Model description

Stephan Leitner
University of Klagenfurt, Austria
stephan.leitner@aau.at

Version as of May 3, 2024

This model description follows the ODD (Overview, Design concept, Details) protocol for describing individual- and agent-based models (Grimm et al. 2006), as updated by Grimm et al. (2020).

1 Purpose and patterns

This model aims to examine how task complexity, decision-making styles, and incentive strategies impact an organization's ability to withstand environmental disturbances. A model of a stylized organization is employed to facilitate this analysis. Depending on the prevailing decision-making approach within the organization, this model may include either several decision-making individuals or a combination of decision-making individuals and an additional coordinating figure.[1]

Patterns It is apparent that some organizations handle such disruptions more effectively than others, and similarly, some are better at recovering from shocks. There is no consensus on the factors driving these patterns. This model seeks to replicate the patterns observed in practice, and it aims to explore whether organizational characteristics (such as decision-making methods, incentives, and task complexity) are a driving force of these patterns.

2 Entities, State variables, and scales

The following three paragraphs outline entities of the model along with the relevant state variables. For a consolidated view of entities and state variables, refer to Tab. 1. The last paragraph in this section discusses the relevant scales.

Decision-making agents The model describes a stylized organization dealing with a problem that follows the logic of the NK framework. This means that the members of the organization are required to N binary decisions, and there are K interdependencies between these decisions,

[1]Please be aware that the model illustrates a scenario in which a stylized organization faces a complex decision-making task. This complex task is segmented into smaller pieces and assigned to different decision-making agents. The agents are tasked with deciding on the actions necessary to address their particular sub-tasks. Consequently, the terms "decision" and "action" are used interchangeably throughout the model description.

Table 1: State variables

Entity / variable name	Variable type	Meaning
Decision-making agents:		
assigned-tasks	Set of tasks, static	Part of the decision problem the entire organization faces that is assigned to a specific agent.
decision-mode	Categorical, static	mode according to which an agent makes decisions about actions to tackle the decision-making tasks assigned to them.
own-actions	Bitstring, dynamic	The actions taken by an agent in a specific period to tackle *assigned-tasks*.
residual-actions	Bitstring, dynamic	An agent's information about the actions that the other agents have taken or will take to tackle their *assigned-tasks*.
own-proposals	Set of bitstrings, dynamic	If the *decision-mode* is set to "Hierarchical: Proposal-mode," agents evaluate and prioritize two viable actions to address their *assigned-tasks* and submit these suggestions to the coordinating agent.
Environment:		
performance-contributions	$\sim U(0, 1)$, dynamic	Performance contributions of individual decisions to overall performance. Should the organization experience a shock, the state variable *performance-contributions* is subject to change.

shaping the complexity of the decision problem. This organization consists of several decision-making agents, each symbolizing a different department. The N tasks requiring decisions are distributed among these agents, with each agent being assigned a fixed set of tasks, denoted by the state variable *assigned-tasks*. As time progresses, these agents are responsible for addressing their respective tasks. The methodology an agent uses to choose their actions to do so is defined by the state variable *decision-mode*, while the specific actions undertaken by an agent are recorded in the state variable *own-actions*. If the decision-making mode "Hierarchical: Proposal-mode" is active during a simulation run, decision-making agents make proposals for actions to the coordinating agent; these proposals are recorded in the state variable *own-proposals*. Furthermore, when evaluating possible actions they can take, agents consider what they know or anticipate about the actions the other decision-making agents will take; this information is recorded in the state variable *residual-actions*.

Coordinating agent In certain cases, a coordinating agent is present to support the decision-making agents. This coordinating agent might, for example, symbolize the organization's headquarters. Only when the organization operates under the "Hierarchical: Proposal-mode" decision-making approach, the decision-making agents submit action proposals to the coordinating agent, who then decides on the final course of action. The coordinating agent's role is solely to facilitate decision-making, which is why this agent is not characterized by a specific state variable.

Environment The task environment in which the decision-making and coordinating agents operate is represented as a performance landscape. This landscape establishes a mapping between all the actions that decision-making agents can take and their contributions to the performance of the organization, with these contributions detailed in the state variable *performance-contributions*. These performance contributions undergo changes when the organization is impacted by an environmental shock.

Scales The model is an abstract model designed for a theoretical exercise to understand the relation between coordination modes, task complexity, and organizational resilience. Timesteps within the model represent decision-making cycles within an organization, but there is no absolute concept of temporal scale.

The binary decision decision problem is of size N resulting in 2^N possible solutions for this decision problem, whereby all these solutions and their corresponding performances are captured by the performance landscape. Agents navigate the landscape by changing their state variable *own-actions*. However, the model does not include a concept of a spatial scale, meaning that agents do physically move in the landscape but rather their "position" in the landscape gives information on the performance associated with their actions.

3 Process overview and scheduling

Algorithm 1 Simulation Algorithm

Initialization:
Set *performance-contributions* based on submodel A.
Assign *assigned-tasks* using submodel B.
Fix starting position and update *own-actions* and *residual-actions* for all decision-making agents using submodel B.
Set *decision-mode* for agents to options submodels C1 to C4.
Initialize simulation time to 1.

Simulation Loop:
while further timesteps required **do**
  if Decentralized decision-mode (Silo-based, Collaborative) **then**
    for each decision-making agent **do** simultaneously
      Update *own-actions* per submodels C1 or C2.
    end for
    Aggregate *own-actions* and update *residual-actions* for all decision-making agents.
  else if Decentralized decision mode (Sequential) **then**
    for each decision-making agent **do** in sequential order per agent index
      Update *own-actions* per submodel C3.
      for each decision-making agent waiting to make decision **do**
        Update *residual-actions* per submodel C3.
      end for
    end for
    Aggregate *own-actions* and update *residual-actions* for all decision-making agents.
  else if Hierarchical decision mode (Proposal-based) **then**
    for each decision-making agent **do** simultaneously
      Generate *own-proposals* per submodel C4 and submit to coordinating agent.
    end for
    Coordinating agent evaluates proposals and makes decision.
    Decision-making agents observe decision and update *own-actions* and *residual-actions*.
  end if
  Track performance of collective actions.
  Check for environmental shocks and adjust *performance-contributions* according to submodel D if needed.
  Increase simulation time by 1 and check continuation condition.
end while

End of algorithm:
Finalize simulation; compile and return results.

Process overview The model encompasses a duration of T periods, representing the lifetime or decision-making cycles of an organization. To effectively simulate this timeframe, several processes are necessary.

