universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Influence of the System Size on the Rate of Combinatorial Innovation Production“

verfasst von / submitted by

Nino Lauber, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2018 / Vienna, 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 876
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Physik
Betreut von / Supervisor:	Univ.-Prof. Mag. DDr. Stefan Thurner

Abstract

Innovation can be formulated as a combinatorial process where objects from a set are combined to create new objects. These are in turn added to the set of objects and can again be recombined to create further new objects. In some instances this is a social process in which the recombination is performed by a group of people working together which can be referred to as a group of collaborators. This social aspect of innovation is especially evident in the the creation of new knowledge or methods which mostly happens when people exchange and combine their current knowledge and ideas to create new knowledge. For this to take place communication between collaborators is necessary. In social psychology, however, one can often observe a phenomenon called the Ringelmann effect which states that as groups grow in size the increasing coordination and communication overhead reduces the rate at which successful communication among the collaborators takes place. This ultimately leads to less innovative exchanges and a decline in novelty production. Within this thesis a simple combinatorial innovation model, simulating the recombination of knowledge elements by a set of agents, is implemented and studied. Additionally, an analysis of data from open-source-software projects is conducted. The aim is to study the emergence of the Ringelmann effect in the model and discuss the findings with regard to the data set.

Zusammenfassung

Innovation kann als ein kombinatorischer Prozess formuliert werden, bei dem Objekte aus einer Menge kombiniert werden, um neue Objekte zu erzeugen. Diese werden anschließend zu der Menge an Objekten hinzugefügt und können wiederum zu weiteren neuen Objekten rekombiniert werden. In manchen Fällen ist dies ein sozialer Prozess, in dem diese Rekombinationen von einer zusammenarbeitenden Personengruppe, einer Gruppe von Mitarbeitern, ausgeführt wird. Dieser soziale Aspekt von Innovation ist bei der Entstehung neuem Wissens oder Methoden besonders evident, welche in der Regel stattfindet, wenn Personen ihr aktuelles Wissen und ihre Ideen austauschen und kombinieren, um neues Wissen zu erzeugen. Damit dies geschieht ist Kommunikation zwischen den Mitarbeitern notwendig. In der Sozialpsychologie kann allerdings häufig ein Phänomen beobachtet werden, welches als Ringelmann Effekt bezeichnet wird. Dieses besagt, dass wenn solche Gruppen anwachsen, der zunehmende Koordinations- und Kommunikationsbedarf die Rate reduziert, mit der erfolgreiche Kommunikation zwischen den Mitarbeitern stattfindet. Dadurch kommt es zu weniger innovativem Austausch und einer Abnahme in der Produktion von Neuem. Innerhalb dieser Masterarbeit wird ein simples, kombinatorisches Innovations-Model implementiert, welches die Rekombination von Wissen durch eine Gruppe von Agenten simuliert. Des weiteren wird eine Analyse von empirischen Daten aus Open-Source-Software Projekten durchgeführt. Ziel ist es, das Auftauchen des Ringelmann Effektes im Model zu studieren und diese Erkenntnisse unter Berücksichtigung der Daten zu diskutieren.

Contents

Chapter 1

Introduction

Etymologically, the term *innovation* origins from Latin and roughly translates to *change* or *renewal*. The Oxford English Dictionary further defines it as: "the alteration of what is established by the introduction of new elements or forms [...]"[1]. One of the first definitions of innovation was given by Joseph Schumpeter in an economic context [2, 3, 4]. In general, an innovation can be the introduction of a new technology to a market, or the improvement of existing ones; the arrival of new species or genetic traits in an ecosystem; or the adaption of new theories and paradigms by a society.

Typically *new things* come into existence by combining entities that already exist at a given time. This phenomenon is especially evident in the creation of new technologies, as emergence of new technologies often originates from a recombination of already existing technologies. In 2006, Brian Arthur *et al.* investigated the evolution of technology with a simple computer model by focusing on this combinatorial aspect of innovation [5, 6]. In other instances one can observe that while some innovations can lead to a large number of further new entities, others can cause the replacement or destruction of already existing ones, a process that Schumpeter described as *creative destruction* [3, 4]. In more recent years this interplay between constructive and destructive aspects of innovation has been extensively studied by Thurner *et al.* within several heuristic models [7, 8, 9, 10]. Furthermore, the introduction of one novelty to a system can open up further possibilities for innovations. The space of possible innovations thus expands. In

2016 Loreto *et al.* implemented a probabilistic model that reproduces this sample space expanding aspect of innovations [11, 12]. Thus it can be seen that several approaches have been made in modelling the dynamics of innovation, each focusing on a different aspect of the phenomenon.

There is, however, a further aspect of innovation that one has to pay attention to as well. Innovation as the creation of new knowledge, in the form of new ideas or paradigms, is, in general, a social process. Even though history provides several examples of the archetypal *singular genius* coming up with new insights all by themselves, in recent history cooperation has become increasingly important in the creation of new knowledge. A trend that can be especially seen among scientists [13]. Therefore, in order to create new concepts it is necessary that people exchange ideas and recombine their knowledge. In other words, sometimes two ideas need to "collide" with each other in order to be recombined to a new and innovative idea [14]. This social aspect of knowledge-creation has already been established by the physician and philosopher of science Ludwig Fleck, who defined the concept of the *thought collective* as: "[...] a community of researchers who interact collectively towards the production or elaboration of knowledge using a shared framework of cultural customs and knowledge acquisition [15]." Fleck's concepts have later been further elaborated by the famous physicist, historian and philosopher of science Thomas Kuhn in his explanation of *paradigm shifts* [16].

One can refer to such a group of people working together as a *group of collaborators*. It could be assumed that the interaction of a larger number of collaborators would affect the innovation processes positively due to the higher number of possible knowledge exchanges. However, this assumption ignores the fact that including a larger group of people into the process can result in higher *coordination* and *communication overhead* that can ultimately hinder the free and efficient exchange of ideas. This phenomenon is known in social psychology as *Ringelmann effect* or *social loafing* [17, 18, 19]. In the specific context of software development this effect is also known as *Brooks' law*. It states that if additional developers are added to a project that is already behind schedule its moment of completion is postponed even further [20]. Much work has been invested to test the validity of these related effects by analysing data from either open-source-software (OSS) projects or academic research groups [21, 22, 23, 24].

This thesis follows up on this work and further studies the influence of group size on the rate of combinatorial innovation production. The central hypothesis that is tested here is the following:

Hypothesis: As a group of collaborators grows in size, the increasing coordination and communication overheads reduce the rate of useful communication among its members for knowledge production. This ultimately leads to less innovative exchanges and a decline in novelty production.

This hypothesis is approached in two steps. As mentioned above, several approaches exist to model the process of innovation, all focusing on different aspects. Based on these models a simple agent-based model is defined that replicates how new knowledge is created within a group of collaborators via exchange and recombination and the emergence of the Ringelmann-effect is tested within the model. In a second step data of OSS projects obtained from the platform Git-Hub is used to look for evidences for the influence of Brooks' law in an empirical data set. Furthermore, the results obtained from this data set are compared with the results from the agent-based model.
The remainder of this thesis is structured as follows. In Chapter 2, the general framework of complex adaptive interactions, evolutionary dynamics and innovation is outlined and a short overview of the most important innovation models that inspired or bear relevance to this work is given. Secondly, in Chapter 3, a description of the model as well as the empirical data that is used to test the adequacy of the model is provided. Thereafter, in Chapter 4, the main findings of the model as well as a comparison between the results of the model and the data is presented. Finally, in Chapter 5, the results are discussed and, in Chapter 6, a conclusion is drawn.

Chapter 2

Theory

2.1 General Framework and Notation

2.1.1 Complex Adaptive Interaction Networks

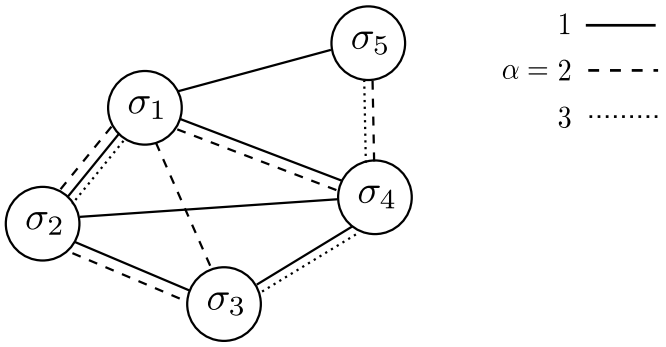


Figure 2.1: A set of five entities is represented as nodes in a network. Each node contains the state σ_i of an entity i. The interactions among the entities are represented as links between the nodes. There are three possible kinds of interactions, each one indicated by a different type of link. This results in what is called a multiplex network of interactions (Figure recreated from [25]).

In physics one can characterize an entity i at time t by a state $\sigma_i(t)$ (e.g. its position, momentum, spin, etc.). Entities might be subjected to interactions mediated by forces (e.g. gravitational force, electromagnetic force, etc.) that act on the states and thereby change them. To describe how the forces change $\sigma_i(t)$ with

time, i.e. the time evolution of the state, one can often use a differential equation which can be solved with suitable initial or boundary conditions. This way one obtains the trajectory of the state that describes $\sigma_i(t)$ at any time t. Typically, one does not have a single entity but rather an ensemble of N entities (for example a gas or a liquid, etc.) where the whole system at time t is then characterized by the state vector $\vec{\sigma}(t) = (\sigma_1(t), \ldots, \sigma_N(t))$. In such ensembles the entities usually interact among each other via internal, pairwise forces that act the same way between all entities and only differ in the strength of the interactions. Often it is still possible to describe the time-evolution of the whole ensemble with a system of coupled differential equations.

One can try to use a similar formalism to describe systems of more complex or abstract entities that are characterized by more general states and interactions, e.g. a group of people and their opinions [26, 27, 28], an ecosystem of species and their abundances [29, 30], a system of banks and their capital [31], etc. In contrast to the ensembles described above, these systems possibly include variables that can only have discrete values. Moreover, the interactions in such systems are *specific* in the sense that only certain pairs of entities interact with each other in a particular way. One can formulate such interactions as a network [32] the topology of which is given by the matrix $M_{ij}(t)$: an entry $M_{ij} \neq 0$ describes the strength of an interaction between entities i, j while $M_{ij} = 0$ indicates that there is no interaction happening. Furthermore, such systems are not limited to one type of interaction. In a social network, for example, there can be multiple ways people can interact. They can be friends, enemies, coworkers, etc [25]. As seen in Fig.(2.1) the interaction network has different types of links, which can be represented as different network layers. If the interaction type is denoted with α, the topology of this multi-layered interaction network can be described with a "tensor" M^{α}_{ij}: an entry $M^{\alpha}_{ij} \neq 0$ describes the strength of an interaction of type α between entities i, j while $M^{\alpha}_{ij} = 0$ indicates that there is no interaction of that type happening between i and j [25]. Within this formalism one can formulate the time evolution of the states as a (symbolic) equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma_i(t) \sim F(M^{\alpha}_{ij}(t), \sigma_j(t)) \tag{2.1}$$

Assuming one knows the exact topology of M_{ij}^{α} as well as the function $F(M_{ij}^{\alpha}(t), \sigma_j(t))$ and all state variables are continuous this set of differential equations could still be solved.
However, the interactions in systems described within this formalism are usually not static. For example: people can form new friendships or end them, new species can enter an ecosystem and change the environment, etc. In other words, new links may be formed or existing links may vanish. The change of the topology of $M_{ij}^{\alpha}(t)$ depends on the existing links as well as on the states. Therefore, in addition to Eq.(2.1), one needs a set of coupled differential equations describing the time evolution of the topology $M_{ij}^{\alpha}(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t} M_{ij}^{\alpha}(t) \sim G(M_{ij}^{\beta}(t), \sigma_j(t)) \tag{2.2}$$

In general it is not possible to come up with an analytical way of solving Eq.(2.1) together with Eq.(2.2) as all the relevant parameters to formulate the functions F and G, or the complete set of relevant entities and interactions to formulate $\vec{\sigma}(t)$ and $M_{ij}^{\alpha}(t)$ are usually not known. It is possible however to make simplifications and a priori assumptions in order to describe the system, or one may try to model the change of states and the change of the networks algorithmically.
Together Eq.(2.1) and (2.2) symbolically describe a complex adaptive system of entities that are characterized by their states. The interactions or relations between the entities are described in the form of multi layered networks. The state changes of the entities depend on the interaction network which in turn depends on the states. In other words the state vector of the system co-evolves with the interaction topology.

2.1.2 Evolutionary Processes

As established in the previous section, evolution in physics means to describe how a state under the influence of external and/or internal forces changes over time. Evolutionary processes, however, are more general in the sense that they describe how a system of entities continuously creates or destroys other entities through interactions among themselves or with an environment. The composition of the system changes in a dynamical process that is typically irreversible and non-ergodic

[33]. One immediately thinks of a biological context here, where the entities are co-evolving species [29, 30]. However, as sketched above, one can look at more general systems of entities, e.g. a mixture of chemicals [34, 7], a set of technologies [5], etc., the choice of this context determines constraints acting on or within the system (mass conservation, supply and demand, etc.).

Evolutionary processes can be described as a three-step algorithm [9, 10]: (1) a new entity is created or assembled and enters the system. (2) The new entity interacts with the environment of the system, composed of all the already existing entities. Through these interactions the new entity is either destroyed or proliferates. (3) If the new entity survives, it becomes part of the environment itself and the environment has been, often irreversibly, changed.

It is possible to describe the abundance of an entity i at time t with a scalar state $\sigma_i(t)$ which is a real and non-negative number. $\sigma_i(t) > 0$ then indicates that entity i exists at time t while $\sigma_i(t) = 0$ indicates that it does not exist. To describe the possible types of interactions among the entities, one can again use a multi-layered interaction network $M^{\alpha}_{ij}(t)$. Since in evolutionary processes interactions can exist between more than just pairs of entities, the interaction topology can be generalized as $M^{\alpha}_{ijk\ldots}(t)$. The dynamics implemented in this three-step algorithm is then essentially the co-evolutionary dynamics described by Eq.(2.1) and Eq.(2.2), where the environment changes the abundances of the species while the abundances of the species change the environment.

