) the profit that can be earned from a given project. This in turn implies, that all decision-making structures have equivalent costs regardless of their size. In the case of the 2-member structure this is reasonable for some (e.g. identical costs for all members) but not all cost models (e.g. identical costs for the members involved in a decision). In a scenario allowing n-member structures of different sizes, this assumption is clearly unsatisfying. An extension of the Sah & Stiglitz model requires two modifications. First, the cost of making a decision must be excluded from project values and accounted for separately and, second, an explicit cost model must be added[2].

Sah & Stiglitz recognized that since profit is the dominant, if not the only, indicator of success in realistic systems, performance should be measured by considering the expected profit of an organization. In this respect error rates are not accurate measures of success, as they are weighted differently by different cost models and project distributions.

2.2 Hierarchies, Polyarchies and Committees

The two basic architectures, the two-member hierarchy and polyarchy, are the simplest non-trivial structures. Due to the limited possibilities for composing intuitively sound architectures, these are fairly representative of two-agent structures in general. Starting out with two nodes in the graph, two obvious (non-cyclic) configurations stand out (from a number of structures to be labeled as illegal in section 3), namely the two-member hierarchy and polyarchy.

Figure 1 below provides a static overview of the graph representing the hierarchy of Sah & Stiglitz, and its environment, the initial portfolio I and the final portfolio F in which projects accepted by the hierarchy are stored. The "grounded" symbol marks a termination node T, the store where the rejected projects are dumped forever. The hierarchy represents a serial processing as a project traverses the structure and it is straight forward to generalize it to n-member hierarchies simply by adding nodes to the sequence between I and F.

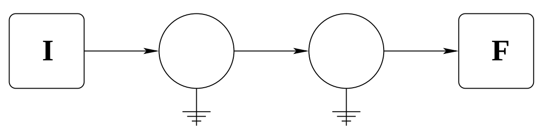

Figure 1: *The two-member hierarchy. In order for the graph to accept a project, both agents must accept the project individually. Or, the other way around, each individual has the ability to terminate the project.*

In contrast to the simple hierarchy, the polyarchy, shown in figure 2 below, has a more specialized behavior and is far more difficult to generalize in an unambiguous way. Since polyarchies represent a parallel processing, the agents must have the same level of competence. This can be modeled in various ways. The agents may incorporate a level of competence in their internal state such that all agents of matching competence belongs to the same polyarchy. Alternatively, and in order to keep the simple picture of agents and channels of communication with no internal structure, three types of edges can be used – accept, standard reject and polyarchy reject. Such

[2] Sah & Stiglitz do not provide an explicit cost model, but in their discussion, mention, among other problems, the possible differences in costs associated with committees of different consensus [2]. Later, Koh raised the issue of fixed versus variable evaluation costs, concluding that the optimal screening standard for the two basic structures of Sah & Stiglitz depends on the cost specification [8]. Koh, however, omitted an explicit cost model and did not extend the basic structures of Sah & Stiglitz.

a setup allows for the type of polyarchy of Sah & Stiglitz, but additional rules of traversal, the dynamics of the architecture, must be supplied to ensure the polyarchy behavior.

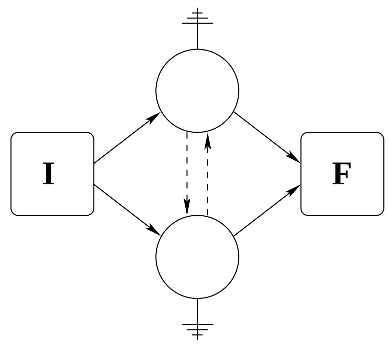

Figure 2: *The two-member polyarchy of Sah & Stiglitz. Here, the project is sent to any of the agents at random. The individual can accept the project on behalf of the graph, but in order to get an overall reject, both agents must reject. This behavior cannot be observed from the graph topology alone but is partly implemented in the rules that constrain the traversal: Enter dynamics.*

Problems arise when considering polyarchies of 3 or more members. It is not obvious how the agents are interconnected inside the polyarchy neither in what temporal order they evaluate a project. Here it is important to note that these properties are related to the architecture as a whole and cannot (in the general case) be extracted from the topology of the graph alone. In order to extend the Sah & Stiglitz polyarchy to include 3 or more members, added specification must be provided, and the actual choice of specification reflects, among other properties, the internal structure and the causal type of the graph. In this way, the space of explicit models is naturally divided according to the supplied assumptions.

Sah & Stiglitz further defined a unifying sub-structure called a *committee* of n members and consensus k. This sub-structure was constructed by picking a polyarchical structure and supplying a dynamic rule stating that the organization should accept only if k or more agents evaluate the project positively.

The invention of the n-member committee of consensus k was both a generalization of the two basic structures and a unification of the two into a common framework. An infinite, but countable set of structures had been made available. However, the generalization from the basic two-member parallel organization is inherently ambiguous unless additional assumptions on internal structure or time sections are supplied. Furthermore, when enlarging the set of structures considered (hybrids of hierarchies and polyarchies, nested/cyclic structures etc.), the committee described above is obviously non-representative for the large number of possible configurations.

Summing up the ground covered so far, Sah & Stiglitz provided a modeling context including the description of projects, agents and basic dynamics, and then advanced to describe the two most obvious 2-member structures, the 2-agent hierarchy (two elements in series) and polyarchy (two elements in parallel) [1]. Later, they generalized and unified these two basic substructures in the form of n-member committees of consensus k [2].

Interestingly, Marschak & Radner [3] (ch. 8) had previously considered the 2-member serial and parallel structures that Sah & Stiglitz are concerned with as well as possible extensions of these structures. Ignoring this source, Sah & Stiglitz did not offer a comparison, but an important difference between the two formulations is in how they model and conceptualize uncertainty. In the Sah & Stiglitz model, uncertainty arises due to limitations of the decision-maker whereas Marschak & Radner's formulation ascribes uncertainty to factors not predictable by the decision-maker. On the more philosophical level, these unpredictable factors of the environment can be seen as a fingerprint of fundamental chaos (much to the dismay of the all-knowing and infinitely wise agents), contrasting a world that is after all, despite its strong non-linearity (blurring the minds of the fallible, finite agents), deterministic. Even if Marschak & Radner opened the possibility of modeling extended structures, we prefer to expand the formulation of Sah & Stiglitz[3] because it solely ascribes uncertainty to the decision-maker, a virtue emphasized by Simon [6].

We have pointed out, however, that further consideration of internal structure, causality, time ordering and cost model is needed in order to extend the Sah & Stiglitz model to an explicit n-member structure. A further ambiguity arises in the correspondence of architectures to known forms of economic organization. Sah & Stiglitz argue that the two basic structures, the hierarchy and the polyarchy, are suggestive of a market-oriented economy and a bureaucracy oriented economy, respectively. While this may be a correct description observed from a large distance, we believe that a better correspondence to actual firms and markets can be obtained by including both of the two basic structures in a n-member structure (a firm may obviously include the polyarchy structure at different hierarchical levels and industrial markets can include obvious hierarchical features). The point is that a particular form of economic organization cannot be inferred solely by the topology of a decision-making structure. Further specification, reflecting the properties of the specific type of organization targeted for modeling, is needed.