The **initialization** of the model involves three key initial steps: (i) determining the performance landscape, (ii) assigning specific tasks to the decision-making agents and the initial position within the performance landscape, and (iii) initialize further state variables. In the first step (i), performance landscapes following the NK model are constructed. This involves setting up an N-dimensional decision problem with specific performance contributions captured in the state variable *performance-contributions*. In the second step (ii), this N-dimensional problem is broken down into smaller, more manageable decision-making tasks. These tasks are then distributed to the respective decision-making agents, which is stored in the state variable *assigned-tasks*. Part of this step also includes establishing each agent's starting point within the performance landscape. This is done by randomly determining a starting position in the landscape (i.e., a random N-dimensional bitstring) and updating *own-actions* and *residual-actions* for each decision-making agent correspondingly when initializing the model. Finally, step (iii) involves the initialization of the state variables. The steps (i) and (ii) are detailed further in submodels A and B, respectively, in Secs. 7.1 and 7.2. Fruther detailed information on all three steps of the initialization phase is provided in Sec. 5.

The **main simulation loop** of the model focuses on the decision-making behavior of agents, paying particular attention to updates to *own-actions* and *residual-actions* during simulations. Every timestep in the model represents one decision-making cycle, and therefore, decision-making agents (and, if necessary, the coordinating agent) make decisions in every timestep throughout the simulation. There are four main decision-making approaches:

1. **Decentral: Silo-Based**. Detailed in submodel C1 in Sec. 7.3.1, this approach has agents making decisions by themselves without exchanging information with others, updating their *own-actions* all at once. After all agents have made their decisions, the behavior of the organization is the aggregation of all decision-making agents' *own-actions* (referred to as overall solution or collective actions). All decision-making agents observe the overall solution and update *residual-actions*.

2. **Decentral: Collaborative**. Here, as detailed in submodel C2 in Sec. 7.3.2, agents have a chance to collaborate with peers for decisions (with a fixed probability). If decision-making agents do not collaborate, they make decisions according to the silo-based approach detailed above. Updates to *own-actions* are made simultaneously, combining both collaborative and individual decisions. The variables *residual-actions* are updated after all decisions have been made and all decision-making agents have observed the overall solution.

3. **Decentral: Sequential**. Explained in submodel C3 in Sec. 7.3.3, in this setup, agents update their *own-actions* one by one (sequentially by agent index). Agents share their decisions with others waiting their turn. The variables *residual-actions* get updated as each agent makes a decision and information flows to the other agents waiting to make their decision. Once all decision-making agents have made their decisions, the overall solution is computed and observed by all agents; then, all decision-making agents update *residual-actions*.

4. **Hierarchical: Proposal-based**. In this method, detailed in submodel C4 in Sec. 7.3.4, agents create action proposals and send them to a coordinating agent who makes the final

decisions. Afterwards, all agents update their *own-actions* and *residual-actions* based on this centrally made decision.

On overview of the four decision-making modes and the corresponding state variable updates, in particular to illustrate the dynamics between decision-making cycles, is provided in Fig. 1.

The last relevant process in the main simulation loop deals with environmental shocks impacting the organization. This is addressed in submodel D, outlined in Sec. 7.4, where the state variable *performance-contributions* gets updated. Throughout the simulation, the organizational performance determined by the overall solution is kept track of. At the end of a simulation run, the results are compiled (as a time series of performances) and returned.

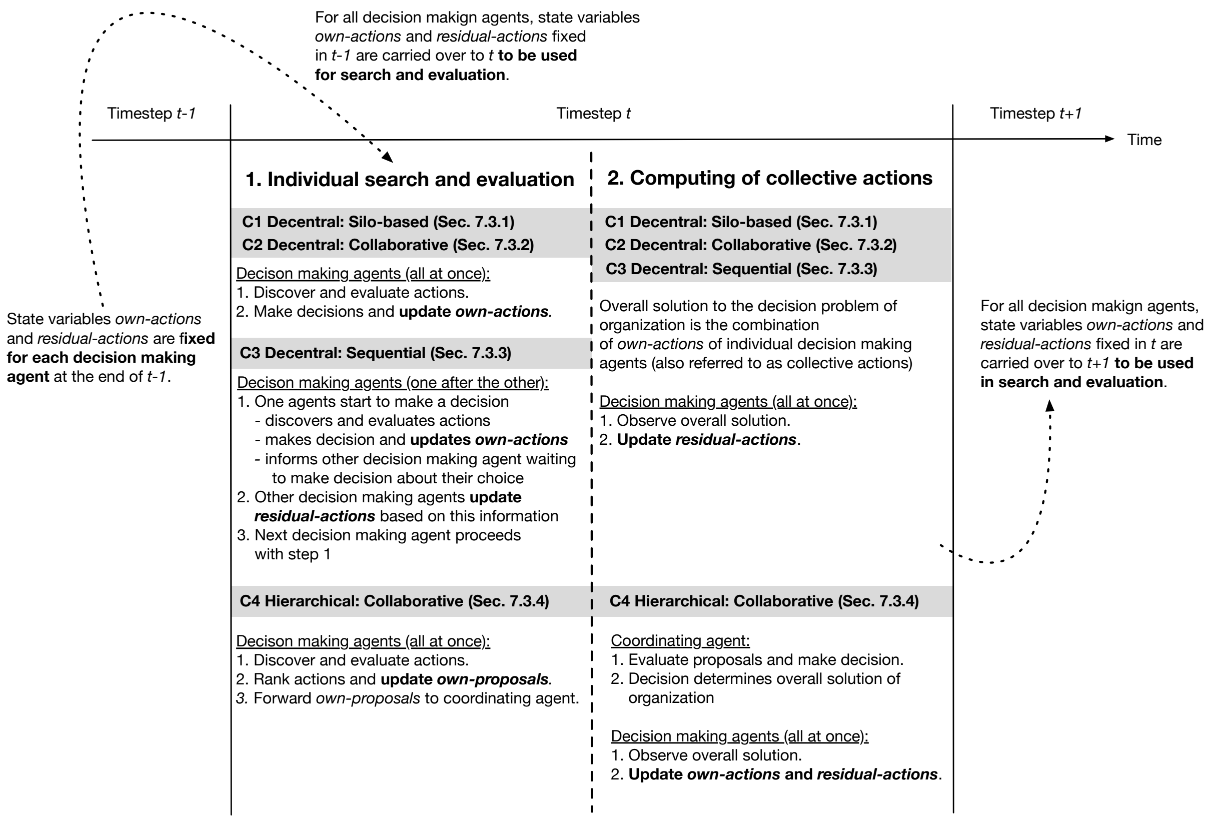

Figure 1: State variable updates of decision-making agents and coordinating agent

Scheduling An overview of the scheduling of the processes described above is provided in Algorithm 1. Each simulation run begins with an initialization phase, during which submodels A and B are run, see Secs. 7.1 and 7.2. This entails generating the performance landscapes (in submodel A, setting initial values of *performance-contribution*), assigning tasks, and establishing the initial placements of agents within the landscape (the initial configuration of decisions) (in submodel B, setting the initial values of *assigned-tasks*, *own-actions*, and *residual-actions*). Additionally, the decision-making mode for this simulation round is determined by the modeler (setting *decision-mode*), and the simulation time is set to a starting value of 1.