One may attempt to come up with equations like Eq.(2.1) and Eq.(2.2) to formally describe evolutionary processes. However, as stated in the previous section, one usually lacks the necessary knowledge about all relevant system parameters. Furthermore, evolutionary processes are in general open-ended as more and more new entities are created and become part of the system, which creates new possibilities for the entities to interact. As a consequence the state vector of all entities as well as the sample space of interactions expand over time. Thus the system is never in equilibrium.

Nevertheless, simplifications can be made in order to formally describe evolutionary processes. One may look at a finite system of N interacting entities whose states $\sigma_i(t) \in [0, 1]$ are the relative abundances with $\sum_i \sigma_i(t) = 1$. Basically one can define four basic types of evolutionary interactions: replication [35], when an

entity simply reproduces itself; competition [36, 37] when an entity drives another entity out of the system; mutation [38] when an entity spontaneously mutates to another entity. In any case, the most fundamental interaction for our purposes is recombination. As illustrated in Fig.(2.3a), it essentially describes how a new entity comes into existence through the combination of already existing entities. The dynamics of $\sigma_i(t)$ is then governed by the catalytic network equation [34]:

$$\dot{\sigma_i}(t) = \sum_{j,k} M_{ijk}\sigma_j(t)\sigma_k(t) - \sigma_i(t)\phi(\sigma), \quad \phi = \sum_{ljk} M_{ljk}\sigma_j(t)\sigma_k(t) \tag{2.3}$$

where ϕ is the normalization term. M_{ijk} is the *rule table* where an entry $M_{ijk} \neq 0$ indicates that there exists a recombination rule like Fig.(2.3a) that j, k can be recombined to form i and gives the rate this recombination process is happening with. $M_{ijk} = 0$ indicates that no such rule exits. One can see from Eq.(2.3) and Fig.(2.3a) that such recombination rules can only be active if both species are present in the system. It can be argued that recombination is the most general form of evolution [39, 40]. A system that is characterized by this dynamics is called a *combinatorial evolutionary system.*

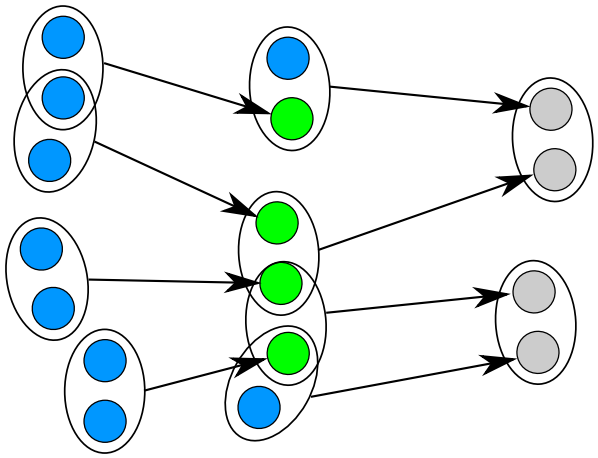
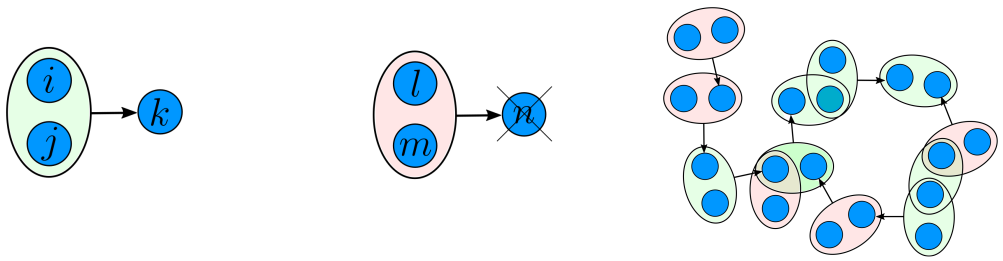


Figure 2.2: The *actual* is the set of all entities that are already present in the system (blue nodes). The *adjacent possible* is the set of entities that can be reached within one step by combining entities from the actual (green nodes). All entities that can be reached in further time steps through recombination are the *far possible* [41] (gray nodes).

In the context of combinatorial evolution, one can define and understand the concept of the *adjacent possible* [42]. As illustrated in Fig.(2.2) the adjacent possible is the set of potential entities that can be created at $t + dt$ from all the entities present in the system at time t. The adjacent possible is a kind of "creative potential" of the "near future". As new entities are created through the combination of existing entities new possibilities for pairing entities are formed. Thus, the composition of the adjacent possible changes with the *actual*. In other words: the adjacent possible co-evolves with the entities of the system. Most approaches to model evolutionary interactions are about the exploration of the adjacent possible. In the following the most important examples are summarized.

2.2 Related Work

2.2.1 Constructive and Destructive Interactions in Innovation Dynamics



(a) **(b)** **(c)**

Figure 2.3: (**a**) A constructive interaction, where entities i, j combine to produce entity k. (**b**) A destructive interaction where entities m', n' together destroy entity i'. (**c**) A system containing constructive and destructive interactions (Figures recreated from [10, 9]).

As established above, a set of N entities interacting via recombination can be analytically described using Eq.(2.3). These systems usually do not only have constructive interactions, Fig.(2.3a), where two entities are recombined to create a new one. As seen in Fig.(2.3b), there can also be destructive interactions, where

two entities together destroy an existing entity e.g. by suppressing its production or creating a new entity that drives the existing one out of the system. Thus, an entity can be involved in constructive as well as destructive interactions. One can see in Fig.(2.3c) that the combined interactions can lead to a complex dynamics where constructive interactions reinforce destructive interactions and vice versa. For instance, whenever a species exercising a destructive rule is created, this rule is in turn activated. On the other hand, destroying an entity exercising a destructive rule deactivates this rule, which in turn promotes the entity suppressed by this very rule. In 2010, Thurner *et al.* formulated a model that is an excellent heuristic means to understand complex dynamics of this kind [10].

In this simple model they assume that the abundance σ_i of an entity i at time t is described by a binary variable: $\sigma_i(t) = 1$ indicates that the entity has an abundance > 0, i.e. it is present in the system. $\sigma_i(t) = 0$ indicates that the entity has an abundance of 0, i.e. it is not present in the system. The whole system at time t is represented by the state vector $\vec{\sigma}(t) = (\sigma_1(t), \ldots, \sigma_N(t))$.

The constructive interactions are captured in a rule table M^+_{ijk}. Assuming that there are multiple ways an entity k can be produced, there is a constructive rule density r^+ that quantifies how many productive pairs for a certain entity exist on average. Likewise the destructive interactions are captured in a rule table M^-_{ijk} with a destructive rule density r^-. For simplicity, the entries of $M^\pm_{ijk}$ are binary as well, where $M^\pm_{ijk} = 1$ indicates that there exists a constructive/destructive interaction between the entities i, j, k, while $M^\pm_{ijk} = 0$ indicates that there are no such interactions. The rule tables are randomly set a priori with the probability of an entry $M^\pm_{ijk}$ being 1 as $P(M^\pm_{ijk} = 1) = r^\pm / \binom{N}{2}$. The number of active productive/destructive sets for species i at any time t is: $N^\pm_i(t) = \sum_{jk} M^\pm_{ijk}\sigma_j(t)\sigma_k(t)$. Furthermore, one assumes that a randomly selected number σ_0 of entities is present in the system at $t = 0$.

It can be shown that in a system that is only based on constructive interactions, the system reaches a steady state of populated entities. By varying σ_0 and r^+, one can observe a phase transition in the production of new entities, from only a few to the maximum number of possible entities populated [7]. Likewise, in a system only characterized by destructive interactions, one observes a transition where the system, depending on σ_0 and r^-, changes from a completely populated

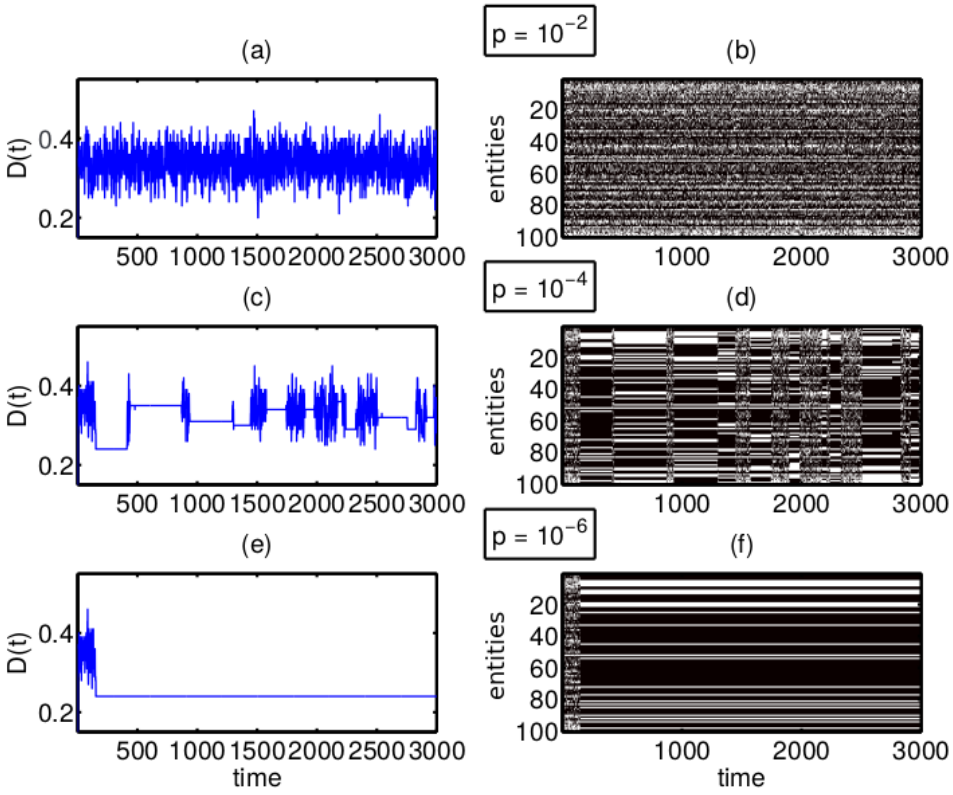


Figure 2.4: The left side shows the *diversity* $D(t)$ measured at each time step for $N = 100, r^+ = 10, r^- = 15$. The right side shows the individual trajectories of each entity: a white cell indicates that an entity exists at time t, a whereas a black cell indicates that it does not exist. $D(t)$ and the individual trajectories are shown for **(a,b)** a high innovation rate p, **(c,d)** an intermediate innovation rate p, **(e,f)** a low innovation rate p (Figure taken from [10]).

to a sparsely populated state of remaining species (the system never becomes fully depopulated, some "substrate" remains) [8, 9]. To describe the combined dynamics of constructive and destructive interactions three simple update rules for an entity i with $\sigma_i(t)$ are formulated [10]:

If $N_i^+ > N_i^-$, the productive outweigh the destructive influences on i, entity i can be created: $\sigma_i(t+1) = 1$

If $N_i^+ < N_i^-$, the destructive outweigh the productive influences on i, entity i is destroyed: $\sigma_i(t+1) = 0$

If $N_i^+ = N_i^-$, the destructive and productive influences on i are in equilibrium, entity i remains unchanged: $\sigma_i(t+1) = \sigma_i(t+1)$

Additionally it is assumed that there is a chance for spontaneous innovations or destructions to happen. Thus with a probability p a state $\sigma_i(t)$ can be flipped. The full algorithm of this model is [10]:

1. Select a random entity i.
2. Compute $N^{\pm}(t)$. If $N^{+}(t) - N^{+}(t) > 0$ set $\sigma_i(t+1) = 1$. If $N^{+}(t) - N^{+}(t) < 0$ set $\sigma_i(t+1) = 0$. If $N^{+}(t) - N^{+}(t) = 0$ set $\sigma_i(t+1) = \sigma_i(t+1)$.
3. Select a random entity j and with probability p switch the state from $\sigma_j(t) = 1(0)$ to $\sigma_j(t+1) = 0(1)$.
4. Repeat (1)-(3) until all entities have been updated once (random sequential update) then advance to the next time step.

On can describe the *diversity* of the system at t as $D(t) = \frac{1}{N}\sum_{i=1}^{N}\sigma_i(t)$, the mean fraction of active species. In Fig.(2.4c), one can see that $D(t)$ changes between stable phases where the diversity remains almost unchanged and chaotic phases where $D(t)$ heavily fluctuates. One can furthermore see that the recovery time from such chaotic phases as well as the life time of stable phases is controlled by the parameter p. In the most extreme cases the diversity either reaches a global steady state (low value of p) or does not reach a stable phase at all (high value p). In general, this model is a good heuristic approximation to the complex combined dynamics of constructive and destructive combinatorial interactions among entities. In an economical context, where the entities are goods and services, the model furthermore provides a good way to algorithmically describe of the Schumpeterian dynamics of creative destruction: the emergence of certain innovations can drive out a large number of existing goods, in so called cascades of destruction. Thus, phases of economic stability (plateaus in $D(t)$) are superseded by phases of large changes (fluctuations in $D(t)$) [3, 4].

2.2.2 Innovation, Urn Processes and Expanding Sample-Spaces

As mentioned above, evolutionary processes and the creation of novelties and innovations are generally open ended. As seen in Fig.(2.2) each newly created entity opens up new ways to be recombined with other new entities or already existing ones. The sample space of possible innovations therefore keeps growing or, in other

words, the adjacent possible is expanding.

A good way of modelling expanding sample spaces is to describe the emergence of innovations in a probabilistic way. Simon models are historically among the first to do so [43]. The basic Simon model describes a stream of tokens S that at $t = 0$ initially consists of a single token. Every further time step t, an additional token is added to S. With a probability p, this is a new token, while with probability $(1 - p)$ the token is randomly chosen from the already existing ones in S. The model produces an expanding sample space as the number of possible tokens that one can select is growing by adding new tokens to S. The model also shows a so called the rich get richer dynamics as tokens that appear more often in S are more probable to be selected again. However, one may note that new elements enter the system simply with a constant rate p. In other words, the innovation rate is assumed a priori and therefore is a model parameter and not an endogenous variable.