3 Modeling n-Member Structures

The purpose of this section is to outline a general framework that can be used to model n-member structures of organizational architecture. Due to the ambiguities identified in the previous section, the simple parallel/serial sub-structure picture must be dissolved in order to allow for more general graphs or at least for some sort of hybridization. The main question to be addressed here is: *Which types of architectures are to be considered as legal (in the sense of realism) implementations of decision-making organizations?*

Realism, especially the fact that any resource is available in finite quantities only, naturally restricts the set of possible models. Still, a number of questions regarding the properties and behavior of the various components of the model will stand unanswered. This is *not* a weakness of the framework. On the contrary, it is exactly what makes it so versatile. The loose ends are

[3]Eventually, the two formulations may prove to be similar in the sense that one can be transformed into the other.

for the model-builder to tie up, since they are strongly related to the interpretation of the model. It leaves room for many different assumptions (yielding distinct families of models), making the framework easily extendible and applicable to a wide range of scenarios. In principle, the set of possible models are limited only by our imagination, though it should be remembered, that the more detailed information a model contains, the fewer are the instances to which it can be applied. Due to large number of families of models, only the simplest ones, yet including the committees of Sah & Stiglitz, will be treated in detail here.

The remaining content of this section is as follows. First an overview of the graph theoretical aspects of the modeling framework is given. The mathematical description of the system to be modeled has two intimately connected facets, a physical structure and a set of dynamic rules. These two facets must always be compatible and support/reflect the phenomena to be studied. Thus, section 3.2 turn the attention toward structural components and dynamic rules. Various possible large-scale properties and the restrictions/consequences implied by their assumptions are displayed. Finally, section 3.4 addresses the question of how to measure performance. Remarks on details of implementation are included.

3.1 An Overview

The architecture of an organization or an economic system can be represented by a *graph* consisting of *nodes* and *edges* [9]. In such a representation each node corresponds to an individual or more generally to the basic building block, the simplest constituent, the atom, here denoted an *agent*. These agents can have very different properties depending on the specifics of the model. Similarly, each edge represents a channel of communication to another agent or more generally the passing of information or control to another unit, and it is here denoted a *choice* because agents must pick only one during the individual decision process. These choices will also be equipped with certain characteristics to incorporate realistic features into the model system.

The basic process of making a decision was described above in section 2.1. The project quality is determined by the initial portfolio, and detailed cost considerations may be an important part of a model too, since they are closely related to the measuring of performance, and since they represent assumptions on time scales, parallel processing capabilities and employment strategies of agents (see the following sections).

In terms of the graph representing the structure, an *input* (I), an *output* (F) and a *termination* (T) node are introduced. These external nodes are of a different type than the internal nodes representing agents, as they represent the environment or store historical information about the system. The projects are drawn from the initial portfolio and may be represented by a d dimensional vector, $\vec{x}$, encoding the relevant information available to the agents and to the organization as an entity. It may even store dynamic information on the decision process itself. The simplest project description is just the real number ($d = 1$) describing the income (excluding the cost of the decision process itself) produced by the project in case of acceptance.

The decision-making process is realized by traversing the graph from the input node to either the output or to the termination node. Such a connected *path* of subsequent choices is simply denoted a *decision*. The overall picture is illustrated in figure 3, where the edges must be interpreted loosely, because a decision-making structure can have several entry points as well as more than one path may lead to acceptance or rejection.

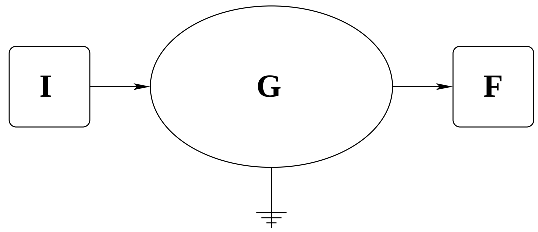

Figure 3: *A static overview of the graph* **G**, *representing a decision-making structure, and its environment, the initial portfolio I and the final portfolio F. The "grounded" symbol marks the termination node T.*

3.2 Graph Components and Dynamics

The basic constituents of the framework will now be described along with their collective behavior. Invariants for the graphs are presented and key questions to answer when building models are identified.

3.2.1 The Cost Model

The explicit cost of making a decision is important to include because it facilitates comparison of graphs across sizes and families, and because it sets a scale on which to measure profit. It can be modeled in various ways as the details may depend on the involved time scales as well as the ability to process several projects at a time and the job descriptions of the agents in the organization. It is equally important to realize that this cost is *not* including construction, production or other realization costs as these are included into the project value, $\vec{x}$. The architecture is viewed as the brain of a larger and more directly productive apparatus, and as such it has its own internal accounts (a cost is paid for evaluating a project even if the production apparatus is never started).

Three basic models are proposed here:

1. Large-scale operation costs (fixed per time unit).

2. Employment costs (scales linearly with the number of agents).

3. Free-lance costs (scales linearly with the number of active agents).

Starting from the bottom of the list, the free-lance cost model reflects small organizations that demand very specific and rapidly varying (over time) competences from its employees. Therefore the best strategy is to recruit people only when needed and contract for one project at a time. Note, that this cost model requires polyarchical structures to have a specific time ordering, in order to know who evaluated the project and who not before acceptance was reached.

The second model, the employment cost, simulates organizations with a fixed staff, either because the organization is big enough to cover all fields of competence needed or, reversely, because there is only a very few fields of competence in the niche. This cost model can be used to effectively model parallel processing of projects in time, since generally not all agents are active during a single decision.

Finally, with the large-scale operation cost the other extreme has been covered, that is, the situation where the organization is large and where several similar projects can be treated simultaneously implying relatively low overall expenses for each project. Again, this model can be used to effectively model parallel processing, and the incoming projects should now be viewed as multi-project cases[4].

3.2.2 The Initial Portfolio

A project may be represented by a d dimensional vector, $\vec{x}$, encoding the relevant information available to the agents and to the organization as an entity. It may also leave room for internal evaluation data as long as an initial default state is uniquely given. The quality of a project is the potential for profit calculated by some scalar field

$$\mathcal{P} : \Re^d \to \Re \tag{1}$$

This potential profit includes all costs of production, effectuation or implementation, but excludes the cost of making the decision whether to realize the project or not, because this cost may depend on the architecture or even the specifics of the decision process. In the simplest case, the project is simply represented by the single scalar value, x, measuring its potential for profit.

The initial portfolio represents any part of the external environment that has nothing to do with the costs of making the decision itself. Thus it reflects the information, resources, available technology, political circumstances and market dynamics. It, and not the organization itself, defines the (available) projects, whose values are determined by a given income distribution G.

3.2.3 Agents and Choices

An agent, represented by a node in the graph, is the basic unit of the decision-making structure. Moreover, the single agent organization is the simplest, non-trivial[5] structure that can be formed. The task of the agent is to receive projects, evaluate or "screen" them and, according to the screening, dispatch the projects along the appropriate channels of communication available

[4]The approach of projects consisting of similar sub-projects in some limits leads to Gaussian income distributions, whence it biases the project distribution of the initial portfolio.