The main simulation loop lasts for T time periods. In this phase, decision-making agents (and a coordinating agent if needed) follow the procedures set out in submodels C1 to C4 in Sec. 7.3 to make their choices, updating their *own-actions* and *residual-actions* (and, if required, *own-proposals*) over time as described in these submodels and illustrated in Fig. 1. Through their decisions, the agents shape the stylized organization's behavior via the collective actions, influencing its overall performance. The performance outcomes are recorded at each time step. If the organization experiences an environmental shock, submodel D is activated, leading to a recalculation of the performance landscape and an update of the state variable *performance-contributions* with new values, as described in Sec. 7.4.

Upon concluding the simulation, the recorded performances throughout the main simulation loop are compiled and returned as a time series result.

4 Design concepts

4.1 Basic principles

The model represents a stylized organization facing complex decision-making problems. This organization consists of various decision-making agents (symbolizing different departments) and a central coordinating agent (representing the central office). Below, the key design principles of the model are outlined.

Task environment The model employs NK performance landscapes to depict the task environment of the stylized organization. Originally introduced in the field of evolutionary biology, the NK model has been used to examine the adaptive capabilities of biological systems, such as entire genomes, as described by Kauffman and Levin (1987). This framework was later adapted to management science by Levinthal (1997), where it has been extensively applied across various contexts (Wall 2016; Wu, Ohya, and Sekiguchi 2023).

These landscapes facilitate the representation of an N-dimensional binary decision problem with K interdependencies among decisions. The interdependencies serve to effectively represent the complexity of the decision problem through the ruggedness of the resulting performance landscape, with more rugged landscapes indicating scenarios where achieving the global optimum is more challenging. For implementation details, refer to submodel A in Sec. 7.1.

Task decomposition As time progresses, the agents within the model explore the landscape to search for solutions to the decision problem which yield a higher performance, which mirrors the evolutionary theories on the dynamics of organizations (Dosi and Marengo 2007). The agents are, however, characterized by bounded rationality, meaning they face limitations in time and cognitive abilities (Simon 1956, 1967), which is further elaborated in the section on search behavior. As a result, in the model, no single agent is capable of solving the entire N-dimensional decision-making problem on their own; instead, this complex problem is segmented, and individual sub-tasks are assigned to various decision-making agents. This modelling choice is supported by the information processing viewpoint on organizational design, which posits that decision-making is a process rich in information processing and that, consequently, delegating decision-making powers to managers, regardless of their access to private information, serves as a strategy to meet the organizational demands for processing information (Galbraith 1974; Garicano, Van Zandt, et al. 2013). Additionally, this strategy of modularization is a fundamental concept in managing complex systems like organizations since it promotes specialization and enhances adaptability to different environmental tasks (Lawrence and Lorsch 1967; Zhou 2013).

Search behavior All agents, including both the decision-making agents and the coordinating agent, aim to maximize their utility but are subject to bounded rationality. This means they face limitations in cognitive capabilities, information, and time, among other factors (Simon 1956, 1967). Additionally, they cannot identify the optimal point at which to cease gathering information (Vriend 1996). Therefore, their behavior corresponds with the concept of bounded rationality outlined by Gigerenzer and Selten (2002). As a result, these agents do not maintain a record of their past actions; instead, they possess only a short-term memory that covers the most recent decision they have made (Frey 2005). Furthermore, their foresight is limited; they only consider the immediate consequences of their decisions without recognition of long-term impacts. This means, according to the perspective of Epstein, Marinacci, and Seo (2007), they are not "sophisticated" agents: they do not recognize their own limitations and therefore do not factor these limitations into their decision-making processes.

The primary role of the agents in the model is to identify and implement improved actions to address their specific tasks, given the constraints mentioned in the previous paragraph. They engage in a sequential search process, aiming to incrementally enhance their performance by moving towards superior positions in the landscape (Simon 1955). This method of exploration mirrors common managerial challenges (Baumann, Schmidt, and Stieglitz 2019). Conceptually, their search method of search aligns with the established hill-climbing algorithm, a step-by-step strategy aiming for local maxima in the hope of achieving the best possible outcomes in the landscape (Cormen et al. 2022). Consequently, decision-makers only transition within the landscape if the new options indicate a performance improvement, avoiding temporary performance setbacks in anticipation of future benefits (Tracy et al. 2017; Wall and Leitner 2021). This search behavior is a consequence of their bounded rationality (particularly of limited foresight).

Coordination mechanisms While breaking down tasks and allowing for specialized searches aids in specialization and adaptation, it also necessitates a form of coordination among decision makers, especially for complex tasks (Simon 1962; Zhou 2013). Following the distinction by Nadler, Tushman, and Nadler (1997), the model incorporates two types of coordination. First, there is an indirect coordination through incentives (Kretschmer and Puranam 2008), where the incentive structure influences the agents' objectives and, consequently, their decision-making behavior. For more information, see Sec. 4.4. Second, the model includes direct coordination mechanisms that affect the way agents communicate and share information during their searches. It details four specific coordination approaches: at one end, there is decentralized (silo-based) decision-making, where agents operate independently without inter-agent information flow, and at the other, hierarchical (proposal-mode) decision-making, in which agents submit proposals to a central authority that makes the final decisions. These two modes align with the archetypes proposed by Siggelkow and Rivkin (2005).[2] The model also includes two intermediate coordination types: decentralized decision-making with sequential communication (Nadler, Tushman, and Nadler 1997; Wall 2018; Blanco-Fernández, Leitner, and Rausch 2023), and a collaborative decision-making mode where agents determine the instances for collaboration based on certain probabilities (Yuan and McKelvey 2004; Leitner 2023). The specifics of these coordination modes are detailed in the submodels C1 to C4, found in Secs. 7.3.1 to 7.3.4.

4.2 Emergence

The key result of the model is the performance of the organization, which stems from the collective actions of the agents within the stylized organization. These actions arise from the decisions of the individual agents within the model. For models employing decentralized decision-making

[2] Note that centralized decision-making, where agents or departments have no autonomy at all (Siggelkow and Rivkin 2005), is intentionally omitted to focus on organizations with some level of decentralized decision-making.

(Sec. 7.3.1 to 7.3.3), the organization's actions are the cumulative result of the actions taken by individual decision-making agents. Conversely, in models using hierarchical decision-making (Sec. 7.3.4), a central coordinating agent is responsible for determining the organization's overall actions, although this agent's decisions are influenced by the initial choices made by the individual decision-making agents.

4.3 Adaptation

As further detailed in the submodels capturing the decision-making modes in Sec. 7.3, decision-making agents encounter various alternative actions to address their assigned tasks through a random process; they randomly discover alternative actions to tackle their assigned tasks. Recall that decisions are encoded as bitstrings, with each agent's most recent decision recorded in the state variable *own-actions*. Agents then randomly identify an alternate action within a Hamming distance of one from their *own-actions*, and then they evaluate these actions as follows: They compare the current action (*own-action*) against this new alternative using a direct objective-seeking approach, assessing which better fulfills their objectives, which involves predicting the performances associated with the two actions. Then, they adapt their behavior, i.e., they switch to the action that promises the higher fulfilment of their objectives. Details on the agents' objectives are provided in Sec. 4.4, and predictions to evaluate actions are detailed in Sec. 4.6.