Variations of the Simon model have been formulated, e.g. the Zanette-Montemurro model, where the innovation rate p is not constant but varies as a time dependent function [44], or the Dorogostev-Mendes model, where each token has an ageing factor that reduces the probability of being selected [45]. Nevertheless, in both variations the innovation rate is hard wired in the system. Another approach to model expanding sample spaces is by using Pólya-Urn processes which describe an urn containing balls of different colours. Whenever a ball is drawn at random, it is put back into the urn with a certain amount of new balls of the same colour [46]. In 2014, Loreto *et al.* modelled innovation by using a variation of such a Pólya-Urn process [12, 11]. This simple model like the Simon models, describes the emergence of innovations in a probabilistic way. Again, a sequence S is constructed by drawing elements from an urn U which initially contains N_0 distinct elements. U and S increase their content and length at each time step the following way [12]:

1. randomly select an element from U and add it to S
2. put the element and ρ copies of it back into U
3. if the extracted element has not yet appeared in S, add $\nu + 1$ new distinct elements to U as well

An innovation in this model is an element drawn randomly from U that appears for the *first* time in the sequence S. For a sequence containing N elements the

number of distinct elements is given as $D(N)$ and its growth quantifies the rate at which innovations are happening. If $\mu < \rho$ the growth of $D(N)$ is [12]:

$$D(N) \sim N^{\frac{\nu}{\rho}} \tag{2.4}$$

Unlike in the Simon models innovations here do not take place with a given rate, instead the probability for an innovation to happen is an intrinsic property of the urn $U(t)$. Each innovation opens up new possibilities for further innovations by adding $\nu + 1$ new distinct elements to U. However, these are just *possible* innovations. In other words: for one of these new elements to actually be an innovation, it has to be drawn from U at some later time step. Theoretically, a distinct element that was added to U might never be drawn over the course of time, i.e. within one simulation run.
Nevertheless, in contrast to other innovation models that study the behaviour of finite systems, this model is capable of reproducing the open ended sample space expanding nature of innovation processes, although the production of innovations does not happen combinatorially but probabilistically. However, this relieves the model from the necessity to make a priori assumptions about any interaction topologies.
Moreover, the model is comparable to results from empirical data. As Eq.(2.4) shows the number of innovations $D(N)$ in the sequence S grows sub-linearly like Heaps' law [47]. One can furthermore find that the frequency-rank distribution of elements in S approximately follows Zip's law [48]. Similar behaviour can be found in the appearance of novelties in texts, Wikipedia pages, online music catalogues etc. [12]

2.2.3 Evolution of Technology: The Arthur-Polak Model

In principle, the creation of new technologies can be seen as a combinatorial evolutionary process: new devices are constructed from already existing ones, which in turn can be used as possible components for the construction of further new devices and so forth. In 2006, Arthur et al. included this aspect in a computer model in order to simulate the evolution of technology [5].
The technological elements in the model are logical circuits which are randomly

wired together to create new circuits. To control the process a list of goals, desirable functions the circuits should fulfil, is defined. This list of goals defines the utility landscape of the process [33]. Furthermore, each circuit has a cost that is determined by the number of its components. The model starts with a set that initially only contains primitive components, like the NAND-gate. A certain number of these primitive components is then randomly selected and the circuits are randomly wired together. These newly created circuits are added to the set of existing circuits, if they fulfil one of the specified goals. Additionally, an already existing circuit might be taken from the set and replaced by a newly created circuit fulfilling its goals more effectively and/or costing less. Therefore the model contains two possible ways an innovation might happen: either by achieving goals that have not been achieved yet or by improving already existing technologies. Thus, it is observed that there are constructive interactions (two circuits combined to a new one fulfilling specific goals) as well as destructive interactions (two circuits create a replacement for an existing circuit).

The adjacent possible in this model is the set of all circuits that can be created by the currently available circuits. In order to reach more complex circuits that are in the far possible it is first required that simple circuits are made in order to serve as *stepping stones* [49]. As more gates are added to the system, more and more complex circuits can be created and the adjacent possible expands. One innovation can trigger the arrival of further innovations and, in certain cases, the creation of a key circuit enables a huge range of new and more complicated circuits to be built [5]. Therefore, one can observe bursts of creation of new technologies, but likewise there can be cascades of destruction. For example, the replacement of a circuit might leave its components with no further use which subsequently leads to them being replaced as well [5].

One can see that even though the Arthur-Polak model focuses on the combinatorial aspect of innovation, can replicate the expansion of the adjacent possible and the phenomenon of creative destruction [3, 4] as well. The model is limited though by the need for a predefined list of desirable functions for circuits to fulfil. Nevertheless, it laid out the foundations to further explore the aspects of expanding sample spaces in the model of Loreto *et al.* [12, 11] as well as the interplay of constructive and destructive interactions in the models of Thurner *et al.* [7, 8, 9, 10].

Chapter 3

Methodology

3.1 An Agent-Based Model for Innovation

The central aim of this thesis is to investigate the hypothesis that, as a group of collaborators is growing in size, the influence of the Ringelmann effect leads to a decline in the production of new, innovative knowledge. For a first approach to this hypothesis, a simple numerical model is implemented to replicate how new knowledge is created by a group of collaborators through exchange and recombination of existing knowledge. The influence of the Ringelmann effect is then thoroughly studied within this model. In the following sections, a detailed description of this model is given.

3.1.1 Agents and Knowledge Elements

In this simple model, the group of collaborators is represented as a set of N agents. At any time t an agent i can be characterized by a scalar state $\sigma_i(t)$ and the whole team is represented by the state vector $\vec{\sigma}(t) = (\sigma_1(t), \ldots, \sigma_N(t))$. The dimension N is the *group size*, which is assumed to be a constant parameter (A table with all variables and parameters is provided in Tab.(3.1)). The setup is similar to opinion dynamic models [26, 27, 28]. However, unlike these models, where the states of the agents represent their opinion on one or more topics, here the states of the agents represent their most recent piece of knowledge they possess, e.g. an idea that they currently have. Like in opinion dynamics models, the agents can interact among

each other at any time and change their states with the additional possibility to create new knowledge elements through these interactions. For simplicity, the knowledge elements are labelled with an index $m \in \mathbb{N}$ corresponding to their order of appearance in the system: $m = 1$ is the first knowledge element, $m = 2$ the second, etc. The set of all unique knowledge elements that have been created up to time t is then $I(t) = \{1, 2, \ldots, m, \ldots, M\}$ with M as the largest, or in this context most recent, knowledge element at time t. $I(t)$ represents the total knowledge that is available to the agents at t as the states of the agents are $\sigma_i(t) \in I(t)$. As already mentioned, the agents can create new knowledge elements by interacting with each other. The new elements are then added to $I(t)$, thus expanding it[1]. This expansion of $I(t)$ is similar to the expansion of the adjacent possible in [12, 11].
At any time, several agents can possess the same amount of knowledge, i.e. their states are the same elements from $I(t)$. The *abundance* of a knowledge element l is defined as the number of agents for whose state at time t one has $\sigma_i(t) = l$. It is denoted as $n_l(t)$, with $\sum_l n_l(t) = N$. Furthermore, to measure the activity of an agent, the *contribution* of an agent i is defined as the number of successful interactions the agent i has had with another agent. It is denoted with $c_i(t)$. How exactly these interaction processes change the states of the agents and create new knowledge elements as well as what counts as a successful interaction will be discussed in detail in the following sections.

3.1.2 Exchanging and Recombining Knowledge

There are two interaction processes linking the agents in the model. First, there is a pairwise interaction between two agents which is referred to as the exchange (**EXCH**) process. It simulates how two people randomly meet, exchange their current knowledge and by recombining it, possibly create a new knowledge element. In general, not all combinations of elements are possible or lead to meaningful outcomes. Therefore, as in the innovation dynamics models of Thurner *et al.* [7, 8, 10], the rule table R^t, indicating which pairs of elements can be recombined to new elements, is defined. By contrast, R^t is not fixed a priori but one

[1]For example: if $\vec{\sigma}(t_1) = (3, 3, 2)$ and $I(t_1) = \{1, 2, 3\}$, the agents $\sigma_1 = 3$ and $\sigma_3 = 2$ could recombine their states to the new element 4. Then one has $\vec{\sigma}(t_2) = (4, 3, 4)$ and $I(t_2) = \{1, 2, 3, 4\}$.

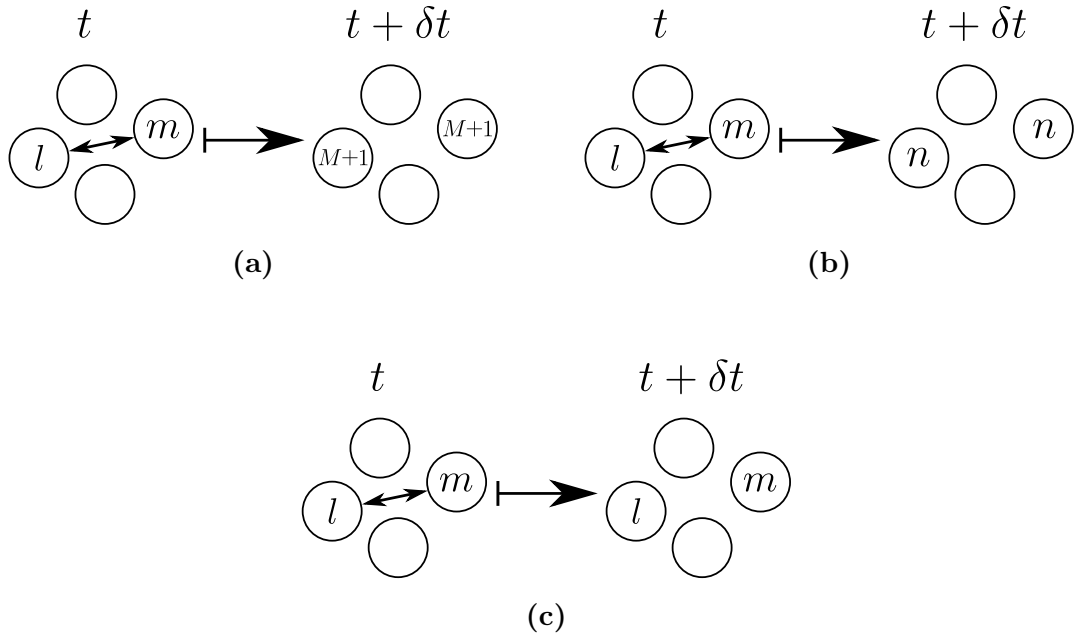


Figure 3.1: There are three possible scenarios for the **EXCH** process. (**a**) Innovation: Two randomly selected agents change their knowledge states from l, m to $M + 1$. (**b**) Update: Two randomly selected agents change their knowledge states from l, m to n. (**c**) Failed communication: Two randomly selected agents remain in their knowledge states l, m.

rather lets the agents through interaction "discover" new rules that then might be added to R^t. So $R^t : I^2 \to I^2$, $(\sigma_i, \sigma_j) \mapsto (\sigma'_i, \sigma'_j)$ is a function containing all the knowledge element pairs and the resulting products at time t. The number of production rules that a knowledge element m is part of at time t is given by the variable $r_m(t)$ which is named the *record* of m. Furthermore, it is assumed that there is an upper bound for the number of production rules that exist for a certain knowledge element which is set by the *saturation parameter* Q. This parameter is comparable to the production rule density in [7, 8, 10]. If for a knowledge element m one has $r_m(t) = Q$, no further rules involving it can be added to R^t. This element is therefore called saturated [2]. The motivation here is that there is a limited potential for a knowledge element to create other elements.

For the **EXCH** process at time t, two agents i, j are randomly selected to change

[2]For example: if $Q = 2$ and at t the element $m = 3$ is already in two production rules: $R^t(2, 3) = 4$ and $R^t(3, 5) = 6$ its record is $r_3 = 2$ therefore $m = 3$ is saturated.

their states from $\sigma_i(t), \sigma_j(t)$ to $\sigma_i(t+\delta t), \sigma_j(t+\delta t)$. Assuming that $\sigma_i(t) = l$ and $\sigma_j(t) = m$, with $l, m \in I(t)$ and the rule table is R^t, there are three possible ways for the this process to play out:

- Nor rule for l, m exists, $\nexists\ R^t(l, m)$, and l, m are both not saturated, $r_l, r_m < Q$. The knowledge element $M + 1$ is created and added to $I(t + \delta t)$, the entry $R^{t+\delta t}(l, m) = M + 1$ is added to the rule table. As seen in Fig.(3.1a), one has:

$$\begin{aligned} \sigma_i(t+\delta t) = M+1 \quad &, \quad c_i(t+\delta t) = c_i(t) + 1 \\ \sigma_j(t+\delta t) = M+1 \quad &, \quad c_j(t+\delta t) = c_j(t) + 1 \end{aligned}$$

  Here an innovation occurs, as a new knowledge element is created. This contribution by both agents increases their contribution counter by 1. For the knowledge elements m, n as well as the newly created element $M + 1$ the abundances n_{ml}, n_m, n_{M+1} are changed accordingly. Furthermore the records r_l, r_m are increased by 1 as a new production rule involving l, m is added to $R^{t+\delta t}$.
- A rule for l, m exists, $\exists\ R^t(l, m) = n$. As seen in Fig.(3.1b), one has:

$$\begin{aligned} \sigma_i(t+\delta t) = n \quad &, \quad c_i(t+\delta t) = c_i(t) + 0.5 \\ \sigma_j(t+\delta t) = n \quad &, \quad c_j(t+\delta t) = c_j(t) + 0.5 \end{aligned}$$

  This outcome is not an innovation as no new knowledge element is created. However, both agents still update their states to a more recent knowledge state. Due to this partial contribution, both agents increase their contribution counter by 0.5. For the knowledge element l, m, n the abundances n_l, n_m, n_n are changed accordingly while the records r_l, r_m are not changed as no new production rule involving l, m is created
- No rule for l, m exists, $\nexists\ R^t(l, m)$, but either l or m is already saturated, $r_l = Q$ or $r_m = Q$. There is neither a knowledge element to which the agents could update nor a new production rule involving l, m that can be created. As seen in Fig.(3.1c), one has:

$$\begin{aligned} \sigma_i(t+\delta t) = l \quad &, \quad c_i(t+\delta t) = c_i(t) \\ \sigma_j(t+\delta t) = m \quad &, \quad c_j(t+\delta t) = c_j(t) \end{aligned}$$

  This outcome represents a failed communication between the agents as their

states are unchanged and no new knowledge is created. As there is no contribution by both agents, $c_i(t)$ and $c_j(t)$ remain unchanged. For the knowledge elements l, m the abundances n_l, n_m as well as the records r_l, r_m are left unchanged as well.