[5]Structures of no agents are a simple matter of definition of behavior, unrealistic and will not be considered as legal structures here.

to the agent. The actual channel used corresponds to a single edge in the graph and represents the agents choice of what to do with the project. The evaluation of the project by the individual agent is described by a mapping from the project space into the closed real unit interval representing a Bayesian probability

$$f \,:\, \Re^d \;\to\; [0,1] \tag{2}$$

This *agent screening function* can be interpreted in several ways, depending on the imposed dynamics. The simplest choice of a screening function is a hard-coded function representing the static properties of the agent. But as internal state and memory of agents is incorporated, reflecting experience and the ability to learn, it may not even be a true function (unless the agent, the history and even the graph are accepted as arguments as well). Nevertheless, the outcome of the screening should always be accept or reject, possibly followed by a transformation

$$\mathcal{T} \,:\, \Re^d \;\to\; \Re^d \tag{3}$$

of the project $\vec{x}$. What happens to the project next is determined by the topology of the graph and the specifics of the dynamics.

The project transformation of equation (3) represents the process of adding or distorting information during the process of communication or even the direct manipulation of the initial project, whether this is an attempt to improve the project or simply an update of internal evaluation data. A transformation with the property

$$\mathcal{P}(\mathcal{T}\vec{x}) \;=\; \mathcal{P}(\vec{x}) \tag{4}$$

is said to conserve the quality of the project, whereas a transformation with the property

$$f(\mathcal{T}\vec{x}) \;=\; f(\vec{x}) \tag{5}$$

is said to conserve the appearance of the project. The simplest transformation is the identity transformation.

Similar to the agents (nodes), the choices (edges) can have various properties, both statical and dynamic. Whereas many models can be build with only one type of agent, it is often easiest to use more than one type of edges in order to get rid of eventual internal states and dynamical variables. Acceptance and rejection choices may be enough, but as illustrated in section 2.2, committees often require an additional polyarchy rejection edge.

3.2.4 Guidelines

Constructing a model for a class of economic systems, the model should obviously allow for the properties of interest. Furthermore, it should be realistic, meaning that only intuitively sound architectures should be allowed, and that any acquired knowledge of real life systems should be built in by making the right assumptions. The hard part is to ensure that the structure and the dynamics are compatible and combined capture exactly the phenomena of interest.

Invariants

Realism and the overall desire to use graphs as representatives for decision-making models pin down the following invariants:

1. The graph is directed, finite and connected.
2. There exists one external node I having only out-bound acceptance edges to internal nodes.
3. There exists one external node F having only in-bound acceptance edges from internal nodes.
4. There exists one external node T having only in-bound rejection edges from internal nodes.
5. Each internal node must have at least one acceptance edge.
6. Each internal node must have at least one rejection edge.
7. The external node F or T must be reachable from any internal node.

Additional structural criteria may be applied to further restrict the type of model being considered and in order to exclude non-intuitive constructs. For example, the concept of a *loop*, defined as a node with an edge to itself, seems unrealistic unless the scarce resource is time (the agent needs more time to process the project). Another possibility is to replace "or" in invariant 7 with "and" and further disallow nodes whose edges (both accept and reject) all point to the same other node, thereby rendering the first one superfluous and possibly (depending on the cost model) an unnecessary cost.

Dynamics

While the above invariants mainly concern structure, it is important to extend the finiteness requirement to the dynamic rules as well. For a decision to have any value, it must be made sufficiently fast. Cyclic architectures may result in infinite loops if the dynamics allow it. Such constructions are obviously in-feasible as it requires a finite time for each agent to evaluate a project[6]. Additional dynamic rules can be imposed to facilitate very specific phenomena, or simply to allow for structures like polyarchies and committees.

Questions

Causality:

Should the graph be stochastic or deterministic? Some agents might have several choices available in case of acceptance or rejection. If this is the case, and the actual edge chosen is picked at random, the graph is *stochastic*. Since agents are picked by random, they seem interchangeable and whence this type of graph promotes symmetry, though different levels of abstraction could be introduced to break some of the symmetry (see section 4), thereby allowing the stochastic sub-structures to be nested.

[6]Finiteness of cyclic graphs can be achieved by counting the number of evaluations by agents in total, by polyarchies in total or by individual agents. Whenever a finite maximal count is reached, some default behavior leading to ultimate acceptance or rejection should be activated

On the other hand, if only one edge is available at a time, the graph is said to be *deterministic.* Note, that the deterministic graphs can be identified, not as the graphs owing only one acceptance and one rejection edge per node, but as graphs where the dynamics allow exactly one of each at every evaluation. Thus, they have a *dynamical out-degree* of unity when counting edges of each type separately, a weaker requirement than simply having static/structural out-degree of unity. A further restriction of the deterministic graphs along these lines is to make it *inward deterministic* as well, meaning that not only the successor in the decision process is fixed, the predecessor is fixed as well. Analogously, this amounts to having a *dynamical in-degree* of unity. The in-degree must be counted separately for the various edges unless a form of *strict* inward determinism is assumed, thereby disallowing polyarchies with more than one entry point[7].

Time ordering:
Do the agents of sub-structures evaluate the project simultaneously or one by one? Since some cost models need to know exactly who evaluated a project, it might be necessary to determine whether all members of a polyarchical sub-structure evaluate projects at the same time (*simultaneous*) or whether it is evaluated by the agents one at a time (*incremental*) as it is passed along inside the sub-structure.

Feedback:
Is it possible to get a re-evaluation? Some organizations have build-in feedback mechanisms (a project can be send backwards in the structure along with additional information in order to get a re-evaluation). Even though the graph structure might be cyclic, this alone is not a fingerprint of a system with feedback. The polyarchy is the obvious counter-example; even though the physical structure has closed paths of edges, the dynamic rules disallow the same agent to evaluate the project more than once. To distinguish the feedback property from structural cycles, graphs that allow feedback are said to be *dynamically cyclic* (henceforth just *cyclic*), which is a stronger requirement than just being structurally/statically cyclic.

3.3 Committees Revisited

Before advancing arbitrarily large structures, the n-member committee of consensus k can be reconsidered in the light of the guidelines presented above. This is illustrating because it completes (or at least makes the implicit assumptions explicit in) the model of Sah & Stiglitz. Moreover, these structures at the intermediate level can be used as sub-structures or building blocks for larger graphs. The committee model of Sah & Stiglitz was described in section 2.2. Based on our guidelines, we can now discuss the various shades of the committee.

First of all, the cost model is absent and must be chosen. If a free-lance cost model is picked, the time ordering must be incremental.