4.4 Objectives

The decision-making agents' objective functions Recall that decision-making agents are boundedly rational utility maximizers, and that they follow an objective-seeking strategy. Their utility is influenced by the incentive mechanism put in place by the organization, with the model incorporating straightforward, linear incentive approaches. The utility of an agent is determined by the performance contributions coming from their own decisions (*own-actions*) and the performance contributions of the decisions of other decision-making agents (stored in *residual-actions*). Let $c(\cdot)$ be the function that returns the performance of a set of actions, then $c(\textit{own-actions})$ and $c(\textit{residual-actions})$ are the two main components of a decision-making agent's utility function. These two components are weighted using a parameter λ, with is referred to as the incentive parameter. Then, the objective of a decision-making agent is

$$U(\textit{own-actions}, \textit{residual-actions}) = \lambda \cdot c(\textit{own-actions}) + (1-\lambda) \cdot c(\textit{residual-actions}) \ \rightarrow \max! \tag{1}$$

Adjusting the incentive parameter λ can influence agent behavior significantly. In detail, when λ is set at higher values, agents tend to focus more on optimizing the performance within their specific domain rather than the organization's collective performance. Conversely, lower values of λ encourage agents to also consider the performance outcomes of other agents within the stylized organization. The scenario with high λ values can be described as adopting **individualistic incentive mechanisms**, whereas the scenario with low λ values introduces **group-based incentive mechanisms**.

The coordinating agent's objective function The coordinating agent only contributes to decision-making when the hierarchical decision-making mode is active in the organization. This agent is focused on optimizing the performance of the solution for the entire decision problem. Thus, the coordinating agent is interested in the aggregate of the individual proposals made by the decision-making agents in the hierarchical decision-making mode (see Sec.7.3.4

for details). Every decision-making agent stores two proposals in *own-proposals*, and the coordinating agent computes two concatenations of the proposals. Let us denote this aggregation by $\mathbf{P} = \{\cup \textit{own-proposals}^{(1)}, \cup \textit{own-proposals}^{(2)}\}$. Then, the coordinating agent's objective function can be formalized by

$$U(\mathbf{P}) = \max_{p' \in \mathbf{P}} c(p') \ . \tag{2}$$

Please note that there might be a conflict of interest between the coordinating agent and the decision-making agents, as their objective functions differ. This means that what might be a superior result for one type of agent, might be the inferior one for the other type of agent.

4.5 Learning

The model includes no learning.

4.6 Prediction

In the decision-making process, agents evaluate two options based on how well they meet their goals. In making this evaluation, agents also take into account what they know about the actions of other agents, which is captured in *residual-actions*. The specifics of what information decision-making agents have at different times are detailed in Sec. 4.7, while the evaluation methods for the various decision-making approaches are outlined in submodels C1 to C4, covered in Secs. 7.3.1 to 7.3.4.

Both decision-making agents and the coordinating agent are presumed to be able to estimate the performance outcomes for the upcoming period, but they lack the ability to make predictions for periods beyond the immediate next one (they are myopic). It is important to note that these predictions are based on information about *residual-actions* that is probably outdated (as detailed in Sec. 4.7), leading to less accurate forecasts. Beyond this source of error in predictions, the model does not incorporate additional forecasting errors since doing so would only further skew these predictions with minimal impact on the outcomes.

4.7 Sensing

Sensing in individual search and assessment The information that decision-making agents use in individual search and evaluation of actions, slightly vary between the decision-making models. Specifically, the state variable *residual-actions* is relevant in this context. Please also see Fig. 1 for a graphical representation of when relevant state variables are updated.

Initially, in the new decision-making cycle, agents explore potential actions within their designated responsibilities (*assigned-tasks*) aiming to better meet their objectives. In their independent search and assessment, agents rely on two kinds of information:

- Firstly, they randomly discover an alternative action close to their most recent action (within a Hamming distance of one from *own-actions*) to address their assigned tasks.

- Secondly, they apply what they know from *residual-actions* during the evaluation, which is necessary because both the outcomes from *own-actions* and *residual-actions* are part of their objective functions (see Sec. 4.4) and they might also influence each other due to interdependencies (see Sec. 4.8).

The state variable *residual-actions* is treated differently across decision-making scenarios: In submodel C3 Decentral: Sequential (Sec. 7.3.3), the state variable *residual-actions* is progressively updated for agents who decide later in the sequence. In contrast, in all other scenarios, agents rely solely on the information from the previous period, which may no longer be current.

Sensing at the end of a decision cycle After each decision-making cycle concludes, agents evaluate the collective organizational actions (i.e., the complete solution addressing every facet of the organization's decision problem) and update their state variables accordingly. As a result, by the cycle's close, every decision-making agent has precise and current information. Consequently, at the cycle's end, all agents hold updated data within their state variables *own-actions* and *residual-actions*.

4.8 Interaction

In the proposed model, interactions are a critical element, with both direct and indirect interactions playing significant roles as outlined below:

- **Direct interactions** are specified within the decision-making frameworks included in the model. In the collaborative and sequential decision-making settings, as detailed in submodels C2 and C3 in Secs. 7.3.2 and 7.3.3, decision-making agents exchange information about the actions under consideration. Further, interactions between decision-making agents and the coordinating agent are essential in the hierarchical decision-making framework (detailed in submodel C4 in Sec. 7.3.4), where the focus is on exchanging information about ranked proposals. In contrast, the silo-based decision-making approach, introduced in submodel C1 in Sec. 7.3.1, involves no direct interactions among agents.

- **Indirect interactions** stem from the structure of the performance landscape itself. Given that the organizational decision problem involves N tasks and K interdependencies between these tasks (see submodel A in Sec. 7.1 for details), the effectiveness of a particular action in addressing a decision may be influenced by the actions selected for other decisions. Thus, when an agent decides, it can inadvertently impact other agents via the interdepencies encoded in the landscape.

4.9 Stochasticity

The following processes within the model are driven by stochasticity:

- **Landscape creation**: The initialization of performance landscapes includes setting up the individual contributions that each decision has towards the overall performance. These contributions are assigned random values from a uniform distribution, $U(0, 1)$. Additional information is available in Sec. 7.1.

- **Starting position initialization**: The model's initialization includes generating a starting point within the landscape, marking the start of the agents' search. This initial position is determined by generating a random bitstring of length N, which serves as the initial location. This random bitstring is then applied to set up the state variables *own-actions* and *residual-actions* for all decision-making agents.