3.1.3 Discarding Knowledge

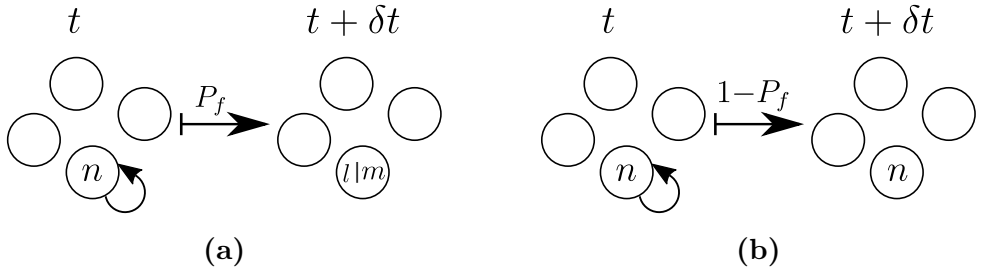


Figure 3.2: There are two possible scenarios for the **DISC** process. (**a**) Discarding: A randomly selected agent changes its knowledge state from n to either l or n. (**b**) Remaining: A randomly selected agent remains in its knowledge state n.

The second process is a self-interaction of agents which is referred to as the discarding (**DISC**) process. It simulates how sometimes a person might forget their current knowledge or simply discards it in favour of older, more trusted knowledge. The discarding of knowledge takes place with the probability P_f, the *forgetting-probability*.
For the **DISC** process at time t an agent k is randomly selected to change its state from $\sigma_k(t)$ to $\sigma_k(t+\delta t)$. Assuming that $\sigma_k(t) = n$ with $n \in I(t)$ and the rule table R^t that contains the rule $R^t(l, m) = n$, there are two possible ways for this process to play out:

- With probability P_f one has, as seen in Fig.(3.2a):

$$\sigma_k(t + \delta t) = l \mid m$$

The agent assumes either l or m with a 50% probability for each possible outcome. This scenario represents an agent forgetting or discarding its current

knowledge. For simplicity, it is assumed here that the agent equally "remembers" both pieces of knowledge that have led to its current state and randomly decides for one of the two possibilities, independently of the state the agent previously occupied. The contribution $c_k(t)$ of the agent is unchanged as nothing new is created. The records of the knowledge elements r_l, r_m, r_n are unchanged as well, while the abundances n_l, n_m, n_n are changed accordingly.

- With probability $P_f - 1$, or if $\sigma_k = 1$, the agent remains in its current state. One has, as seen in Fig.(3.2b):

$$\sigma_k(t + \delta t) = \sigma_k(t)$$

For the knowledge elements the records r_l, r_m, r_n as well as the abundances n_l, n_m, n_n are left unchanged. The contribution $c_k(t)$ of the agent is unchanged as well.

3.1.4 Implementation of the Model

To simulate the model described above the following steps are implemented:

1. Initialize the states of all agents to $\sigma_i = 1 \ \forall \ i$, one therefore has $I = \{1\}$ with $n_1 = N$. Like in the Arthur-Pollak model where the system initially only contains primitive circuits [6] one starts with all agents in the most basic state of knowledge. Furthermore set $S = \{\}$ with $r_1 = 0$. Set the time to $t = 1$
2. Select two random agents i, j in the states σ_i and σ_j. Apply the **EXCH** process to them
3. Draw another random agent k in the state σ_k. Apply the **DISC** process to it.
4. update the time step $t \mapsto t + 1/N$

Steps 2-4 are then repeated. The motivation behind the last step is that in one full time step $t \mapsto t+1$, N pairs of agents are selected to update their states. This way every agent gets on average updated once in one time step. The whole algorithm is implemented using the programming language Julia[3]. All relevant variables and parameters are summarized in Tab.(3.1).

[3]For the complete source code see Appendix C.

Variable	
$\sigma_i(t)$	state of agent i
$c_i(t)$	contribution of agent i
$I(t)$	set of knowledge elements
n_m	abundance of element m
r_m	record of element m
R^t	rule table

(a)

Parameter	
N	number of agents/group size
Q	saturation parameter
P_f	forgetting probability

(b)

Table 3.1: Summary of model variables and parameters.

3.2 Analysing Data from Open-Source-Software Projects

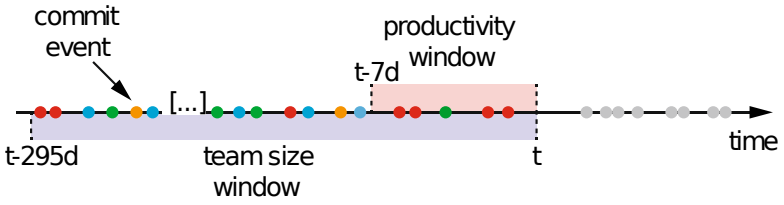


Figure 3.3: A graphical representation of the time-slice analysis. Each dot represents a commit, the colour represents the responsible developer. All dots in the window $[t - 7d, t]$ are the commits $C^R(t)$ while the number of unique colours in the time window $[t - 295d, t]$ represents the team size $M^R(t)$.

The second aim of this thesis is to study data from open-source-software (OSS) projects and look for empirical evidence of the Ringelmann effect which, in the context of this specific data set, corresponds to Brooks' law [20]. The results obtained from this investigation are then compared to the results from the agent-based model of the previous section by interpreting the set of agents as a team of developers and the knowledge elements as lines of code. An exchange of knowledge might occur by two developers co-editing the same file, while an innovation is to add or change a line of code possibly resulting in an improved version of the initial file.

The specific data set that is considered, is a collection of the histories of 85 Open-

Source-Software (OSS) projects from GitHub [24][4]. It contains, for each repository, the information of all commits made over the course of the project, time and date of each commit, the author who committed and the files that have been edited.
The possible influence of Brooks' law is studied by measuring how the number of commits from the developers depends on the size of the respective developer team. For this purpose one needs a way to extract these two variables from the data set. The method for doing so has been extensively discussed in [24]. The same method is adopted here.

3.2.1 A Time-Slice Analysis of Commit-Events

Intuitively, one could take the total number of developers as well as the total number of commits within a repository R as measures for team size and productivity. However, the authors of [24] chose a more fine grained analysis to better capture the fluidity of collaborative teams. Looking at the distribution of time differences between two commit events for all repositories, it is observed that most commits fall in a time window of 7 days. Thus, one defines the variable $C^R(t)$ as the number of commits that occurred in a time window of $[t - 7d, t]$ within repository R. Likewise, one may look at the distribution of times the developers in all the repositories are inactive. It can be observed that apart from one time contributors most developers are active within a time window of 295 days, while developers who have been inactive for a period longer than that are most likely not to commit again. Therefore, the variable $M^R(t)$ is defined as the number of developers that have been active within a time window of $[t - 295d, t]$ within repository R.
As pictured in Fig.(3.3), the team that is responsible for the commits done in a 7-day time window is interpreted as consisting of all the developers who have been active within the last 295 days. Therefore, the variable $M_R(t)$ measures the team size and roughly corresponds to the group size N in the agent-based model of the previous section. The variable $C^R(t)$ measures the productivity of the team and roughly corresponds to the contribution of the agents $c_i(t)$.

[4]The raw data was kindly provided by Ingo Scholtes with the permission of Frank Schweitzer.

Variable	
$C^R(t)$	Number of commits in time window $[t - 7d, t]$ within repository R
$M^R(t)$	Number of active developers in time window $[t - 295d, t]$ within repository R

Table 3.2: Summary of the data set variables.

Chapter 4

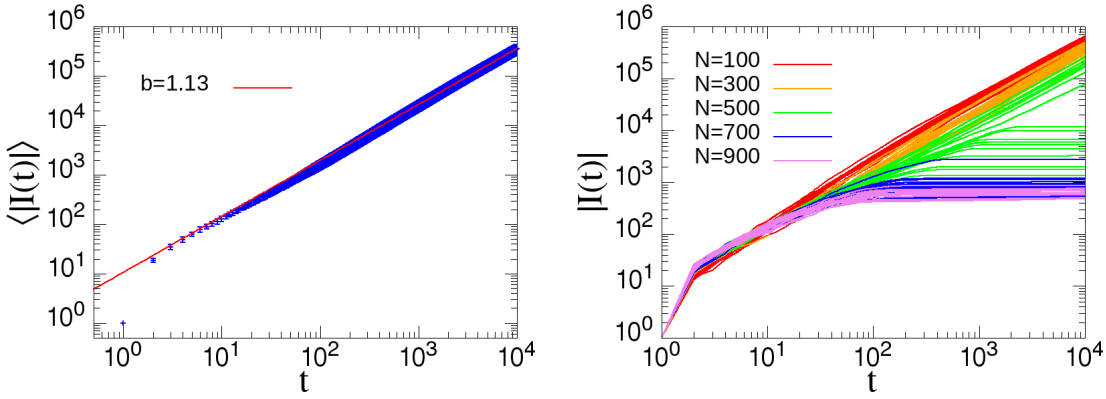
Results

4.1 Simulation Results

In the first part of the analysis, the behaviour of the variables from Tab.(3.1a) is studied. In particular, it is investigated how these variables, as the simulation unfolds, change over time and how their dynamics depends on the parameters from Tab.(3.1b) with a focus on parameter N.

4.1.1 Increase of Knowledge Elements

As established in Section 3.1.1, the set of knowledge elements $I(t)$ gradually grows as the simulation progresses and new elements are added to the set through combination of existing elements. In this section, the dynamics of this growth is analysed in detail. To do so, the simulation is run with particular choices of parameters N, Q, P_f for a fixed number of time steps T. At each time step t the number of knowledge elements, which corresponds to the size of the set of knowledge elements $|I(t)|$, is measured. This analysis is repeated for different N in order to get a first impression of how the growth of $I(t)$ is affected by the number of agents. Furthermore, to test the stability of the behaviour of $I(t)$, each simulation run is repeated multiple times.
We see in Fig.(4.1a) that for $N = 300$ the set $I(t)$ grows for all time steps. Fitting the data with a power-law function $f(x) = a\ x^b$ shows that the growth is approximately linear as $b \approx 1$.



(a) **(b)**

Figure 4.1: The size of the set of knowledge elements $|I(t)|$ was measured each time step with $N = 100, 300, 500, 700, 900$ and the other parameters fixed to $Q = 4$ and $P_f = 0, 15$ for all selections of N. Each simulation was run for $T = 10000$ time steps and for each choice of parameters the simulation was repeated 20 times to test the probability of the behaviour of $|I(t)|$. All plots are in log-log scale. **(a)** shows $|I(t)|$ averaged over all repetitions for $N = 300$. The obtained trajectory was fitted with a function $f(x) = a\ x^b$. **(b)** shows all obtained trajectories of $|I(t)|$ for all sections of N.

This is not surprising as at each time step N pairs of agents are drawn for the **EXCH** process. Therefore, it is not possible that more than N new knowledge elements are produced in one time step. In other words, $I(t)$ cannot grow by more than N. We see in Fig.(4.1b) that for all observed values of N the set $I(t)$ initially grows linearly[1]. However, for $N = 100, 300$ we observe that $I(t)$ continues to grow for all further time steps whereas for $N = 700, 900$ the growth of $I(t)$ slows down at some point and eventually $I(t)$ completely ceases to grow for all further time steps. Furthermore, we observe that for both cases $N = 100, 300$ and $N = 700, 900$ the respective behaviour is consistent for all repeated simulation runs.
More precisely, Fig.(4.1b) shows that as N gets larger we can observe a transition in the growth dynamics of $I(t)$. We change from an *innovation phase* where the trajectories of $|I(t)|$ keep growing linearly for all t to a *saturation phase* where they converge for all t. We also see in Fig.(4.1b) that for $N = 500$ and $Q = 4, P_f = 0.15$ we are at the edge of this transition as for some simulation runs $I(t)$ diverges while

[1]Fitting the data over the first time steps yields exponents $b \approx 1$ for all the observed N.

for others it converges.

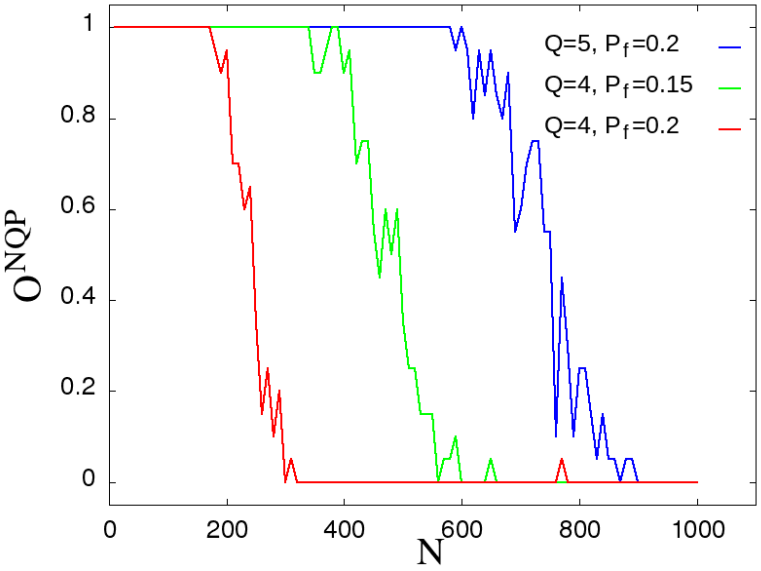


Figure 4.2: The order-parameter O^{NQP_f} was measured for different choices of N with fixed parameters $Q = 4$ and $P_f = 0.15$. In order to obtain O^{NQP_f} for each choice of N the simulation was repeated 20 times for 10000 time steps.

In order to further investigate the growth dynamics of $I(t)$, again a system is simulated with certain parameter choices for a fixed number of time steps and each simulation run is repeated multiple times. The order-parameter O^{NQP_f} which corresponds to the probability that $I(t)$ grows linearly over time for a certain choice of N, Q, P_f[2] is then computed. In Fig.(4.2), we already see the above-mentioned transition in the trajectory of O^{NQP_f}, measured for different N with the other parameters set to $Q = 4, P_f = 0.15$. To get a more detailed picture of the transition, O^{NQP_f} is measured for various values of N, Q, P_f over sufficiently large size ranges.
In Fig(4.3), we see two clearly separated regions for all plots where $I(t)$ either always diverges or converges. By increasing N, we observe a critical value above which the transition from the innovation phase to the saturation phase takes place. Decreasing P_f increases the critical value of N. However, no matter how small P_f is chosen, there is always a critical N.