Moreover, a stochastic graph that is completely structure-less internally were picked by the authors in references [1, 2]. Taking this symmetry to the extreme, all the agents of the committee share all inbound and outbound edges, and they are all symmetrically connected internally. The dynamics simply pick the next to evaluate the project at random amongst those

[7]The strict version of outward determinism results in trivial graphs that either accept or reject every project.

who haven't yet seen the project. Intuitively, this approach also promotes simultaneous time ordering, since all the agents are equal in all respects. For the committee itself, the choice of stochastic causality may seem somewhat unimportant, but in case the committee is used as a building block for larger graphs, it becomes quite relevant. Since all agents share the outbound edges and due to the internal symmetry, this model cannot describe the concept of *key positions.*

Making the committee deterministic by choice requires the model-builder to provide detailed specification of the internal structure. A simple choice may be to connect all the agents in a ring. The agents need not share inbound or outbound edges, and suddenly it may be of great importance where a polyarchical structure is entered and what exit point is used. This type of committee has a detailed internal structure that promotes incremental time ordering and facilitates the concept of key positions, as the otherwise identical agents acquire different levels of importance due to the different positions they occupy in the structure.

A committee of n identical members and consensus k has an overall accept probability of

$$\mathcal{F}_{\text{committee}}(\alpha; n, k) \;=\; \sum_{i=k}^{n} \binom{n}{k} \alpha^{i} (1-\alpha)^{n-i} \tag{6}$$

where $\alpha = f(x)$ is the agent screening function.

The cost depends on the time ordering. Simultaneous evaluation (preferred by the stochastic version) has a cost corresponding to its n members, whereas incremental evaluation gives rise to a distribution over the actual number of evaluations needed. In the case of acceptance the probability that $k \leq i \leq n$ evaluations were needed is

$$\text{Prob}(\text{accept}, i, \alpha; n, k) \;=\; \binom{i-1}{k-1} \alpha^{k} (1-\alpha)^{i-k} \tag{7}$$

and in the face of rejection

$$\text{Prob}(\text{reject}, i, \alpha; n, k) \;=\; \binom{i-1}{n-k} \alpha^{i-(n-k+1)} (1-\alpha)^{n-k+1} \tag{8}$$

is the probability that $n-k+1 \leq i \leq n$ evaluations were needed. Simply multiply these with their respective cost (proportional to i) and sum it all up[8].

In conclusion, to complete the n-member committee of consensus k within this framework, a cost model (possibly along with a time ordering) and an internal structure (dictating the causality) must be decided upon.

[8] The proof of the above distributions over the number of evaluations needed is straight forward and therefore not included here. The summation over i in equation (7) (equation (8)) reproduces the overall probability for acceptance (rejection).

3.4 Performance and Implementation

The goal of building models for various systems is to test which architecture performs the best[9]. Profit is the fuel for these systems and is therefore a good indicator of performance. Given a graph, all the various decisions leading to acceptance of a project $\vec{x}$, along with its probability and the combined project transformation $\mathcal{T}_i$, can be enumerated. These are denoted

$$\mathcal{A} \;=\; \{(q_i(\vec{x})\,,\,\mathcal{T}_i)\}_{i\in I_a} \tag{9}$$

where $q_i(\vec{x})$ is the probability that decision i was picked. The cost can be calculated from the transformed project[10] as $C(\mathcal{T}_i\vec{x})$. Similarly, the decisions leading to rejection is denoted

$$\mathcal{R} \;=\; \{(q_i(\vec{x})\,,\,\mathcal{T}_i)\}_{i\in I_r} \tag{10}$$

Summing up the accept probabilities for each individual path through the graph, the overall graph screening function becomes

$$q(\vec{x}) \;=\; \sum_{i\in I_a} q_i(\vec{x}) \tag{11}$$

Using the above enumeration, the expected value of any quantity ϕ, depending on the project distribution G, and the outcome of the decision, is

$$E[\phi] \;=\; \int_{\Re^d} G(\vec{x}) \left(\sum_{i\in I_a} q_i(\vec{x}) \;\; \phi(\mathcal{T}_i\vec{x})|_{\text{accept}} + \sum_{i\in I_r} q_i(\vec{x}) \;\; \phi(\mathcal{T}_i\vec{x})|_{\text{reject}} \right) d\vec{x} \tag{12}$$

The profit function

$$\phi_{\text{profit}}(\vec{x}) \;=\; \begin{cases} \mathcal{P}(\vec{x}) - C(\vec{x}) & \text{for accept} \\ -\,C(\vec{x}) & \text{for reject} \end{cases} \tag{13}$$

is no special in this respect. Neither is the simple constant 1 which serves as a check of the enumeration, as the integral in this case must evaluate to unity. Fortunately, equation (12) simplifies considerably for the most basic models.

3.5 Evaluation of Decision Probability and Cost

The measuring of any quantity requires all the decisions to be enumerated and their probabilities to be calculated. A standard method for achieving this will now be constructed. In the first attempt only graphs employing identical agents with trivial project transformations are

[9] In a given model, the architecture, which performs the best, will only do so within the family of legal architectures under the assumptions made concerning its types of nodes, edges, composition, dynamic rules, cost model and the project distribution. Thus performance comparisons require the specification of quite a few assumptions. It may, for example, very well be impossible to find an optimal architecture without knowing quite accurately how the agents work internally (their screening function).

[10] Here it is assumed that the required internal information (such as the evaluation count) is stored in the project. In the case of stochastic changes to the actual project quality or appearance further integration should be applied accordingly.

considered. Corrections are discussed.

For graphs with identical agents, only one functionality is inherently present, namely that of the agent screening function, f. As the agent screening maps the project, $\vec{x}$, to values between 0 and 1, the graph itself must process these Bayesian probabilities into an overall probability of accept,

$$\mathcal{F} : [0,1] \to [0,1] \tag{14}$$

denoted the *reduced graph screening* function or the *graph screening polynomial*. The entire mapping of projects into ultimate accept probability is just the composition, $\mathcal{F}(f(\vec{x})) = q(\vec{x})$ of the reduced graph and the agent screening, thereby yielding the graph screening of equation (11). Furthermore, as any acceptance will occur with probability $\alpha \equiv f(\vec{x})$ (here α is used as a token of accept from a single agent) and any rejection will occur with probability $1 - \alpha$, the functionality of $\mathcal{F}$ will be that of a polynomial constructed as a sum of products of the above two factors. The minimal number of accepts needed for ultimate acceptance dictates the minimal order of the lowest term, and similarly the largest number of agents on a path dictates the maximal order possible of the highest term and thereby the maximal order possible for the polynomial.