- **Environmental shocks**: In scenarios where the simulated organization encounters an environmental shock, there is a recalculation of the performance landscape. Specifically, the model addresses correlated shocks meaning that the existing performance contributions within the landscape are substituted with correlated counterparts. This adjustment is carried out according to a method by Demirtas (2014), which includes drawing of two distinct values from a uniform distribution and a Beta distribution, respectively. More information on this process can be found in submodel D, detailed in Sec. 7.4.
- **Search Procedure**: Within each decision-making cycle, agents explore alternative strategies to address their assigned tasks. Their exploration is focused around the neighbourhood of the action currently in place, as recorded in *own-actions*, with this neighbourhood specified by a Hamming distance of one. The exploration mechanism involves randomly flipping one bit of the bitstring in *own-actions*, ensuring each bit has an equal chance of being altered.
- **Collaborative search**: In the decision-making model C2 "Decentral: Collaborative", decision-making agents are interconnected in a ring-like network, and they may engage in collaboration with a neighboring agent based on a specific probability. This approach incorporates several stochastic elements. Initially, a number is randomly selected from a uniform distribution to assess whether a decision-making agent opts to collaborate, comparing it against a predetermined probability. Subsequently, if agents decide to collaborate, the neighbor with whom they will work is chosen at random from their network, ensuring each neighbor has an equal chance of selection. Details are provided in submodel C2 in Sec. 7.3.2.

4.10 Collectives

In the decision-making model "Decentral: Collaborative", decision-making agents are arranged in a ring network and have a predefined chance of forming a collective with one of their adjacent neighbors to engage in a joint search effort. This collaborative approach means that agents work together to search for and assess potential actions. Specifically, they share information about their current actions (*own-actions*) and any new potential actions they have identified. Together, they then decide on the most suitable actions to address their assigned tasks. Further information on this collaborative decision-making approach can be found in Sec. 7.3.2.

4.11 Observation

The collective performance of the organization within the simulation is shaped by the actions of all decision-making agents, as detailed in Section 4.2. This performance aligns with the NK-framework's tenets, described in submodel A (refer to Section 7.1). The performance levels are continuously tracked and documented throughout the simulation process. For comparability between different simulation runs, the recorded performances are normalized against the highest possible performance in a given landscape, adjusting the scale to a range from zero to one. After the simulation concludes, the performance data is presented as a sequential time series.

5 Initialization

Parameters To begin using the model, certain initial parameters must be fixed, as summarized in Tab. 2. These variable parameters are crucial for designing different scenarios to explore with

Table 2: Initialization parameters

Type	Variables	Notation	Submodel	Section
Variable parameters	Interdependence pattern	Pattern	A	7.1
	Incentive parameter	λ	C	7.3
	Decision-making mode	–	C	7.3
	Collaborative search probability	$\mathbb{P}$	C2	7.3.2
	Shock correlation	ρ	D	7.4
Constant parameters	Number of tasks	N	A	7.1
	Number of decision making agents	M	–	–
	Number of coordinating agents	–	–	–
	Observation periods	T	–	–
	Time until shock	τ	–	–
	Number of simulations	S	–	–

the model:

- **Interdependence patterns**: The model incorporates interdependencies among decision-making tasks, which are essential to its design. Researchers can specify the patterns of these interdependencies, which must then be input into the model. More information is available in submodel A, as discussed in Sec. 7.1.

- **Incentive parameter**: The model assumes a linear incentive structure for the hypothetical organization (see Sec. 4.4). Researchers need to determine and input the specific incentive parameter. Further explanations can be found in Sec. 7.3, which formalizes these mechanisms.

- **Decision-making model**: Researchers are required to fix and input the decision-making model that applies to the hypothetical organization. Additionally, if the decision-making mode "Decentral: Collaborative" is selected, the researcher must also define the **collaborative search probability**. Refer to Sec. 7.3.2 for more information.

- **Shock correlation**: The simulation can accommodate environmental shocks of varying severity, which are governed by correlation parameters. The severity level must be input by the researcher. Detailed information can be found in submodel D, described in Sec. 7.4.

Additional constant parameters are also detailed in Tab. 2, including their respective values.

Initialization of model elements The initialization of the model happens via the following submodels:

- *Landscape initialization*: The initialization of performance landscapes is detailed in submodel A (Sec. 7.1). This process requires the parameter N and an interdependence pattern as input. Here, N determines the dimensionality of the decision problem faced by the organization, and the interdependence pattern outlines the interactions between the N decision-making tasks, thus defining the complexity of the issue. This step initializes the state variable *performance-contributions*.

- *Agent initialization*: Initializing agents involves multiple steps:
  - **Actions**: Submodel B fixes the state variables *assigned-tasks*, *own-actions*, and *residual-actions* (see Sec. 7.2).

Table 3: Mapping between state variables and notation in submodels

Entity / variable name	Submodel	Notation	Meaning
Decision-making agents:			
assigned-tasks	B	$\mathbf{d}_m$	Part of the decision problem the entire organization faces that is assigned to a specific agent m.
decision-mode	C	–	Mode according to which an agent makes decisions about actions to tackle the decision-making tasks assigned to them.
own-actions	B and C	$\mathbf{d}_{mt}$	The actions taken by an agent m in a specific period t to tackle *assigned-tasks*.
residual-actions	C1, C2, C4 / C3	$\mathbf{d}_{-mt}$ / $\mathbf{d}^{\text{seq}}_{-mt}$	An agent m's information about the actions that the other agents have taken or will take to tackle their *assigned-tasks*.
own-proposals	C4	$\{\mathbf{d}^{(1)}_{mt}, \mathbf{d}^{(2)}_{mt}\}$	If the *decision-mode* is set to "Hierarchical: Proposal-mode," agent m evaluate and prioritize two viable actions to address their *assigned-tasks* and submit these suggestions to the coordinating agent in t.
Environment:			
performance-contributions	A	c_n / $s(c_n)$	Performance contributions of individual decisions to overall performance.

- **Decision-making mode**: The researcher fixes the state variable *decision-mode* as a constant for the duration of the simulation. The available decision-making modes are outlined in submodel C in Sec. 7.3.
- **Proposals**: The state variables *own-proposals* is initially unset and will be defined upon their first engagement and only if the decision-making mode "Hierarchical: Proposal-mode" is active within the stylized organization (see Sec. 7.3.2).

6 Input Data

The model does not use input data to represent time-varying processes.

7 Submodels

The following subsections describe in detail the main processes in the simulation model. A mapping between the state variables describes above and the notation used to formally describe the submodels is provided in Tab. 3.