[2]See Appendix A for a detailed definition.

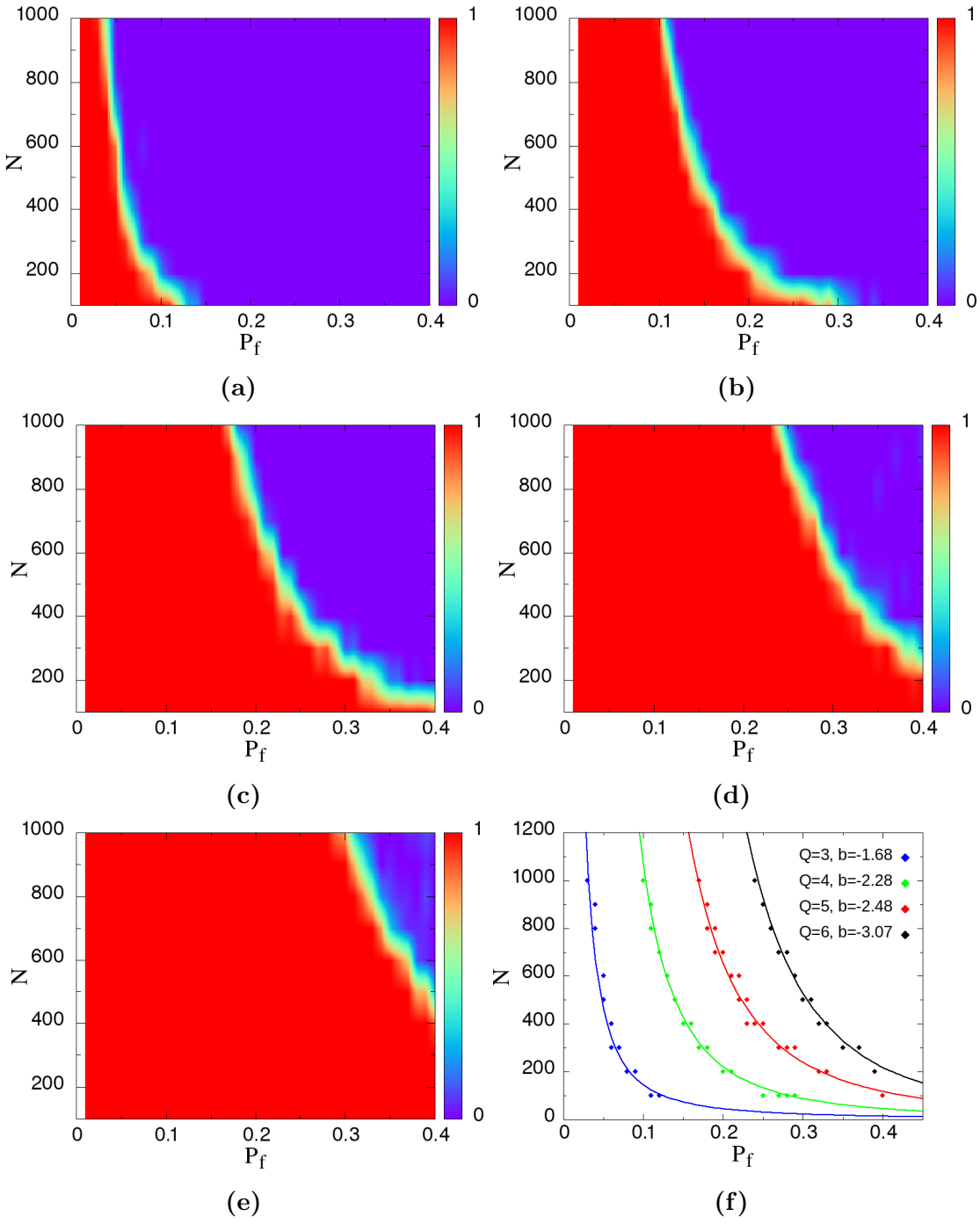


Figure 4.3: The order-parameter O^{NQP_f} was measured for different choices of N, P_f with Q fixed; (**a**) $Q = 3$ (**b**) $Q = 4$ (**c**) $Q = 5$ (**d**) $Q = 6$ (**e**) $Q = 7$. In order to get O^{NQP_f} for each choice of parameters N, Q, P_f the simulation was repeated 20 times for 10000 time steps. The obtained values for O^{NQP_f} are plotted as heat-maps. **(f)** The boundary curves of the transitions fitted with $f(x) = a\ x^b$ using the algorithm of [50].

In Fig.(4.3f), we see that by increasing Q the boundary of that phase transition moves further to the top right. So we observe that the size of the system acts as a critical parameter that allows us to separate the innovation phase from the saturation phase. Where this transition takes place depends significantly on the parameters Q, P_f.

4.1.2 Contribution per Agent

The next variable that is studied, is the contribution of an agent $c_i(t)$. As established in Section 3.1.2, in the **EXCH** process two agents can either innovate, update or the communication between them fails. Depending on this outcome, their individual contribution $c_i(t)$, reflecting their activity, is increased or not. To investigate the activity of the whole set of agents, the contribution of all agents at time t is computed as $c_{tot}(t) = \sum_{i=1}^{N} c_i(t)$ and to get the average contribution per agent $c_{rel}(t) = c_{tot}/N$ is computed. In order to measure how the dynamics of $c_{rel}(t)$ depends on the parameter N, the simulation is run with particular choices of parameters N, Q, P_f for a fixed number of time steps T and $c_{rel}(T)$ is measured after each run. This way, $c_{rel}(T)$ is computed for different choices of N, with Q, P_f fixed. Each simulation run is repeated several times to test the stability of the behaviour of $c_{rel}(T)$.
In Fig.(4.4a), we observe that for $Q = 4, P_f = 0.15$ the trajectories of $\langle c_{rel} \rangle$ plotted against N generally decline for a growing N. We furthermore observe that choosing a larger value for the parameter T shifts the trajectory upwards on the y-axis, as more time has passed and the agents have had more possibilities to exchange knowledge, thus increasing their $c_i(t)$. However, a more detailed inspection reveals that the choice of T has a more significant impact on the trajectory of $\langle c_{rel} \rangle$. In Fig.(4.4a), we discover for $T = 1000$ that around $N = 700$ there is a sharp transition where $\langle c_{rel} \rangle$ converges to a constant value. This is in accordance with Fig.(4.2) and Fig.(4.3b), where we see a corresponding transition in $|I(t)|$ changing from diverging to converging around the same value for N and the same choices for Q, P_f. When $I(t)$ converges this means that no new knowledge elements are created by the agents, which results in a stagnation in the contribution per agent. For smaller values of $T = 10, 100$ we do not see this sharp transition in Fig.(4.4a).

From this one can conclude that the transition we have observed in the previous section only happens after the simulation has run for a sufficiently long time. This indicates that the creation of new knowledge elements by the agents is stable in the short run, whereas in the long run the productivity of a larger number of agents reaches a steady state.

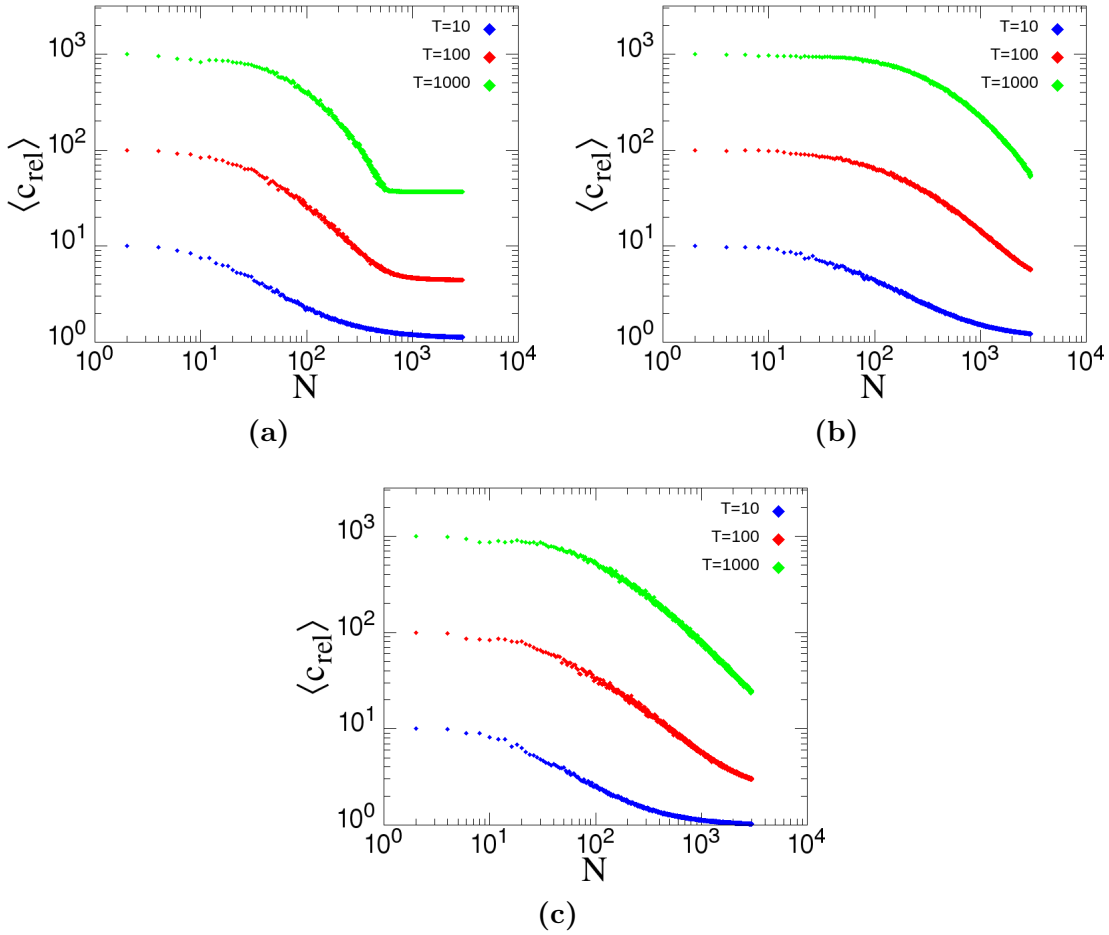


Figure 4.4: The contribution per agent $\langle c_{rel}(T)\rangle$ was measured after $T = 10, 100, 1000$ time steps for different selections of N and with **(a)** $Q = 4, P_f = 0.15$, **(b)** $Q = 6, P_f = 0.14$, **(c)** $Q = 4, P_f = 0.05$. The averages were taken over 20 repetitions. The standard deviation was taken as error. Due to the error bars being of the dimension of the point size they are not shown in the plots. All plots are in log-log scale.

We furthermore see in Fig.(4.4b) and Fig.(4.4c) that $\langle c_{rel}\rangle$ is declining with N, but

there is no sharp transition for all choices of T. If we compare these results with Fig.(4.3), we see that we are in the region where $I(t)$ is diverging. So we observe that regardless of whether $I(t)$ converges or diverges, the contribution per agent declines with a larger number of agents. Furthermore, the choices of simulation parameters Q, P_f as well as T change the shape of the function $c_{rel}(T)$ vs. N.

4.1.3 Change of Abundances of Knowledge Elements

The next variable that is studied, is the abundance of the knowledge elements $n_l(t)$. As established in Section 3.1, the agents change their knowledge states through the **EXCH**- and **DISC** process to other elements from $I(t)$. Therefore, the abundance $n_l(t)$ of a particular knowledge element l is not constant but rather changes over the time of the simulation. The dynamics of $n_l(t)$ is investigated by letting the simulation run with fixed parameters N, Q, P_f for a certain number of time steps and measuring n_l at each time step t for every $l \in I(t)$. Plotting $n_l(t)$ against l gives the frequency distributions of knowledge elements at time t. The set $I(t)$ is the domain of this distribution, which, as shown in Fig.(4.1b), grows with t. As established in Section 3.1.1, the magnitude of l indicates the sequence of appearance for a knowledge element. Therefore, the x-axis of the distribution shows the knowledge elements from oldest (low valued l) to novel (high valued l)[3]. Furthermore, n_l is the number of agents with state $\sigma_i(t) = l$. As a consequence, $n_l(t)$ plotted against l at a certain time t shows the distribution of agents over the existing knowledge elements, from novel to old.

In Fig.(4.5a), we see that for $N = 300, Q = 4$ and $P_f = 0.15$ at $t = 50$ the distribution has two distinct modes. The first one is located around the smallest values of $I(t)$ while the second mode is around the maximum value of $I(t)$. This bi-modal structure remains the same for all observed time steps apart from local fluctuations in the frequencies of some elements. We always observe one mode around the smallest and one mode around the largest values of $I(t)$. As this set is expanding with t, we see that the maximum value of $I(t)$ and with it the second mode gradually moves further to the right.

[3]The terms low and high valued, respectively old and novel, for an element l are in relation to the time t as the domain of the distribution $I(t)$ expands: an element l can be of high value compared to all elements of $I(t_1)$ but of low value compared to $I(t_2)$, $t_2 > t_1$.