The knowledge of the overall accept probability only provides sufficient description in the simplest (cost) models. In order to calculate free-lance cost or to estimate other quantities that depend on the details of the decision, every single path, whether it leads to acceptance or rejection, must be accounted for. This can be accomplished by traversing the graph recursively, constantly updating some external record of the data collected so far. Finally, the recorded data can be processed, as probabilities, costs and other characteristics may need to be summed. Here is a pseudo-code sketch for the *recursive algorithm*:

```
CollectData(Node n,Project x,Info i,InfoList a,InfoList r)

   for all possible rejection choices c of n:
      if ultimate rejection is reached
         AddToList(r,RejectionUpdate(i))
      else
         CollectData(c,Transform(x),RejectionUpdate(i),a,r)

   for all possible acceptance choices c of n:
      if ultimate acceptance is reached
         AddToList(a,AcceptanceUpdate(i))
      else
         CollectData(c,Transform(x),AcceptanceUpdate(i),a,r)
```

The function `CollectData` is to be called at the initial portfolio (implement some default behavior) and any information of interest must be kept in the object `i` of type `Info` and manipulated by `RejectionUpdate` and `AcceptanceUpdate`. Every time an ultimate acceptance or rejection is reached, the information is stored in a list for later processing (`a` and `r` of type `InfoList`). Suitable implementations must be supplied for the agents, the graph itself, the project and its transformation, but these details depend on the specifics of the model.

If the agents are identical and the transformation $\mathcal{T}$ is conserving the project appearance then the reduced graph screening can be found analytically by a single recursive sweep.

In the case of more than one type of agent, the calculation becomes a little harder. The graph screening is now a multinomial over all the possible types of agents. Fortunately, since the graph is finite, only a finite number of different types of agents will ever be involved and hence only a finite number of tokens α_n representing accept by the various agent types need to be introduced. The multinomial object is harder to maintain than a simple polynomial but it can certainly be done, implying that the functionality of the graph screening can still be found by a single, recursive traversal of the graph. The pseudo-code is easily corrected as the additional passing of the specific agent type to relevant update functions will suffice.

If the agent transformation is conserving project quality as well then the performance (profit) can be measured by a single sweep as well, though in this special case a much faster algorithm (analogous to the backward method illustrated in section 4) should be used.

Even in the general, non-simplified case, the recursive algorithm sketched above can still be used to extract probabilities and information for specific project values. Thus, a vectorized version can be made, calculating high precision numerical results in still only a single sweep.

In conclusion, for any finite graph (in the sense of a finite number of agents *and* evaluations) the graph screening, the profit and any other expectancy quantity can be calculated[11].

4 Examples

The purpose of this section is to illustrate some of the simplest and most consistent models arising from the previous considerations. The focus is on the formulation of the models as well as the basic calculation of the reduced graph screening function needed for performance comparisons. This will be done, not by the recursive algorithm presented in section 3.5, but by a faster method traversing the graph in reverse order.

The selection of examples below constitute some of the simplest models and serve as a stepping stone from committees toward larger and more complicated models. Common to these examples are the following assumptions:

- The agents are identical with no internal state.
- The choices have no dynamic internal state.
- Projects are described simply by income but may leave room for internal information.
- Project quality and appearance cannot be changed by the agents.

[11]Note that the number of possible decisions to be enumerated explodes (sup-exponentially) with the graph size for realistic levels of inter-connectivity, thereby making large graph problems in-feasible in the general case.

Even with all these simplifications, lots of models can be constructed with even more variations, combinations and extensions possible.

4.1 A Stochastic Graph

A simple model for non-cyclic stochastic architectures could be as follows. Allow all finite graph structures compatible with the 7 invariants and the additional requirement that no loops or cycles exist. Only one type of agent with a given screening function f is present, and three types of edges may be used to connect them. An edge representing accept, a standard rejection edge leading to the termination node and a polyarchy rejection edge. The latter is combined with a dynamic rule stating that the polyarchy edges are transitive and symmetric and can only be used if there is an agent left in the sub-structure that hasn't been visited yet during the present traversal. Symmetry is further enhanced by supplying the rule that if more than one accept edge exists, one is picked at random (uniform distribution). The project value is simply the income, x, in the case of ultimate acceptance and this scalar is drawn from a given income distribution, G. This means that the project quality is $\mathcal{P}(x) = x$, and similarly that the transformation $\mathcal{T}$ is the identity operator. Employee costs, $C(n) = \nu n$ with $\nu \geq 0$, are chosen. Evaluations within sub-structures are considered to be simultaneous in the spirit of the internal symmetry.

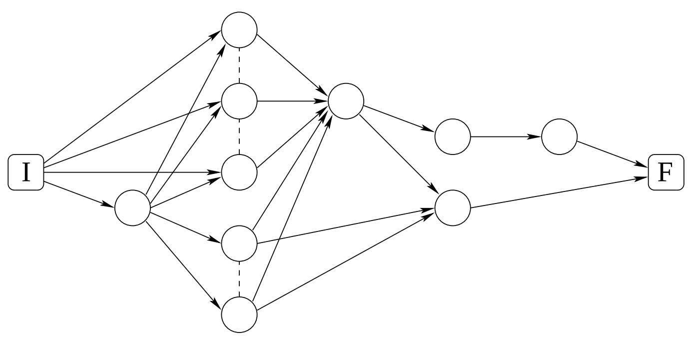

Figure 4: *An example of a stochastic graph with no feedback. The dashed edges symbolize the symmetric and transitive edges that are needed in order to build polyarchies within this type of graphs. All nodes are implicitly connected to the termination node which, for simplicity, is not shown here.*

The local lack of structure within the polyarchies and hierarchies makes it possible to reduce this type of architecture. Introducing a countable infinite set of agent types corresponding to polyarchies (P) and hierarchies (H) of different sizes, the transitive edge can be discarded and a much simpler top-view of the graph can be obtained. The graph screening function (the overall accept probability) and other quantities can easily be calculated using the recursive algorithm. The properties of the newly introduced macro-agents are described in section 3.3.

The most natural extension to this model is to allow multiple rejection edges as well (not only those leading to termination) and similarly supplying a rule stating that the edge chosen should be picked at random with no bias.

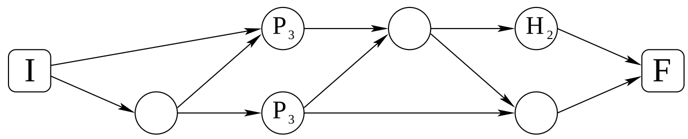

Figure 5: *By considering the hierarchy (H) and polyarchy (P) sub-structures as individual types of macro-agents, the graph in figure 4 can be simplified considerably. A sub-index displays the number of primitive agents in the sub-structure, and a node with no label is just a single agent, here being equivalent to both* H_1 *and* P_1*.*

As Sah & Stiglitz mention [2], the committee sub-structures can be nested and then nested again any given number of times. To allow such deep structures, the different abstraction levels must be separated. This can be done by recursively constructing a larger set of macro-agents from which to build the graph structures. The set of possible agents is still countable and in any practical situation, since the constraint of finiteness of the graph translates into a finiteness of the available family. The properties of these macro-agents can be derived by applying equation (6) recursively as well.

4.2 A Deterministic Graph

A simple model of (outward) deterministic architectures go pretty much along the lines of the previous model – there is no way around the invariants. The main difference is the presence of internal structure of the polyarchies, implying that deterministic graphs cannot be reduced simply by introducing macro-agents.