7.1 A: Performance landscapes

The conceptual model for a stylized organization is grounded in the NK framework (Levinthal 1997; Wall and Leitner 2021; Blanco-Fernández, Leitner, and Rausch 2023). In this framework, agents navigate through performance landscapes, which consist of $N \in \mathbb{N}$ decisions interconnected by up to $K \in \mathbb{N} \leq N-1$ dependencies. The N-dimensional decision space is represented by the vector

$$\mathbf{d} = (d_1, \ldots, d_N) \; , \tag{3}$$

where each decision d_n can either be 0 or 1, for $n = 1, \ldots, N$. Consequently, there are a total of 2^N potential ways in which the decision problem can be solved within this space. Each decision d_n contributes to the overall performance, denoted by c_n. The performance contributions c_n are stored in the state variable *performance-contributions*. The contribution of any given decision d_n is influenced by up to K other decisions, indicating that each decision's performance impact,

c_n, can be affected by K interconnected decisions besides its own. This interdependency is mathematically expressed as

$$c_n = f(d_n, d_{i_1}, \ldots, d_{i_K}) \;, \tag{4}$$

with the set $\{i_1, \ldots, i_K\}$ being a subset of $\{1, \ldots, n-1, n+1, \ldots, N\}$. The overall performance for a decision vector $\mathbf{d}$ is then calculated by

$$c\left(\mathbf{d}\right) = \frac{1}{|\mathbf{d}|} \sum_{n=1}^{n=|\mathbf{d}|} c_n \;, \tag{5}$$

where $|\cdot|$ denotes the vector's length. Performance contributions, c_n, are assumed to be randomly drawn from a uniform distribution $U(0,1)$. Following this formulation, the overall performance of all possible 2^N solutions within the decision space is expected to be normally distributed due to the aggregation of individual contributions as specified.

7.2 B: Task decomposition

Task decomposition The decision problem $\mathbf{d}$ is broken down into M separate sub-problems, with each agent tasked with making decisions for a specific portion of the decisions in $\mathbf{d}$. The allocation of decisions to agents is carried out in a sequential and symmetrical manner, ensuring that each agent is accountable for $Q = N/M$ tasks. This distribution follows to the following rule:

$$\mathbf{d}_m = \left[d_{Q\cdot(m-1)+1}, \ldots, d_{Q\cdot m}\right] \;. \tag{6}$$

For instance, if there are $N = 15$ decisions to be made and $M = 5$ agents, then each agent is tasked with making 3 decisions. According to the allocation rule, agent 1 would handle decisions 1 to 3, agent 2 would cover decisions 4 to 6, and so on, ensuring an equitable distribution of decision-making responsibilities. For every agent m this allocation of tasks is stored in the state variable *assigned-tasks*

Initial configuration of decisions The choice of an agent m at a specific time step t is denoted by $\mathbf{d}_{mt}$. Since these choices are of a binary type, the decision at time step t is expressed as a bit-string. At the start of the simulation, a random bit-string of length N is generated, and the decisions of the agents at $t = 1$ are adjusted accordingly. This means that based on this initial random bit-string, the state variables *own-actions* and *residual-actions* are updated for all agents.

7.3 C: Decision-making models

The model encompasses four modes of decision-making, detailed in Secs. 7.3.1 to 7.3.4. During the initialization phase of a simulation, the decision-making mode for that specific run can be externally set by the modeler and is recorded in the state variable *decision-mode*.

The four modes of decision-making are based on the agents' objective functions, which were initially described in Sec. 4.4. For the sake of clarity and to enhance readability, let us here explicitly define the objective functions using the notation established in the descriptions of the submodels.

Decision-making agents' objective function The objective function for decision-making agents, for instance, agent m, is based on two key elements. The first element is the actions agent m selects to address their assigned tasks, represented by $\mathbf{d}_{mt}$ and recorded in the state variable *own-actions*. The second element encompasses the actions chosen by the other decision-making agents for their respective tasks, symbolized by $\mathbf{d}_{-mt}$ and stored in *residual-actions*. The performance outcomes of these components are computed according to Eq. 5. As discussed in Sec. 4.4, the organization applies a linear incentive mechanism that weighs the performances related to departmental objectives against those of other departments, modulated by the incentive parameter $\lambda \in (0,1) \subset \mathbb{R}$. With these considerations, the utility for decision-making agent m at time t is formulated as:

$$U(\mathbf{d}_{mt}, \mathbf{d}_{-mt}) = \lambda \cdot P(\mathbf{d}_{mt}) + (1-\lambda) \cdot P(\mathbf{d}_{-mt}) \rightarrow \max! \; . \tag{7}$$

Coordinating agent's objective function The coordinating agent's objective is to maximize the overall performance of the organization. As mentioned in Sec. 4.2, the organization's collective behavior results from the combination of individual agents' actions, which in turn influences the organization's overall performance. We represent these collective actions by $\mathbf{d}_t$, and the methods for calculating them are detailed subsequently. Therefore, the objective of the coordinating agent in t can be expressed as:

$$U\left(\mathbf{d}_t\right) = c\left(\mathbf{d}_t\right) \rightarrow \max!, \tag{8}$$

where the performance metric follows the formulation given in Eq. 5.

7.3.1 C1: Decentral: Silo-based

In situations characterized by silo-based decision-making, the information accessible to decision makers is restricted within their respective departments. This means that individuals gather information independently and refrain from sharing it with their colleagues. Instead, they base their decisions on the outcomes observed from the prior period, $\mathbf{d}_{-m(t-1)}$, stored in the state variable *residual-actions*.

In the neighbourhood of the decision currently in force within their domain, which was determined in the last period, $\mathbf{d}_{m(t-1)}$, an agent m identifies an alternative set of decisions, $\mathbf{d}^*_{mt}$. The term "neighbourhood" is specified by a Hamming distance of 1, indicating that $\mathbf{d}^*_{mt}$ is different from $\mathbf{d}_{m(t-1)}$ in only one bit.

Then, the decision currently implemented $\mathbf{d}_{m(t-1)}$ and the alternative solution $\mathbf{d}^*_{mt}$ are assessed based on their anticipated incremental benefits. Consequently, the decision for agent m in period t is determined by the following criterion:

$$\mathbf{d}_{mt} = \underset{\mathbf{d}' \in \{\mathbf{d}_{m(t-1)}, \mathbf{d}^*_{mt}\}}{\arg\max} U\left(\mathbf{d}', \mathbf{d}_{-m(t-1)}\right) \; . \tag{9}$$

For every agent, the result of Eq. 9 is used to update the state variable *own-action* during the main simulation loop. All agents independently determine their choices, and the collective solution to the decision problem arises from the aggregation of these individual decisions:

$$\mathbf{d}_t = \cup_{m=1}^{M} \mathbf{d}_{mt} \; . \tag{10}$$

All agents observe $\mathbf{d}_t$ after it was implemented and use this information to update the state variable *residual-actions*.

7.3.2 C2: Decentral: Collaborative

When decisions are made in a collaborative manner, agents are linked in a ring network. With a fixed probability $\mathbb{P}$, they perform collaborative decision making as described below, and with a probability of $(1-\mathbb{P})$, they perform silo-based decision-making, as outlined in Sec. 7.3.1.