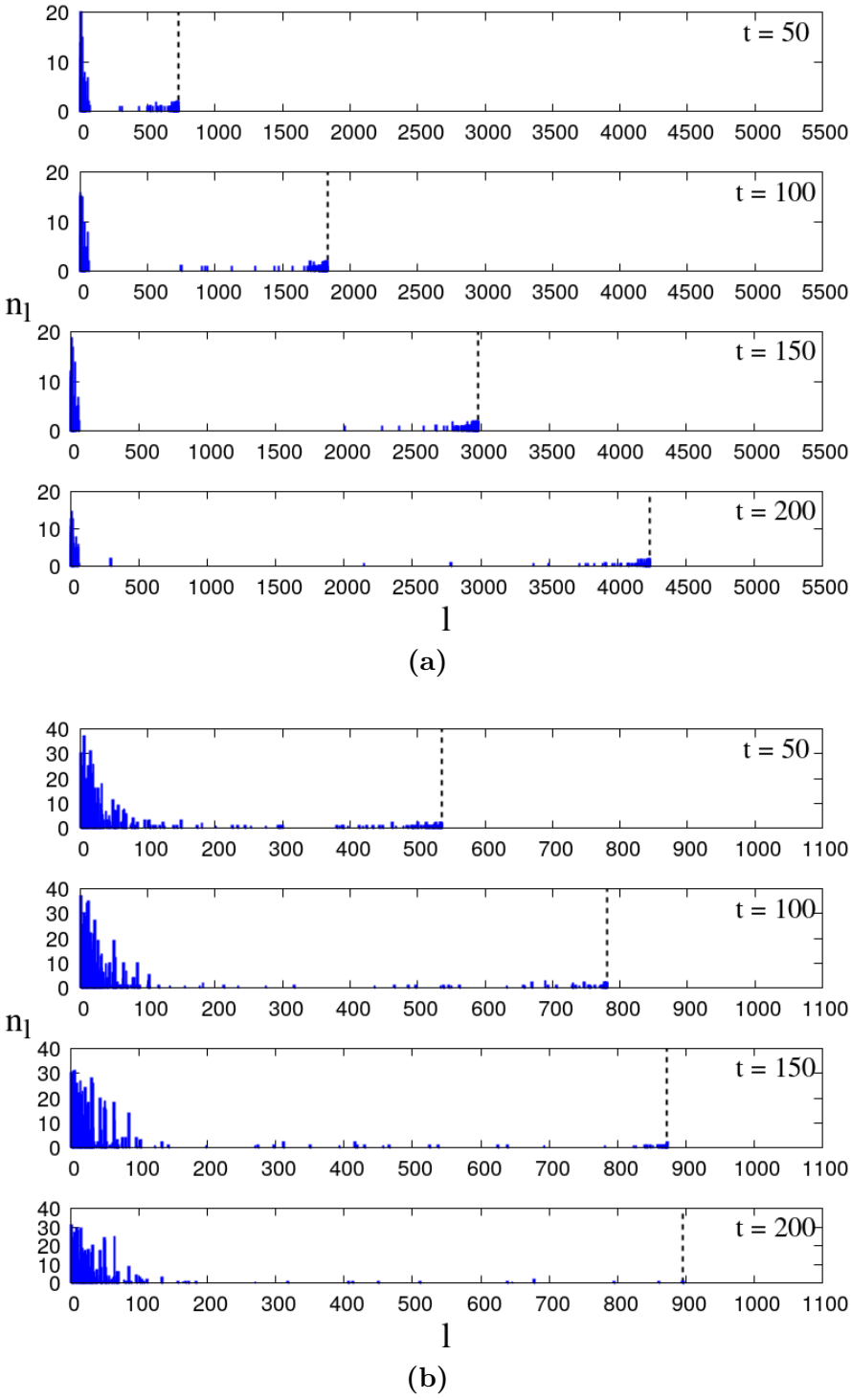


Figure 4.5: The abundance of a knowledge element n_l was measured for each $l \in I(t)$ at selected time steps $t = 50, 100, 150, 200$ with fixed parameters $Q = 4$ and $P_f = 0,15$ and with (**a**) $N = 300$ and (**b**) $N = 700$; the dashed line marks the highest valued knowledge element of $I(t)$ at the current time step t.

The behaviour of the second mode is similar to a pulse train of signals that is moving to the right of the distribution.
In Fig.(4.5b), we see for $N = 700, Q = 4$ and $P_f = 0.15$ initially a similar bi-modal behaviour of the distribution where one mode is always around the smallest values of $I(t)$ while the other one is always around the maximum value of $I(t)$ and gradually moves to the right.
At later time steps we observe that $I(t)$ ceases to grow and the second mode of the distribution vanishes. For all further time steps only the first mode occupying the smallest values of $I(t)$ remains.
The observations in Fig.(4.5) are in agreement with what we see in Fig.(4.1b) where the growth of $I(t)$ diverges for $N = 300$ or converges for $N = 700$ (with $Q = 4$ and $P_f = 0.15$ for both). In other words, the disappearance of the second mode of the distribution and the convergence of the growth of $I(t)$ are two aspects of the same phenomenon.
If we look back to the definition of the **EXCH** process in Section 3.1.2, we see that when two random agents are selected, there are three possible outcomes for this process: the states of both agents are knowledge elements from the second mode of the distribution. As these elements have just recently been added to the system they are probably not saturated yet, i.e. their record is not full, and it is unlikely that there is already a production rule involving both elements. For this reason, the **EXCH** process will probably result in an innovation event and a new knowledge element is created by the agents, compare Fig.(3.1a). On the other hand, the states of both agents can be knowledge elements from the first mode of the distribution. As these knowledge elements have been present in the system for a longer time, both elements are probably already saturated, and it is very likely that there exists a production rule involving both elements. Therefore, the **EXCH** process will probably result in an update event, compare Fig.(3.1b). Finally, the state of one agent can be an element from the first mode and the state of the other one an element from the second mode. The knowledge element of the first mode is again probably saturated and it is very unlikely that there exists a production rule involving both elements. Consequently, the **EXCH** process will probably result in a case of failed communication, compare Fig.(3.1c).
This tells us about the significant role of the second mode of the distribution in

the production of innovations. This part of the distribution is therefore named the *Innovation Train* (IT) for further references. As we also see that the bulk of agents is in knowledge states from the first mode, this part of the distribution is named *Bulk Mode* (BM) for further references. Based on the results in Fig.(4.1b), Fig(4.3) and Fig(4.5), one can conclude that a system being either in the innovation or the saturation phase depends on the IT continuing to exist for all time steps or disappearing at some point.

4.1.4 Further Investigation of the Innovation Train

The size of the IT is defined as as the number of agents whose states at time t are knowledge elements from this mode of the distribution and it is denoted as $N_{train}(t)$[4]. To further investigate the IT, it is studied how the trajectory of $N_{train}(t)$ changes over time with a focus on cases where the IT either remains in the system or eventually disappears. To do this, the simulation is run with particular choices of parameters N, Q, P_f for a fixed number of time steps T. $N_{train}(t)$ is then measured at each time step t. Each simulation run is again repeated several times to test the stability of the behaviour of $N_{train}(t)$ vs. t.
We see in Fig.(4.6a) that, in the long run, for $N = 300$ the IT settles down to a constant size, apart from minor fluctuations. On the other hand, for $N = 700$, we see that N_T converges to zero in the long run, in accordance with Fig.(4.5b). We also observe in Fig.(4.6b) that $N_{train}(t)$ is not settling to a constant value before eventually decreasing but rather $N_{train}(t)$ is "strictly" decreasing to zero. For $N = 500$, we see that for some simulation runs $N_{train}(t)$ settles to a constant size while for others it decreases to zero. This is in accordance with Fig.(4.1b), where we observe that for some simulation runs $|I(t)|$ diverges (the IT remains) while for others $|I(t)|$ converges (the IT disappears) for the same number of agents with the same choices for Q, P_f. To continue the analysis of the IT, the dependence of $N_{train}(t)$ on the parameter N itself is investigated. In order to do this, the simulation is run with particular choices of parameters N, Q, P_f for a fixed number of time steps T and $N_{train}(T)$ is measured after each run (a sufficiently large T is chosen). Each simulation run is repeated several times and the average $\langle N_{train}(T) \rangle$

[4]For a detailed definition of $N_{train}(t)$ and how to properly measure it see Appendix B.

is taken over all runs. $\langle N_{train}(T)\rangle$ is computed this way for different choices of N. In Fig.(4.6c), we see that as N is increased, $\langle N_{train}(T)\rangle$ grows and reaches a maximum value around $N = 300$ from where it decreases again until it converges to zero around $N = 600$. If we compare these results with Fig.(4.2) and Fig.(4.3b), we see that for this choice of parameters $Q = 4$ and $P_f = 0.15$ for $N \leq 300$, we

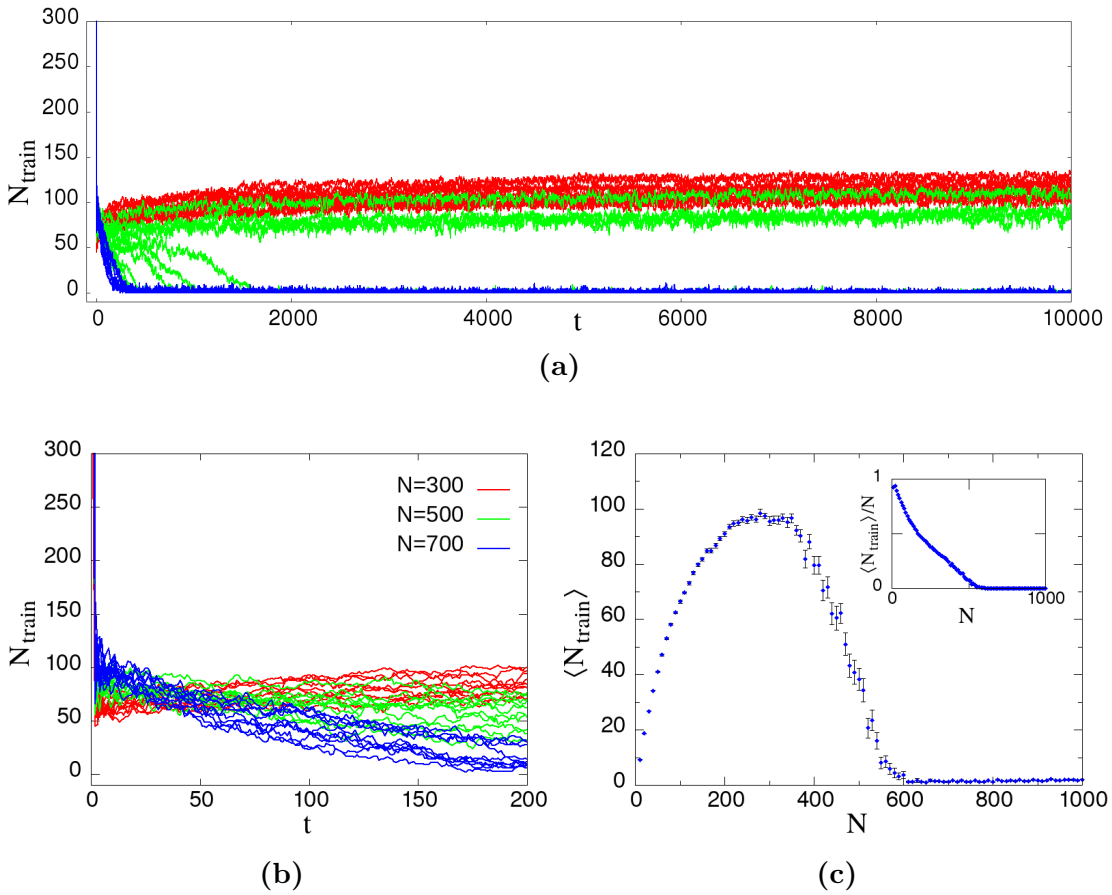


Figure 4.6: The size of the IT N_{train} was measured at each time step for $N = 300, 500, 700$ with fixed parameters $Q = 4$ and $P_f = 0, 15$, and a fixed time frame of 10000 simulation steps, for each choice of parameters the simulation was repeated 20 times. **(a)** shows the long run of $N_{train}(t)$ for all 10000 time steps **(b)** shows the short run of $N_{train}(t)$ for the first 200 time steps. **(c)** Main Plot: $\langle N_{train}(T)\rangle$ was measured after 1000 time steps for different choices for N with fixed parameters $Q = 4$ and $P_f = 0, 15$. The average was taken over 100 repetitions. The error bars show the standard error of the average. Inset: $\langle N_{train}(T)\rangle/N$ was measured with the results from the Main-Plot.

are in the innovation phase where the IT always remains. $\langle N_{train}(T)\rangle$ grows with N because when there are more agents in the system, more of them can potentially gather in the IT. For $300 < N \leq 600$ we are in the transition phase where the IT disappears in some cases while in others it remains, as we observed in Fig.(4.6a) for $N = 500$. This means, in some repetitions of the simulation one has $N_{train}(T) \neq 0$ while in others one has $N_{train}(T) = 0$, which ultimately causes the average size of the IT to decline, as the proportion of trajectories where the IT "survives" becomes smaller. We observes that in Fig.(4.6c) the error bars are larger in this region of N. Finally, for $N > 600$ we are completely in the saturation phase where the IT always disappears and thus $\langle N_{train}(T)\rangle$ becomes zero.
The inset of Fig.(4.6c) shows $\langle N_{train}(T)\rangle/N$ the fraction the IT has on the whole system. We see that the trajectory of $\langle N_{train}(T)\rangle/N$ vs. N is strictly decreasing towards zero. We especially see that in the case of very small numbers of agents ($N = 5 - 10$) the size of the IT might be small compared to systems with a higher number of agents. However, its relative size is almost one, meaning the innovation-train encompasses almost the whole system. In other words, almost all agents take part in the innovation process.

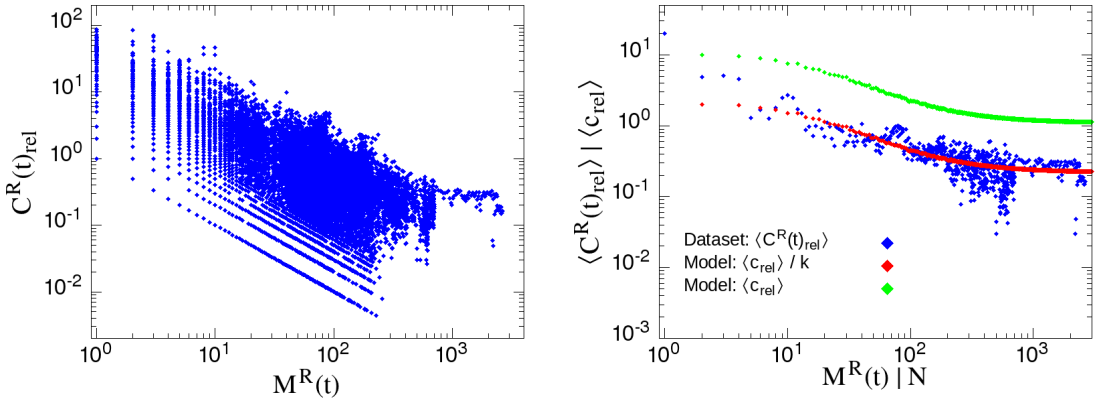
4.2 Open-Source-Software Projects Results

The analysis is continued in this section by looking at the data set described in Section 3.2 in order to find empirical evidence for the influence of the Ringelmann effect, or in the context of OSS projects, Brooks' law. The data analysis in this section mainly follows the one in [24]. The results obtained from the data set are then compared to the results from the computer model of the previous section.