The same single type of agent and three types of choices as before are present. Polyarchies are build as ring-formed structures with suitable dynamic behavior. No (dynamical) cycles are allowed. Evaluation is considered incremental and free-lance costs are used. In order to keep track of the cost, the project is implemented as a two-dimensional vector, $\vec{x} = (x, k)$ where x is drawn from a given project distribution G specifying the project quality, which can later be read off by $\mathcal{P}((x, k)) = x$, and k is initially set to 0. The quality- and appearance-conserving transformation is non-trivial this time as it must update the internal information storing the number of agents that have evaluated the project, $\mathcal{T}((x, k)) = (x, k + 1)$. Thus at ultimate acceptance or rejection, the cost can be read off as $C((x, k)) = \nu k$ where ν is a positive constant.

The enumeration of decisions through the specific graph shown in figure 6 is easy because of its small size. Only two paths, namely

```
(1)    I - 1 - 3 - 4 - 5 - F
(2)    I - 1 - 2 - 5 - F
```

with probability $q_1(\alpha) = \alpha^4$ and $q_2(\alpha) = \alpha^2(1 - \alpha)$, respectively, lead to ultimate accept. Thus the overall accept probability is just the sum of the two above probabilities. The rejection paths must be enumerated too in order to process the cost correctly. They are

```
(3)    I - 1 - 3 - T
```

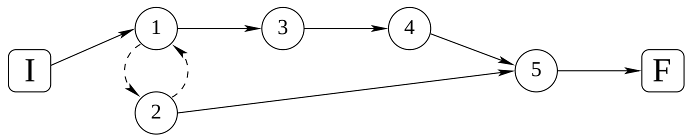

Figure 6: *This example of an outward deterministic graph contains a two-member polyarchy, a two-member hierarchy and a simple agent. In contrast to the stochastic graphs of section 4.1, this graph type cannot be reduced as the internal structure of the polyarchy may be important for the behavior of the graph. The numbers shown inside the nodes are merely labels used in the calculation of the graph acceptance probability. Once again the termination node, which is reachable directly from every internal node, is suppressed.*

```
(4)    I - 1 - 3 - 4 - T
(5)    I - 1 - 3 - 4 - 5 - T
(6)    I - 1 - 2 - T
(7)    I - 1 - 2 - 5 - T
```

and their probabilities are calculated from unity by multiplying by α for any accept and by $(1-\alpha)$ for any reject during the decision. Finally, the average cost can be computed from these decision probabilities and a simple count of the participating agents.

Often only the graph screening function is of interest (as is the case when simpler cost models are applied). Apart from the enumeration scheme used above (which can be done recursively, see section 3.5), a faster *backward method* (scaling according to the number of agents involved – squared in the worst case), can be used in simplified cases like this. Let $p[i]$ denote the probability of arriving successfully at the final portfolio if started at node i. Then these probabilities are related in the following way:

$$
\begin{aligned}
p[1] &\sim \alpha\, p[3] + (1-\alpha)\, p[2] \\
p[2] &\sim \alpha\, p[5] + (1-\alpha)\, p[1] \\
p[3] &\sim \alpha\, p[4] \\
p[4] &\sim \alpha\, p[5] \\
p[5] &\sim \alpha
\end{aligned}
$$

Here the tilde ($\sim$) means that any term proportional to the probability itself should be discarded because of the dynamics of a polyarchy (an agent cannot evaluate the same project twice). The graph screening function is equal to $p[1]$ and can be found by inserting the other probabilities in the proper way:

$$
\begin{aligned}
\mathcal{F}_{\text{example}}(\alpha) &= p[1] \;=\; \alpha\, p[3] + (1-\alpha)\, p[2] \\
&= \alpha\,(\alpha\, p[4]) \;+\; (1-\alpha)\,(\alpha\, p[5] + (1-\alpha)\, p[1]\cdot \mathbf{0}) \\
&= \alpha^2 \alpha\, p[5] \;+\; \alpha(1-\alpha)\alpha \\
&= \alpha^3\alpha \;+\; \alpha^2(1-\alpha) \;=\; \alpha^2\,(1-\alpha+\alpha^2) \qquad (15)
\end{aligned}
$$

Again, a natural extension of the model at hand is to allow rejection edges to go to nodes different from the termination node. To maintain the property of determinism, agents with such non-trivial rejection edges cannot have edges to the termination node as well, and the edges leading to termination should be included in the drawing of the graph to ease the interpretation. Cyclic structures may even be allowed as long as they are not dynamically cyclic. This could be implemented by allowing agents only to evaluate a project once; the second time, a termination is forced. In other words, sub-structures are not reset at re-entrance.

It may also seem natural to consider the class of architectures with restrictions on the dynamic in-degree as well. Supplying this extra requirement a simple model for inward deterministic graphs is obtained. The inward determinism is more a simplification than a complication as the graphs can always be placed in a 2D rectangular grid with polyarchies along one direction (apply cyclic boundary conditions to form ring-like structures) and the flow of control directed along the other. If there is no agent to catch the project in case of an accept it means that the project reached the final portfolio (see section 5 for examples).

4.3 A Note on Cyclic Graphs

Models allowing feedback support the most general graph structures. In order to ensure the finiteness of the evaluation process, the removal of the structural rule that disallow cycles must be accompanied by the introduction of a suitable dynamic rule. Such a rule could allow only some fixed number $k_0 > 0$ of evaluations (a project type like that of section 4.2 already keeps track of the evaluation count) before forcing some default behavior. Typically, sub-structures are reset upon reentry within these models. Using agents with internal dynamic variables, a rule that no single agent can evaluate the project more than k_0 times could be implemented.

The simpler backward method for calculating the reduced graph screening polynomial can generally not be applied to cyclic structures. Only if the cycles are cleanly nested, non-overlapping or some special symmetry is present this analytic approach can be applied with success. In these simple cases, substitution should proceed until the maximal evaluation count is reached, and the involved sums of multiple traversal of the same cycle can always be simplified using (arithmetic-)geometric series.

For more general graphs of several interconnected cycles, the truncation rule does, however, pose a problem. Fortunately, the finiteness of the evaluation process ensures a finite number of possible decisions. These can then be enumerated by the recursive method of section 3.5. Thus, in principle cyclic graphs are not more difficult to treat than most other finite graphs.

As a last digression, consider a cyclic graph in the absence of truncation rules. The graphs then contain infinite cycles. Although this is unrealistic with respect to real world problems, they are still interesting either from a mathematical viewpoint or as the limit of long evaluation processes ($k_0 \to \infty$). It turns out, that due to the allowed infinite number of decisions, most results can now be expressed as infinite sums yielding very simple results. Even though the previously mentioned methods cannot be applied successfully, the reduced graph screening (no longer a polynomial) can always be solved explicitly from a system of equations constructed by taking a round-trip through the various loops.

5 Analytic Results for Two Simple n-Member Models

The purpose of this section is to illustrate how predictions about alternative configurations of organizational architecture can be derived from performance comparisons.

To approach the study of real-world phenomena, it is useful to distinguish between project-evaluation models in terms of the correlation between choice and problem. If the correlation is perfect, the screening is perfect, corresponding to an enlightened agent receiving projects with a perfect description, and the analysis is trivial. If the correlation is imperfect, the screening function is variable and the agent receives projects with imperfect description. Finally, if the correlation is absent, the agent suffers pervasive uncertainty regarding project descriptions, and the screening function is trivial. We begin by considering the apparently most interesting case, projects that come with imperfect descriptions.