If departments m and n engage in collaborative decision-making, they apply an adjacent hillclimbing method (Yuan and McKelvey 2004). Independently, they identify $\mathbf{d}^*_{mt}$ and $\mathbf{d}^*_{nt}$ within the neighbourhood of $\mathbf{d}_{m(t-1)}$ and $\mathbf{d}_{n(t-1)}$, respectively. This neighbourhood is defined by a Hamming distance of one. The combined residual solution (beyond their individual responsibilities) is represented as $\mathbf{d}_{-(mn)(t-1)} = \mathbf{d}_{(t-1)} \setminus (\mathbf{d}_{m(t-1)} \cup \mathbf{d}_{n(t-1)})$.

For the period t, agents m and n will collectively decide on a solution for their respective parts of the decision problem from the set of available pairs:

$$\mathbf{D}^{\text{collab}}_t = \left\{(\mathbf{d}_{m(t-1)}, \mathbf{d}_{n(t-1)}), (\mathbf{d}^*_{mt}, \mathbf{d}_{n(t-1)}), (\mathbf{d}_{m(t-1)}, \mathbf{d}^*_{nt}), (\mathbf{d}^*_{mt}, \mathbf{d}^*_{nt})\right\} . \tag{11}$$

This implies that they have the option to apply neither, one, or both of the solutions they found for their individual problems. They assess the pairs as shown in Eq. 11 according to the following principle:

$$(\mathbf{d}_{mt}, \mathbf{d}_{nt}) = \underset{(\mathbf{d}'_m, \mathbf{d}'_n) \in \mathbf{D}^{\text{collab}}_t}{\arg\max} \; U^{\text{adj}}(\mathbf{d}'_m, \mathbf{d}'_n, \mathbf{d}_{-(m,n)(t-1)}) , \tag{12}$$

where $U^{\text{adj}}(\cdot)$ returns the mean of the two individual utilities:

$$U^{\text{adj}}\left(\mathbf{d}_{mt}, \mathbf{d}_{nt}, \mathbf{d}_{-(m,n)t}\right) = \frac{1}{2} \cdot \Big(U(\mathbf{d}_{mt}, \underbrace{\mathbf{d}_{-mt}}_{\mathbf{d}_{-(m,n)t} \cup \mathbf{d}_{nt}}) + U(\mathbf{d}_{nt}, \underbrace{\mathbf{d}_{-nt}}_{\mathbf{d}_{-(m,n)t} \cup \mathbf{d}_{mt}})\Big) \tag{13}$$

All agents engaging in a collaborative search use the outcome of Eq. 12 to update their state variable *own-actions*. The overall solution to the decision problem $\mathbf{d}_t$ is computed in accordance with Eq. 10. Once $\mathbf{d}_t$ is fixed, all agents observe it and use this information to update their state variable *residual-actions*.

7.3.3 C3: Decentral: Sequential

In a setting where decisions are made in a decentralized manner with sequential communication, agents determine their actions one after another. For simplicity, let us assume that the sequence of decision-making follows the order of the agents' indices m. Therefore, agent $m = 1$ begins the decision-making process and informs the other $M - m$ agents about their chosen action. Consequently, agents must consider not only the residual decisions from the previous period, $\mathbf{d}_{-m(t-1)}$, but also the information obtained through sequential communication. Thus, whenever agents are informed about another agent's action, they update their state variable *residual-actions*. In detail, the vector containing agent m's residual decisions in period t is structured as follows:

$$\mathbf{d}^{\text{seq}}_{-mt} = (\mathbf{d}_{1t}, \ldots, \mathbf{d}_{(m-1)t}, \mathbf{d}_{(m+1)(t-1)}, \ldots, \mathbf{d}_{M(t-1)}) . \tag{14}$$

This indicates that agent 1 depends entirely on the solutions adopted in the prior period to identify the residual decisions for time t. In contrast, agents who make their decisions later in the sequence progressively receive more information regarding their residual decisions.

During the decision-making process, agents identify alternative options $\mathbf{d}^*_{mt}$ for their decision problem, which are within a Hamming distance of one from the solution currently in place

$\mathbf{d}_{m(t-1)}$. They then assess these two choices as follows:

$$\mathbf{d}_{mt} = \underset{\mathbf{d}' \in \{\mathbf{d}^{\mathrm{d}}_{m(t-1)}, \mathbf{d}^*_{mt}\}}{\arg\max} \; U\left(\mathbf{d}', \mathbf{d}^{\mathrm{seq}}_{-mt}\right) \;. \tag{15}$$

All agents use the result of Eq. 15 to update their state variable *own-actions*. The overall solution to the decision problem $\mathbf{d}_t$ is determined according to Eq. 10. All agents observe the overall solution after it was computed and they use this information to update their state variable *residual-actions*.

7.3.4 C4: Hierarchical: Proposal-mode

In the proposal mode, agents identify an alternative solution $\mathbf{d}^*_{mt}$ to their part of the decision problem. This solution is found within a neighbourhood defined by a Hamming distance of one from the solution currently in effect, $\mathbf{d}_{m(t-1)}$. Subsequently, agents use information regarding the residual decisions from the prior period, $\mathbf{d}_{-m(t-1)}$, to assess both $\mathbf{d}^*_{mt}$ and $\mathbf{d}_{m(t-1)}$, as outlined in Eq. 9. Specifically, they rank the solution that is expected to yield greater utility first

$$\mathbf{d}^{(1)}_{mt} = \underset{\mathbf{d}' \in \{\mathbf{d}_{m(t-1)}, \mathbf{d}^*_{mt}\}}{\arg\max} \; U\left(\mathbf{d}', \mathbf{d}_{-m(t-1)}\right) \;, \tag{16}$$

and they rank the other solution second $\mathbf{d}^{(2)}_{mt} = \{\mathbf{d}_{m(t-1)}, \mathbf{d}^*_{mt}\} \setminus \{\mathbf{d}^{(1)}_{mt}\}$. Every agent follows this procedure updates the state variable *own-proposals* with their ranked solutions. Subsequently, they transmit their ordered choices to a central (or coordinating) agent that could, for example, represent the headquarters of an organization.

The central agent assesses the prioritized suggestions in the following manner: It concatenates all proposals ranked first and all proposals ranked second based on

$$\mathbf{d}^{(j)}_t = \cup_{m=1}^{M} \mathbf{d}^{(j)}_{mt} \;, \;\; \text{where} j \in \{1, 2\} \;, \tag{17}$$

consolidates the merged proposals into $\mathbf{P}_t = (\mathbf{d}^{(1)}_t, \mathbf{d}^{(2)}_t)$. Ultimately, from these two potential choices (resulting from the concatenation outlined above), the central agent selects the solution that delivers the highest performance to be implemented:

$$\mathbf{d}_t = \underset{\mathbf{p}' \in \mathbf{P_t}}{\arg\max}\, c(\mathbf{p}'). \tag{18}$$

The performance is calculated according to Eq. 5. It is important to note that the central agent must adhere to the agents' proposals, aligning with the principles of decentralization. Due to the procedural design, the proposals naturally include the option for maintaining the current state, which means individual agents have the option to prefer keeping their existing position in the performance landscape. The central agent's main task is to select the most beneficial combination from the individual proposals made by the agents. Once this decision is made, agents can observe the decision selected by the central agent and use this information to update their state variables *own-actions* and *residual-actions*.