4.2.1 Contribution per Developer

The aim of this section is to observe how, within a team of software developers affiliated to a certain GitHub project, the commits of the developers depend on the team size. In order to do this, the variables $C^R(t)$, the number of commits within a time window of 7 days in a repository R, and $M^R(t)$, the number of active developers within a time window of 295 days in a repository R, are measured. As

in [24], the entire commit history of a repository R is split into intervals of 7 days, which gives us intervals of $[t_0, t_1], [t_1, t_2], \ldots$ (note that one has $[t_{i-1}, t_i] = [t_i - 7d, t_i]$). $C^R(t)$ is then measured as the number of commits that have been done in a time-interval $[t_{i-1}, t_i]$ for this repository. Moreover, for each interval $[t_{i-1}, t_i]$, the number of all developers that have been active within a time-interval $[t_i - 295d, t_i]$ is measured, in order to obtain $M^R(t)$. This quantity characterizes the size of the developer team that is responsible for the commits within the time window $[t_{i-1}, t_i]$ and roughly corresponds to the parameter N, the number of agents in, the model.



(a) **(b)**

Figure 4.7: (a) The average number of commits per team member $C^R_{rel}(t)$ plotted against the team size $M^R(t)$ measured for all repositories **(b)** $\langle C^R_{rel}(t)\rangle$ vs. $M^R(t)$ compared to the results of $\langle c_{rel}(T)\rangle$ vs. N. The values for $c_{rel}(T)$ where computed for $Q = 4, P_f = 0.15$ and $T = 10$. The obtained results where re-scaled by taking $\langle c_{rel}\rangle / k$ with $k = 5$. Each simulation run was repeated 20 times to obtain $\langle c_{rel}(T)\rangle$. All plots are in log-log scale.

For all 58 repositories of the data set the cumulative results for $C^R(t)$ and $M^R(t)$ are collected. Then, by dividing $C^R(t)/M^R(t)$, the mean number of commits per team member $C^R_{rel}(t)$ is obtained which characterizes the activity of the developers and roughly corresponds to the variable $c_{rel}(t)$, the mean contribution per agent, in the model.

We see in Fig.(4.7a) that as the size of a team of software developers grows the mean number of commits per team member declines. For a final analysis this result is compared to the results of the computer model by comparing the behaviour of

$C^R_{rel}(t)$ and $M^R(t)$ with their corresponding variables from the model. We have seen in Fig.(4.4) that c_{rel} declines as well, with a growing number of agents N. To better compare the results of Fig.(4.7a) with Fig.(4.4), the average $\langle C^R_{rel}(t)\rangle$ of all values of $C^R_{rel}(t)$ that correspond to the same team size $M^R(t)$ is taken. c_{rel} is computed as in Fig.(4.4) and averaged over several simulation runs in order to get $\langle c_{rel}\rangle$. The parameters of the simulation are chosen to make sure that the function $\langle c_{rel}\rangle$ vs. N matches the averaged results of Fig.(4.7a).
In Fig.(4.7b) we see $\langle c_{rel}\rangle$ vs. N for $Q = 4, P_f = 0.15$ and $T = 10$. The resulting curve for this choice of simulation parameters shows a similar shape as the mean number of commits per team member, but the results of the simulation show a larger offset on the y-axis. There is, however, not a one-to-one correspondence between a file access in an OSS project and an innovation event in the agent-based model, for instance it could be the case that a comparable innovation in an OSS projects requires multiple commits by the developers. Thus, the results of $\langle c_{rel}\rangle$ can be re-scaled by dividing them through a constant k. Choosing $k = 5$ the trajectory of $\langle c_{rel}\rangle/k$ can be fitted to $\langle C^R_{rel}(t)\rangle$ vs. $\langle M^R(t)\rangle$.
This demonstrates that there is a significant size dependence of the activity of a developer team on its size, confirming the emergence of the Ringelmann-effect, or more specific Brooks'-law, within OSS projects. Furthermore, comparable results to those from this empirical data set have been obtained in a simple computer model.

Chapter 5

Discussion

To summarize this project: a simple agent-based model that replicates how new knowledge is created via pairwise exchange and recombination of existing knowledge has been implemented. By investigating how several of the variables within this model behave, the emergence of the Ringelmann effect in the creation of new, innovative knowledge by "teams" of agents has been tested. Secondly, data from OSS projects has been taken and it was investigated whether the influence of Brooks' law (the Ringelmann effect in the context of OSS projects) on the productivity of software developer teams could be observed. Finally, the results obtained from the data set have been compared with the results from the agent-based model.

5.1 Analysis of the Results

Results of the Agent-Based Model

The agent-based model shows that a larger number of agents eventually slows down the innovation process and in extreme cases even brings it to a complete halt. It becomes obvious that in the model the innovation train (IT) is the main driving force in the process of creating new, innovative knowledge elements. Fig.(4.1b) shows that the convergence or divergence of the set of knowledge elements $I(t)$ depends on whether the IT disappears or not. However, it remains somewhat unclear why the disappearance of the IT happens exactly.
As established in Section 4.1.3, in the **EXCH** process of the model selecting two

agents from the IT most probably results in a new knowledge element being produced. However, as we saw in Fig.(4.6c) the fraction of agents that take part in the IT N_{train}/N becomes smaller for larger N. Thus, the probability to select two agents from the IT becomes smaller and it is more likely to select two agents from the so-called bulk mode (BM) or one agent from the IT and one from the BM. Both of the latter cases are unlikely to result in an innovation. This means that in a system with a larger number of agents, many pairings of agents for the **EXCH** process yield neither an update nor an innovation event. Thus, the IT moves forward more slowly. Moreover, one has to consider the **DISC** process that causes agents to discard their current knowledge state in favour of a previous (older) one, with probability P_f. If an agent that is part of the IT is selected for this process it might change its state to a smaller valued knowledge element that is already saturated but not yet part of the BM of the distribution. One can see these agents whose states are *intermediate* knowledge elements as small spikes between the two modes in Fig.(4.5). There are two possibilities for such an agent to become part of the IT again: either through the update case of the **EXCH** process, when the resulting knowledge element to which the agent changes its state is part of the IT again. The second possibility is through an innovation event as the resulting knowledge element is new, thus not saturated, which automatically makes it part of the IT.

Yet, in a larger system the fraction of agents in the BM is greater than the fraction of agents in the IT. Therefore, if an agent in an intermediate knowledge state is selected for the **EXCH** process, it is more likely that the second agent that is selected is from the BM of the distribution. It is then unlikely that the **EXCH** process results in an update of knowledge as there probably does not exist a production rule involving the knowledge elements of both agents. It is also unlikely that an innovation event takes place as the agent from the BM is likely to be in a saturated knowledge state. Thus, the agent remains in the intermediate knowledge state. It might happen that this agent is again selected for the **DISC** process until its state is a knowledge element from the BM. If the process outlined above occurs to all agents from the IT, it eventually vanishes. As we saw in Fig.(4.3) the number of agents N is not the only parameter that might have an influence on whether the IT disappears or not. The saturation parameter Q determines how

many times a knowledge element can be represented in a production rule. The value of Q therefore determines whether there is a higher or lower chance that the **EXCH** process results in an innovation or update case. Likewise, choosing a smaller or larger P_f decreases or increases the probability that agents whose states are elements of the IT change their states to intermediate knowledge elements thereby falling behind the IT.
Nevertheless, the disappearance of the innovation train is in principle a combinatorial problem as a large N means that there is a greater number of possible pairs of agents, but only a pair of agents where both are from the IT can surely create new knowledge elements. The probability for selecting such a pair is $\binom{N_{train}}{2}/\binom{N}{2} \sim \left(\frac{N_{train}}{N}\right)^2$ which gets smaller for larger N, as we saw in Fig.(4.6c) that N_{train}/N declines with N. The probability that two agents, which according to the **EXCH** process could produce an innovation together, "meet", therefore declines with the number of agents. In other words: the larger number of possible communication channels between the agents hinders successful exchange of knowledge and as a consequence the creation of innovations. This is, in principle, exactly what is commonly referred to as a *communication overhead* among the agents.

Results of the Open-Source-Software Projects

The data of the OSS projects have revealed that a larger number of developers eventually slows down the innovation process, in the sense that the commits per developers decline. To further investigate this effect, one can look at the co-editing network between the software developers and observe that in a larger developer team the number of communication links increases significantly [24]. Thus, the conclusion is that in the same way as in the agent-based model this increase of communication overhead is exactly why one observes a decline in the productivity of larger developer teams. When comparing the results of the computer model to the results of the OSS projects, we see that in Fig.(4.7b) one needs to choose the parameter T relatively small in order to fit the data. Fig.(4.5b) and Fig.(4.6b) show that at such small time scales even in larger systems the IT has not disappeared yet. From this it can be concluded that "real" innovation trains are most

probably rather short-lived. The team of collaborating developers reaches their respective goals before the increasing communication overhead causes a complete breakdown of knowledge exchange. Nevertheless, it can still be observed that the larger number of communication links between the developers causes a decline in productivity. The same way we saw in Fig.(4.5b) that even as long as the IT existed, it moved much more slowly compared to a system with a smaller number of agents.

5.2 Limitations

Within the agent-based model that has been implemented and explored in this work, simplifications and a priori assumptions that are crucially for its dynamics were made. As the phenomenon this model explores is rather complex, some of these simplifications might have limited the capacity of the model to properly reproduce all important aspects of it. One such simplification concerns the saturation parameter Q that is assumed constant for all time steps as well as equally affecting all knowledge elements. This is a quite harsh constraint on the system. One could soften this constraint by implementing a variation of the initial model that assigns each knowledge element l an individual saturation value Q_l drawn from a discrete random distribution. Such a variation has actually been implemented and it was initially observed that one still sees a transition as in Fig.(4.3). However, it is more washed out and not as sharp as the one we have seen in the base model. For future work one should investigate in detail how this variation affects the rest of the observed dynamics of the model. Further variations that should be considered as well, for building up on this thesis project, are for example: representing the states of the agents as vectors, in order to capture the knowledge of each agent more realistically. Each entry of a state vector could then either represent their knowledge on different topics or serve as memory about past knowledge. Another variation could be to implement additional types of interactions among the agents, e.g. in addition to exchanging and recombining knowledge, an agent could just copy the knowledge state of another agent, or agents from the IT could directly communicate their knowledge to agents that are still in the BM.
Further limitations to the results of this thesis concern the analysis of the OSS

projects data. One has to take into consideration that in [24] it is argued that the pure number of commits is not quite enough to capture the productivity of a developer team as each change made to a file has a different impact on the project, depending on how significant the changes are. The authors therefore analyse the activity of the software developers by weighting each commit according to how significantly a file was edited. Furthermore, for the time slice analysis of the data set, the number of active developers and commits was measured by using fixed time windows that represent the average activity time of the developers of all repositories. The results in Fig.(4.7a) are then the cumulative results of the time slice analysis for all repositories. A team of developers in one repository might achieve the same output within a certain time window, whereas a team in a different repository achieves it in a much smaller time window. In other words, within a project the speed with which the developers work might be significantly different from those of other projects. Thus, for a more detailed analysis of the OSS-project data one should not only weight the commits but also measure the active developers and number of commits within characteristic time windows for each repository.

Nevertheless, for the simple computer model of this thesis, the significance of knowledge exchanges between the agents is not weighted and each possible innovation has the same value. One merely investigates whether it is possible for the agents to exchange knowledge or not. Therefore, it is sufficient to compare the model to the pure number of commits of the software developers. Moreover, an extension of the time sliced analysis of the OSS-project data by using individual time windows should be considered, but as this would exceed the scope of a master thesis it is as well left open for future work.

Chapter 6

Conclusion

In this thesis, a simple agent-based model as well as data on OSS projects obtained from the platform GitHub has been extensively investigated. In both instances there is significant evidence that, as a group of collaborators grows in size, the increasing communication overhead reduces the rate of successful communications among the collaborators, which ultimately leads to a decline in the production of new, innovative knowledge.
Nevertheless, there are still open questions to be dealt with for future projects. On the one hand, one could try to modify the agent-based model such that it captures interactions among collaborators more realistically by either softening the constraints on the simulation parameters or by implementing more types of interactions among the agents. On the other hand, one could extend the investigation of empirical data by repeating the time sliced analysis of the GitHub projects with time windows that better reflect the individual activity time of each project. Furthermore, one could try to investigate further empirical data sets for the influence of the Ringelmann-effect.
In conclusion, it can be said that despite the problems that can arise through the increasing communication overhead, team-work and cooperation are still important for the creation of knowledge and a large number of collaborators, especially from different backgrounds, is, in principle, not a disadvantage. Therefore, one should not react to the negative effects of the Ringelmann-effect by only supporting small groups of collaborators, but rather restructure larger groups in such a

way that communication among the group members can be more effective. One such approach would be to compartmentalize a group into smaller sub-groups. Exchange of knowledge can then take place within those sub-groups unobstructed by any communication overhead. However, this comes with the risk of creating information flow barriers through the creation of communication "silos". Additionally, there is the possibility of knowledge exchange between individuals of different sub-groups.

One can already observe such processes occurring naturally in the way research groups within universities are organized. A study of 2010 showed that universities tend to have a large number of research groups where each group itself is rather small. This way, communication and knowledge exchange happens among members of a research group on a regular basis without coordination problems and from time to time members from two or more different research groups can collaborate with each other. These cases of intergroup communications are essential for research, as it involves exchanging knowledge from various backgrounds which can result in groundbreaking new ideas (e.g. the collaboration of biologists and chemists which lead to the interdisciplinary field of biochemistry). It turns out that this compartmentalized structure of universities is the most effective way to organize research [51]. In other words: compartmentalization is a direct consequence in order to minimize communication overhead.