5.1 Projects with Imperfect Descriptions

Consider the family of finite, non-cyclic models satisfying the 7 usual invariants defined in section 3.2.4. For now simplify by taking the common limit of zero cost. The simple scalar project, x, representing the income can be used ($\mathcal{P}(x) = x$) and projects are drawn from a given distribution G. Assume that agents have a monotonous agent screening function f, and that the edges support no dynamic state variables. There is no transformation of the projects ($\mathcal{T} = I$). Still, as described above, there is room for various types of edges, evaluation modes, causality modes and graph geometries/topologies. Non-trivial costs will be discussed below.

A single agent has a monotonous, reduced screening function of $\alpha \equiv f(x)$, whereas the graph as an entity has a screening function of $\mathcal{F}(\alpha) = q(f(x))$, where $\mathcal{F}$ is the polynomial dictated by the geometry and topology of the graph. The highest order is equal to the number of agents on the longest path to overall accept. The lowest order (possible) is equal to the minimal number of accepts by agents required to get an overall accept. Since the invariants require the graph to have an agent and the final portfolio to have only in-bound accept edges, the polynomial will always have degree of 1 or more, and similarly the lowest, non-trivial order cannot be zero.

Which is best, agent or organization? To answer that question (using expected profit as a measure of performance), the exact agent screening, f, and the exact income distribution, G, must be known as the difference in performance. Invoking equations (12)-(13)

$$E[\Delta\text{profit}](f, G) = \int_{-\infty}^{\infty} x\, G(x)\ (\mathcal{F}(f(x)) - f(x))\ dx \tag{16}$$

$$= \int_0^1 f^{-1}(\alpha)\, \frac{G(f^{-1}(\alpha))}{f'(f^{-1}(\alpha))}\ (\mathcal{F}(\alpha) - \alpha)\ d\alpha \tag{17}$$

may have any sign depending on these two functions.

An n-member hierarchy has a graph screening of $\alpha^n < \alpha$ and an n-member polyarchy has a graph screening of $1 - (1 - \alpha)^n > \alpha$, assuming non-trivial structures ($n > 1$). Thus if the projects take only positive values, then any polyarchy performs better than the single agent, which again performs better than any hierarchy[12]. If, on the other hand, the projects take only negative values, then order is reversed as any hierarchy will outperform the single agent, which again will outperform any polyarchy. By continuity, examples with distributions taking both positive and negative project values will exist, where the ordering after performance is not perturbed.

Nevertheless, as illustrated in figure 7, it is possible that the graph-screening may outperform the agent screening for both positive and negative values of the project distribution.

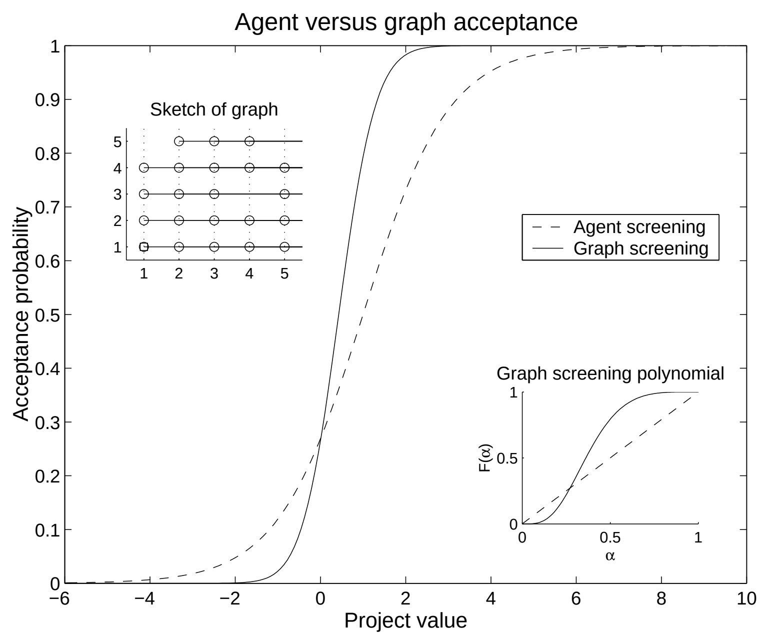

Figure 7: *An example of an inward deterministic graph. Projects always enter at node* $(1,1)$*. Applying cyclic boundary conditions, rejection moves a project one step further within the same hierarchical layer (possibly ending in final rejection). Acceptance moves a project one step further across hierarchical layers (possibly ending in final acceptance). Both the agent and the graph screening is plotted for positive as well as negative project values, clearly illustrating that there exist forms of economic organization that definitely help agents reduce judgment error. Agent screening* $f(x) = (\tanh((x-1)/2)+1)/2$ *was used in the plot.*

For more general organizations, "deep" graphs (dominated by hierarchical structures) will tend to have only a few very high order terms in the polynomial $\mathcal{F}$, and "wide" graphs (dominated by polyarchical structures) will tend to have terms of all orders and possibly with high coefficients.

From the assumption of monotonous agent screening follows that f' is positive, hence the sign of the performance difference depends on the sign of the project value x and the difference in reduced screening in equation (17), i.e. the first and last factor of the integrand. This leads to the following weak conclusion

[12]Note that Sah & Stiglitz omit comparisons with the single agent.

Proposition 1 *For models including agents with monotonous agent screening function f, and no transformation of the projects ($\mathcal{T} = I$), graphs dominated by polyarchies perform better than the single agent when projects have positive value, and those dominated by hierarchies perform even worse than the single agent. When projects have negative value, the order is reversed.*

In the case of projects taking both positive and negative values, the picture is more blurred. Nevertheless, again invoking equation (17), a strong conclusion can be drawn

Proposition 2 *Considering the family of legal models including agents with monotonous agent screening function f, and no transformation of the projects ($\mathcal{T} = I$), for any given distribution, G, taking both positive and negative values, agent screenings, f, exist so that the single agent performs marginally better than any graph.*

As illustrated in figure 8 below, the challenge of deriving such results is increased, however, by the possibility that even if the agent screening function $\alpha \equiv f(x)$ is monotonous it does not follow that the screening function of the graph $\mathcal{F}(\alpha)$ is monotonous.