7.4 D: Shocks to performance landscapes

The model considers that the stylized organization undergoes an environmental shock, which changes the performance landscape by altering the performance contributions c_n. The intensity of these shocks is modulated through the parameter ρ, which ranges within $(-1, 1) \subset \mathbb{R}$.

Upon the occurrence of a shock, the performance contributions are recalculated, leading to a transformed performance landscape. The post-shock performance contributions $s(c_n)$ maintain a correlation with the pre-shock contributions c_n, following a method introduced by Demirtas (2014): random values $v_n \sim U(0, 1)$ and $w_n \sim B(a, 1)$ are generated, where the parameter a of the Beta distribution is determined by ρ as:

$$a = \frac{1}{2}\left(\sqrt{\frac{49 + \rho}{1 + \rho}} - 5\right) \ . \tag{19}$$

Subsequently, v_n and w_n are used to calculate the correlated post-shock performance contribution $s(c_n)$ as follows:

$$s(c_n) = \begin{cases} |w_n - c_n| & \text{if } v_n < 0.5 \\ 1 - |1 - w_n - c_n| & \text{if } v_n \geq 0.5 \ . \end{cases} \tag{20}$$

Here, ρ acts as a correlation coefficient: as ρ approaches 1, the impact of the shock diminishes, resulting in the post-shock performance landscape bearing a closer resemblance to the landscape before the shock. The post-shock performances computed according to Eq. 20 are used to update the state variable *performance-contributions*.

References

Baumann, Oliver, Jens Schmidt, and Nils Stieglitz. 2019. "Effective search in rugged performance landscapes: A review and outlook." *Journal of Management* 45 (1): 285–318.

Blanco-Fernández, Darío, Stephan Leitner, and Alexandra Rausch. 2023. "The effects of group adaptation on task performance: An agent-based approach." *PloS ONE* 18 (8): e0290578.

Cormen, Thomas H, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms.* MIT press.

Demirtas, Hakan. 2014. "Generating bivariate uniform data with a full range of correlations and connections to bivariate binary data." *Communications in Statistics-Theory and Methods* 43 (17): 3574–3579.

Dosi, Giovanni, and Luigi Marengo. 2007. "On the evolutionary and behavioral theories of organizations: A tentative roadmap." *Organization Science* 18 (3): 491–502.

Epstein, Larry G, Massimo Marinacci, and Kyoungwon Seo. 2007. "Coarse contingencies and ambiguity." *Theoretical Economics* 2 (4): 355–394.

Frey, Bruno S. 2005. ""Just forget it."Memory distortions as bounded rationality." *Mind & Society* 4:13–25.

Galbraith, Jay R. 1974. "Organization design: An information processing view." *Interfaces* 4 (3): 28–36.

Garicano, Luis, Timothy Van Zandt, et al. 2013. "Hierarchies and the civision of labor." In *Handbook of Organizational Economics,* edited by Robert Gibbons and John Roberts, 604–654. Princeton: Princeton University Press.

Gigerenzer, Gerd, and Reinhard Selten. 2002. *Bounded rationality: The adaptive toolbox.* MIT press.

Grimm, Volker, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K Heinz, Geir Huse, et al. 2006. "A standard protocol for describing individual-based and agent-based models." *Ecological modelling* 198 (1-2): 115–126.

Grimm, Volker, Steven F Railsback, Christian E Vincenot, Uta Berger, Cara Gallagher, Donald L DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, Juergen Groeneveld, et al. 2020. "The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism." *Journal of Artificial Societies and Social Simulation* 23 (2).

Kauffman, Stuart, and Simon Levin. 1987. "Towards a general theory of adaptive walks on rugged landscapes." *Journal of Theoretical Biology* 128 (1): 11–45.

Kretschmer, Tobias, and Phanish Puranam. 2008. "Integration through incentives within differentiated organizations." *Organization Science* 19 (6): 860–875.

Lawrence, Paul R, and Jay W Lorsch. 1967. "Differentiation and integration in complex organizations." *Administrative Science Quarterly,* 1–47.

Leitner, Stephan. 2023. "Collaborative search and autonomous task allocation in organizations of learning agents." In *Advances in Social Simluation,* edited by Flaminio Squazzoni, 345–357.

Levinthal, Daniel A. 1997. "Adaptation on rugged landscapes." *Management Science* 43 (7): 934–950.

Nadler, David, Michael Tushman, and Mark B Nadler. 1997. *Competing by design: The power of organizational architecture.* Oxford University Press.

Siggelkow, Nicolaj, and Jan W Rivkin. 2005. "Speed and search: Designing organizations for turbulence and complexity." *Organization science* 16 (2): 101–122.

Simon, Herbert A. 1955. "A behavioral model of rational choice." *The Quarterly Journal of Economics,* 99–118.

Simon, Herbert A. 1956. "Rational choice and the structure of the environment." *Psychological review* 63 (2): 129.

Simon, Herbert A. 1962. "The architecture of complexity." *Proceedings of the American Philosophical Society* 106 (6): 467–482.

Simon, Herbert A. 1967. "The business school a problem in organizational design." *Journal of management Studies* 4 (1): 1–16.

Tracy, William M, Dmitri G Markovitch, Lois S Peters, BV Phani, and Deepu Philip. 2017. "Algorithmic representations of managerial search behavior." *Computational Economics* 49:343–361.

Vriend, Nicolaas J. 1996. "Rational behavior and economic theory." *Journal of Economic Behavior & Organization* 29 (2): 263–285.

Wall, Friederike. 2016. "Agent-based modeling in managerial science: an illustrative survey and study." *Review of Managerial Science* 10 (1): 135–193.

Wall, Friederike. 2018. "Emergence of task formation in organizations: Balancing units' competence and capacity." *Journal of Artificial Societies and Social Simulation* 21 (2).

Wall, Friederike, and Stephan Leitner. 2021. "Agent-based computational economics in management accounting research: Opportunities and difficulties." *Journal of Management Accounting Research* 33 (3): 189–212.

Wu, Jiunyan, Takaki Ohya, and Tomoki Sekiguchi. 2023. "Applications of agent-based modeling and simulation in organization management: a quarter-century review through bibliometric mapping (1998–2022)." *Computational and Mathematical Organization Theory,* 1–31.

Yuan, Yu, and Bill McKelvey. 2004. "Situated learning theory: Adding rate and complexity effects via Kauffman's NK model." *Nonlinear Dynamics, Psychology, and Life Sciences* 8 (1): 65–101.

Zhou, Yue Maggie. 2013. "Designing for complexity: Using divisions and hierarchy to manage complex tasks." *Organization Science* 24 (2): 339–355.