Acknowledgements

First and foremost, I want to thank Stefan Thurner for agreeing to be my thesis supervisor and giving me the opportunity to work within his research group. I want to thank Rudolf Hanel for further overseeing my project and continuously supporting me with advice. I want to thank Yu "Ernest" Liu for laying out the foundations of the computer model investigated and extended in this thesis and being available for the exchange of further ideas. I want to thank Frank Schweitzer and Ingo Scholtes for agreeing to give me access to the data that was studied in this thesis. I furthermore want to thank the members of the research group *Section for Science of Complex Systems* for giving me valuable input and feedback, especially: Leonard Horstmeyer, Peter Klimek, Jan Korbel, Tuan Pahm-Minh and Vito D.P. Servedio. Additionally, I want to thank all my friends and colleagues from the collective *Nauturwissenschaftscafe* and the *THINK-Association* for interesting discussions and continuously inspiring me to critical thinking. I also want to thank the students' union *Roter-Vektor Physik* for their support they gave me during my studies. I want to thank the following people for motivating me and providing emotional support: Patrick Braun, Chiara Cardelli, Bernat Corominas-Murtra, Flavio Del Santo and Irene Krassnitzer. I want to thank Brigitte Jeschek and Thomas Kronschläger for helping me with the proofreading of this thesis. Last but no least I want to thank my parents, Johann Lauber and Ingrid Lauber-Pils, whose support made it possible for me to pursue my studies in the first place, as well as my sisters, Jana Lauber and Simone Lauber, who inspired me to continue with my studies.

Bibliography

[1] OED Online. *innovation, n.* http://www.oed.com/view/Entry/96311, July 2018. Oxford University Press, 2018. (accessed November 28, 2018).

[2] J.A. Schumpeter. *Theorie der wirtschaftlichen Entwicklung.* Duncker & Humblot, 1912.

[3] J.A. Schumpeter. *Business cycles: A Theoretical, Historical, and Statistical Analysis of the Capitalist Process.* McGraw-Hill Book Company, inc., 1939.

[4] J. A. Schumpeter. *Capitalism, Socialism and Democracy.* Harper & Brothers, 1942.

[5] W. B. Arthur and W. Polak. The evolution of technology within a simple computer model. *Complexity*, 11(5):23–31, 2006.

[6] W. B. Arthur. *The Nature of Technology: What It Is and How It Evolves.* Free Press, Simon & Schuster, 2009.

[7] R. Hanel, S. A. Kauffman, and S. Thurner. Phase transition in random catalytic networks. *Phys. Rev. E*, 72(3):036117, 2005.

[8] R. Hanel, S. A. Kauffman, and S. Thurner. Towards a physics of evolution: Critical diversity dynamics at the edges of collapse and bursts of diversification. *Phys. Rev. E*, 76(3):030101, 2007.

[9] P. Klimek, S. Thurner, and R. Hanel. Evolutionary dynamics from a variational principle. *Phys. Rev. E*, 82(1):011901, 2010.

[10] S. Thurner, P. Klimek, and R. Hanel. Schumpeterian economic dynamics as a quantifiable model of evolution. *New J. Phys.*, 12(7):075029, 2010.

[11] V. Loreto, V. D. P. Servedio, S. H. Strogatz, and F. Tria. Dynamics on expanding spaces: Modeling the emergence of novelties. In M. Degli-Esposti, E. G. Altmann, and F. Pachet, editors, *Creativity and Universality in Language*, pages 59–83. Springer International Publishing, 2016.

[12] F. Tria, V. Loreto, V. D. P. Servedio, and S. H. Strogatz. The dynamics of correlated novelties. *Sci. Rep.*, 4:5890, 2014.

[13] S. Wuchty, B. F. Jones, and B. Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

[14] S. Johnson. *Where Good Ideas Come From: The Natural History of Innovation.* Penguin Books Limited, 2010.

[15] L. Fleck. *Entstehung und Entwicklung einer wissenschaftlichen Tatsache: Einführung in die Lehre vom Denkstil und Denkkollektiv.* Benno Schwabe & Co, 1935.

[16] T. S. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, 1962.

[17] M. Ringelmann. Recherches sur les moteurs animes: Travail de l'homme. *Annales de l'Institut National Agronomique*, 12(2):1–40, 1913.

[18] A. G. Ingham, G. Levinger, J. Graves, and V. Peckham. The ringelmann effect: Studies of group size and group performance. *J. Exp. Soc. Psychol.*, 10(4):371–384, 1974.

[19] K. D. Williams and S. J. Karau. Social loafing and social compensation: The effects of expectations of co-worker performance. *J. Pers. Soc. Psychol.*, 61(4):570–581, 1991.

[20] F. P. Brooks. *The Mythical Man-month: Essays on Software Engineering.* Essays on software engineering. Addison-Wesley Publishing Company, 1995.

[21] A. Mockus, R. T. Fielding, and J. D. Herbsleb. A case study of open source software development: The apache server. In *Proceedings of the 22Nd International Conference on Software Engineering*, ICSE '00, pages 263–272, New York, 2000. ACM.

[22] A. Mockus, R. T. Fielding, and J. D. Herbsleb. Two cases of open source software development: Apache and mozilla. *ACM Trans. Softw. Eng. Methodol.*, 11(3):309–346, 2002.

[23] R. Kenna and B. Berche. The extensive nature of group quality. *EPL*, 90(5):58002, 2010.

[24] I. Scholtes, P. Mavrodiev, and F. Schweitzer. From aristotle to ringelmann: a large-scale analysis of team productivity and coordination in Open Source Software projects. *Empir. Software Eng.*, 21(2):642–683, 2016.

[25] M. Szell, R. Lambiotte, and S. Thurner. Multirelational organization of large-scale social networks in an online world. *Proc. Natl. Acad. Sci. U.S.A.*, 107(31):13636–13641, 2010.

[26] E. Ben-Naim, L. Frachebourg, and P. L. Krapivsky. Coarsening and persistence in the voter model. *Phys. Rev. E*, 53(4):3078, 1996.

[27] P. L. Krapivsky and S. Redner. Dynamics of majority rule in two-state interacting spin systems. *Phys. Rev. Lett.*, 90(23):238701, 2003.

[28] R. Axelrod. The dissemination of culture: A model with local convergence and global polarization. *J CONFLICT RESOLUT*, 41(2):203–226, 1997.

[29] S. Jain and S. Krishna. Autocatalytic sets and the growth of complexity in an evolutionary model. *Phys. Rev. Lett.*, 81(25):5684–5687, 1998.

[30] S. Jain and S. Krishna. A model for the emergence of cooperation, interdependence, and structure in evolving networks. *Proc. Natl. Acad. Sci. U.S.A.*, 98(2):543–547, 2001.

[31] Thurner S. and Poledna S. Debtrank-transparency: Controlling systemic risk in financial networks. *Sci. Rep.*, 3:1888, 2013.

[32] M. E. J. Newman. *Networks: An Introduction*. Oxford University Publishing, 2010.

[33] S. A. Kauffman. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, 1993.

[34] P. F. Stadler, W. Fontana, and J. H. Miller. Random catalytic reaction networks. *Physica D*, 63(3):378–392, 1993.

[35] J. F. Crow and M. Kimura. *An Introduction to Population Genetics Theory*. Burgess Publishing Company, 1970.

[36] P. Schuster and K. Sigmund. Replicator dynamics. *J. Theor. Biol.*, 100(3):533–538, 1983.

[37] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[38] K. P. Hadeler. Stable polymorphisms in a selection model with mutation. *SIAM J. Appl. Math.*, 41(1):1–7, 1981.

[39] W. Fontana, G. P. Wagner, and L. W. Buss. Beyond digital naturalism. *Artif. Life*, 1(1_2):211–227, 1993.

[40] K. M. Page and M. A. Nowak. Unifying evolutionary dynamics. *J. Theor. Biol.*, 219(1):93 – 98, 2002.

[41] P. Gravino, B. Monechi, V. D. P. Servedio, F. Tria, and V. Loreto. Crossing the horizon: exploring the adjacent possible in a cultural system. In *Proceedings of the Seventh International Conference on Computational Creativity, UPMC, Paris, France, June 27 - July 1, 2016.*, pages 115–122, 2016.

[42] S. A. Kauffman. *Investigations*. Oxford University Press, 2000.

[43] H. A. Simon. On a class of skew distribution fucntions. *Biometrika*, 42(3-4):425–440, 1955.

[44] D. Zanette and M. Montemurro. Dynamics of text generation with realistic zipf's distribution. *J. Quant. Linguist.*, 12(1):29–40, 2005.

[45] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks with aging of sites. *Phys. Rev. E*, 62(2):1842–1845, 2000.

[46] G. Pólya. Sur quelques points de la théorie des probabilités. *Annales de l'I. H. P.*, 1(2):117–161, 1930.

[47] H. S. Heaps. *Information Retrieval: Computational and Theoretical Aspects.* Academic Press, Inc., 1978.

[48] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley Press, Inc., 1949.

[49] R. Lenski, C. Ofria, R. Pennock, and C. Adami. The evolutionary origin of complex features. *Nature*, 423(6936):139–44, 2003.

[50] R. Hanel, B. Corominas-Murtra, B. Liu, and S. Thurner. Fitting power-laws in empirical data with estimators that work for all exponents. *PLoS One*, 12(2):1–15, 2017.

[51] T. Brandt and T. Schubert. Is the university model an organizational necessity? Scale and agglomeration effects in science. *Scientometrics*, 94(2):541–565, 2013.

Appendix

A. Definition of the Order Parameter

The order parameter O^{NQP_f} is defined as the probability that for a choice of parameters N, Q, P_f the growth of the set $I(t)$ diverges. On further inspection of Fig.(4.1b), we see that the transition between the innovation- and saturation phase can be characterized by the slope of $|I(t)|$ for $t \to \infty$ changing from $\lim_{t\to\infty} \frac{d|I(t)|}{dt} > 1$ to $\lim_{t\to\infty} \frac{d|I(t)|}{dt} = 0$. Therefore, to obtain the order-parameter O^{NQP_f}, the simulation is repeated m-times with the same selection of N, Q, P_f. For each $i-$th repetition $|I(t)|_i$ is measured in order to compute the limiting slope:

$$D_i^{NQP_f} = \lim_{t\to\infty} \frac{d|I(t)|_i}{dt}$$

O^{NQP_f} is then obtained as the fraction of cases where the limiting slope of $|I(t)$ is larger than zero. However, as the slope never converges completely to zero, we introduce the threshold h and set it sufficiently small. The order parameter is then the fraction of cases where we have $D_i^{NQP_f} > h$:

$$O^{NQP_f} = \frac{\sum_{i=1}^{m} \Theta(D_i^{NQP_f} - h)}{m}, \quad \text{where} \quad \Theta(D_i^{NQP_f} - h) = \begin{cases} 0, & D^{NQP_f} < h \\ 1, & D^{NQP_f} \geq h \end{cases}$$

which gives the probability that for a choice of parameters N, Q, P_f the growth of the set $I(t)$ diverges.

B. Definition of the Size of the Innovation Train

The size of the IT, denoted as $N_T(t)$, is defined as the number of agents whose states at time t are knowledge elements from this specific mode of the distribution seen in Fig.(4.5). As the knowledge elements from the IT have just recently been added to the system, they are most likely not saturated yet, i.e. their record is not full. Therefore, in order to get $N_T(t)$, the number of agents whose states at t are non-saturated knowledge elements is measured. To do this, the abundance $n_l(t)$ of each knowledge element $l \in I(t)$ (defined as the number of agents whose state is $\sigma_i(t) = l$) is taken as well as the record $r_l(t)$. $N_T(t)$ is then computed as:

$$N_T(t) = \sum_{l \in I(t)} n_l(t)\ \Theta(Q - r_l(t)), \quad \text{where} \quad \Theta(Q - r_l(t)) = \begin{cases} 0, & r_l(t) > Q \\ 1, & r_l(t) \leq Q \end{cases}$$

C. Source Code

```
#<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
function forgetmod(N::Int64, Q::Int64, P_f::Float64, T::Int64)

 #-----------------------------------------
 innov = 1
 pair = [1,1]

 sigma = ones(Int64,N)
 distr = [N]
 record = [0]
 I_t = Vector{Int64}[ [1,1,1] ]
 I_max = [1]

 S = Dict{Array{Int64,1},Int64}()

 # size innov-train
 N_T = [N]

 # contribution of each agent
 contrib = zeros(N)
 #-----------------------------------------

 #-----------------------------------------
 for t = 1:T
  #----------------------------------------
  for n = 1:N
   # 1.)-----Pick Pair and try to innovate/update--
   i = rand(1:N)
   j = rand(1:N)
   while(i == j)
    j = rand(1:N)
   end
   if ( sigma[i] <= sigma[j] )
    pair[1] = sigma[i]
    pair[2] = sigma[j]
   else
    pair[1] = sigma[j]
    pair[2] = sigma[i]
   end

   #----------------------------------------
   if !haskey(S, pair)
    #---------------------------------------
    if ( record[pair[1]] < Q && record[pair[2]] < Q )

     inov = I_t[end][1]+1
```

```
      sigma[i] = inov
      sigma[j] = inov
      S[pair[:]] = inov

      record[pair[1]] += 1
      record[pair[2]] += sign(pair[1]-pair[2])^2

      distr[pair[1]] -= 1
      distr[pair[2]] -= 1

      push!(record,0)
      push!(distr,2)
      push!(I_t,[inov,pair[1],pair[2]])

      # increase contribution of agents i,j
      contribute[i] += 1.0
      contribute[j] += 1.0

    end
    #-----------------------------------------
  #---------------------------------------
  else

    sigma[i] = S[pair]
    sigma[j] = S[pair]

    distr[S[pair]] += 2
    distr[pair[1]] -= 1
    distr[pair[2]] -= 1

    # partially increase contribution of agents i,j
    contribute[i] += 0.5
    contribute[j] += 0.5

  end
  #---------------------------------------

  # 2.)-----Pick Agent and discrad with P_f--
  if ( rand() <= P_f )

    i = rand(1:N)
    id_old = sigma[i]
    id_new = Int( I_t[id_old][1+rand(1:2)] )
    sigma[i] = id_new
    distr[id_old] -= 1
    distr[id_new] += 1

  end
```

```
    #-----------------------------------------
   end
   #-----------------------------------------
   push!(I_max,I_t[end][1])

   # measure size of innov-train
   push!(N_T,sum(record[sigma].!=Q))

 end
 #-----------------------------------------

 return I_max,contrib,distr,N_T

end
#<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```