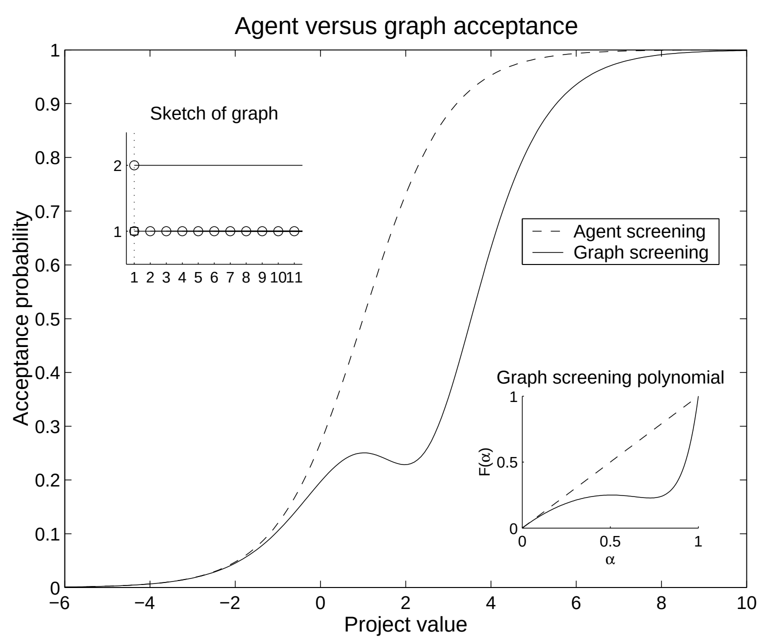

Figure 8: *Another example of an inward deterministic graph (for details, see the above Figure). Even though the agent-screening, $f(x) = (\tanh((x-1)/2) + 1)/2$, is monotonous it does not follow that the graph screening is so too. Note also that for this instance, the agent is better off than the organization.*

The introduction of any realistic cost model (as the three defined in section 3.2.1) will further favor agents in cases where they already perform the best, whereas it will lead to inconclusive pictures in cases favoring organizations (the optimum might now, quite naturally, be an act of balance between income and cost).

5.2 Projects Lacking any Description

Again, consider the family of legal models satisfying the usual list of 7 invariants. Assume a scalar project, x, constant agent screening $f(x) = f$ implying constant graph screening function as well $q(x) = q$, and no transformation of the projects ($\mathcal{T} = I$) drawn from a project distribution G. This is the situation where the uncertainty is so high that the decision looses its correlation to the project.

Invoking equations (12) and (13), the expected profit

$$E[\text{profit}] \;=\; q\mu - C \tag{18}$$

(where $\mu = E[x]$ and $C = E[\text{cost}]$) is the best measure of the performance of the architecture and the conclusion is simple. Given some cost model, choose the architecture with the lowest average cost. This can be done independent of the project distribution G because of the constant value of the screening function. With this done, the optimal graph screening is $q = 1$ in good times ($\mu > 0$) and $q = 0$ in bad times ($\mu < 0$). These graph screenings can only be achieved (for finite graphs) by similar screening functions, $f = 0$ or $f = 1$ relatively, for the agents. Note, that in the special case, $\mu = 0$, the value of q is indeterminate, but still the architectures of lowest average cost will outperform the rest.

For agent-screening $0 < f < 1$, the best graph screening that can be obtained is the one closest to the limit of $q \to 1$ or $q \to 0$ for good or bad times, respectively. When the benefit of incrementally increasing the size of the graph to approach the desired limit is weighed against the costs of doing so, finding the best configuration requires further specification of the cost model and prospect distribution.

To sum up, the propositions and considerations offered here illustrate how predictions about alternative configurations of organizational architecture can be derived from performance comparisons. The results presented here are preliminary, but also serve to focus the work ahead. In the case of models including agents with non-trivial screening functions (i.e., the projects with imperfect descriptions), the next step is to further extract information about graph-screening and performance. This problem is considerably complicated since a monotonous agent-screening does not necessarily imply monotonous graph-screening (as shown in the above example in figure 8). A further step ahead lies in the study of models including non-trivial project transformation (affecting the acceptance probability or even the project quality).

6 Conclusion

In order to extend previous project selection models, we have developed a general framework that allows modelling of any finite structure. The previous models were restricted to modelling explicit structures of two members. That is, although these models in principle considered committees including any finite number of agents, the internal structure of these committees was left unspecified. A number of ambiguities in the previous models further prevented any straightforward generalization to decision-making structures including three members or more.

In order to clarify these ambiguities, we have proposed a general framework to model n-member decision-making structures. This framework presents the organization of connected agents as a graph, and outlines a clear distinction between the decision-making structures, the dynamic rules imposed on it, the properties of the agents, and the properties of the projects that are screened by the agents.

In order to promote the study of realistic social organizations (including firms, auditing teams, and marketing teams), we identified seven invariants that must be imposed on the graph in order to exclude empirically infeasible structures. In addition, we proposed that the dynamic rules used to model realistic structures should restrict the processing of projects to finite time. A further fundamental issue was the definition of the causality of the graph being either stochastic or deterministic. As we have argued, deterministic graphs represent a high degree of internal structure and promote the modelling of incremental project evaluation, whereas stochastic graphs may lack any structure and promote simultaneous project evaluation.

With these building blocks in place, it is now possible to generalize the project selection model of two members to any finite structure. We have further provided the necessary tools that allow the evaluation of the relative performance of alternative structures in terms of their expected profits. Typically, the performance evaluation of alternative social organizations assumes a known distribution of projects along with a definition of the graph, the dynamic rules imposed on it, and the properties of the agents in terms of their screening function and ability to transform projects as these traverse the structure. In order to proceed with the performance evaluation, it is therefore typically necessary to extract information about the graph screening function. As we have illustrated, this is in principle possible for all types of decision-making structures. The focus in this paper has been to clarify the ambiguities in the previous project selection models that prevented generalization to larger structures. We also presented a set of preliminary results to illustrate how predictions about alternative structures can be derived from performance comparisons, and to point out the work ahead.

Finally, it is worth considering the implications of our proposed framework for the study of real-world phenomena. It was suggested that the study of larger project selection models may lead to consideration of ignored problems related to the design of marketing and audit teams, vertical integration of business firms, and the possible emergence of new organizational forms in response to electronic communication networks. From the above exposition, it seems that our proposed framework is rich enough to capture such a variety of problems. We also believe that the proposed general framework is worth considering as a basis for design-issues (e.g. of marketing and audit teams), for the analysis of emergent empirical phenomena (e.g. the so-called new organizational forms), and for the possible correction of the predictions offered by Transaction Cost Economics and other theories of economic organization.

References

[1] R.K. Sah and J.E. Stiglitz. The architecture of economic systems: Hierarchies and polyarchies. *Am. Econ. Rev.*, 76:716, 1986.

[2] R.K. Sah and J.E. Stiglitz. Committees, hierarchies and polyarchies. *Econ. Jour.*, 98:451, 1988.

[3] J. Marschak and R. Radner. *Economic Theory of Teams*. Yale University Press, 2nd edition, 1976.

[4] H.A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69:99, 1955.

[5] H.A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63:129, 1956.

[6] H.A. Simon. Rational decision making in business organizations. *Am. Econ. Rev.*, 69(4):493, 1979.

[7] J.G. March and H.A. Simon. *Organizations*. New York: Wiley, 1st edition, 1958.

[8] W.T.H. Koh. Variable evaluation costs and the design of fallible hierarchies and polyarchies. *Economics Letters*, 38:313, 1992.

[9] N. Christofides. *Graph Theory - An Algorithmic Approach*. Academic Press, 1975.